

eLektor

www.elektor.com

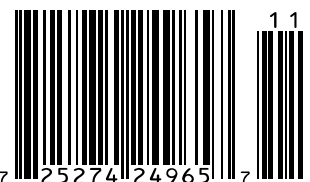
Web Server DIY Style with the Elektor Linux Board

- + **Nixie VU meter**
Glowing columns span 40 years of electronics history
- + **Electricity Meter on the Web**
It's openEnergy, SO-compatible, openSource
- + **A Library for the ElektorBus**
Your collection of instant code chunks

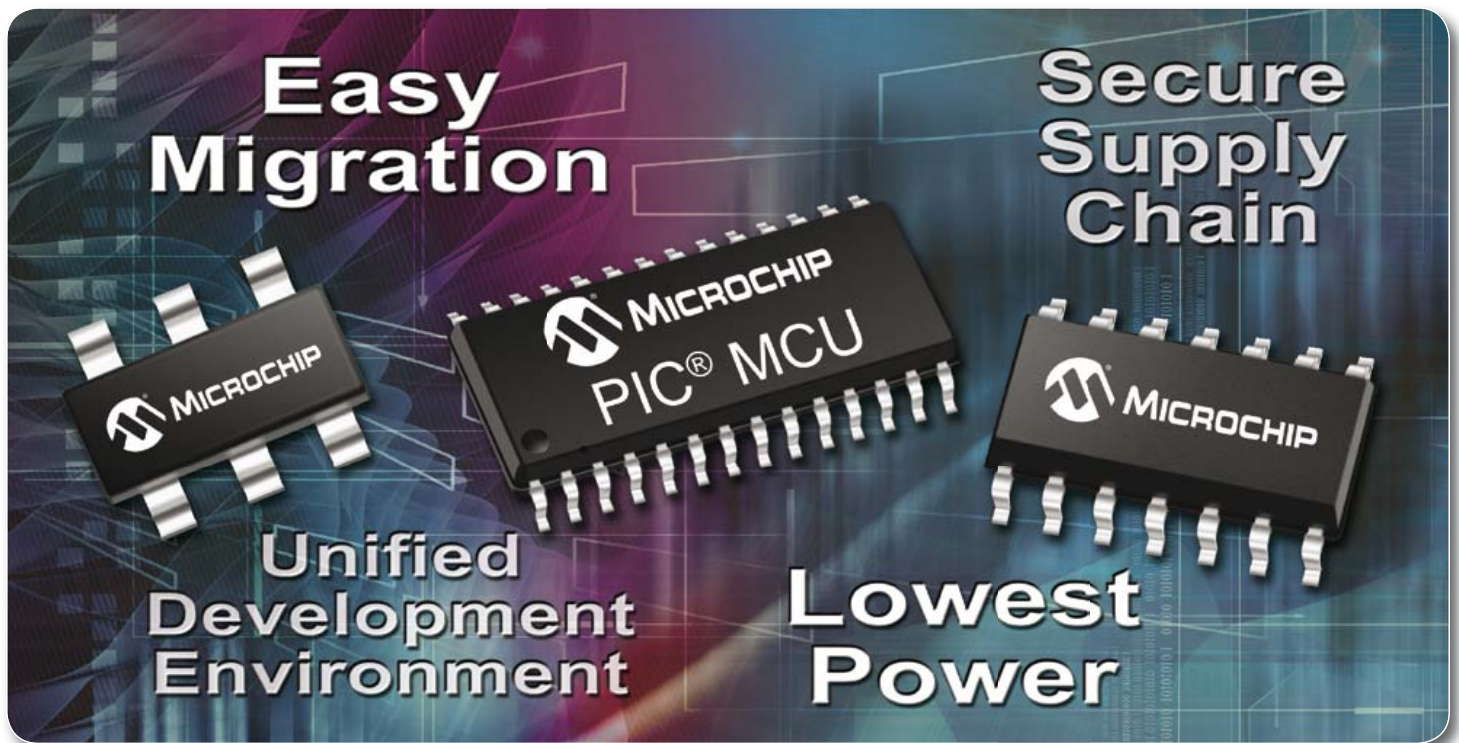
Farewell 7805 & 7905

Switch-mode replacements
do a much better job

US \$ 9.00 - Canada \$ 10.00



The Most Popular 8-bit Microcontrollers



See what makes Microchip the most popular choice for embedded designers:

- Broad portfolio of more than 325 8-bit PIC® microcontrollers
- Easy migration with pin and code compatibility
- Industry's lowest active and sleep power consumption
- Integrated peripherals for USB, CAN and Ethernet with free software
- Interface to the world with LCD drivers and capacitive touch
- Continuous innovation with 70 new 8-bit MCUs in the last two years
- MPLAB® IDE is free, and supports ALL of Microchip's microcontrollers
- Low-cost development tools help speed up prototyping efforts
- Comprehensive technical documentation, app notes and code examples
- World-class 24/7 technical support and training

GET STARTED IN 3 EASY STEPS

1. Purchase the XLP 8-bit Development Board
2. Download free MPLAB® IDE
3. Order samples and start designing!

www.microchip.com/8bit



XLP 8-bit Development Board
(DM240313)



Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

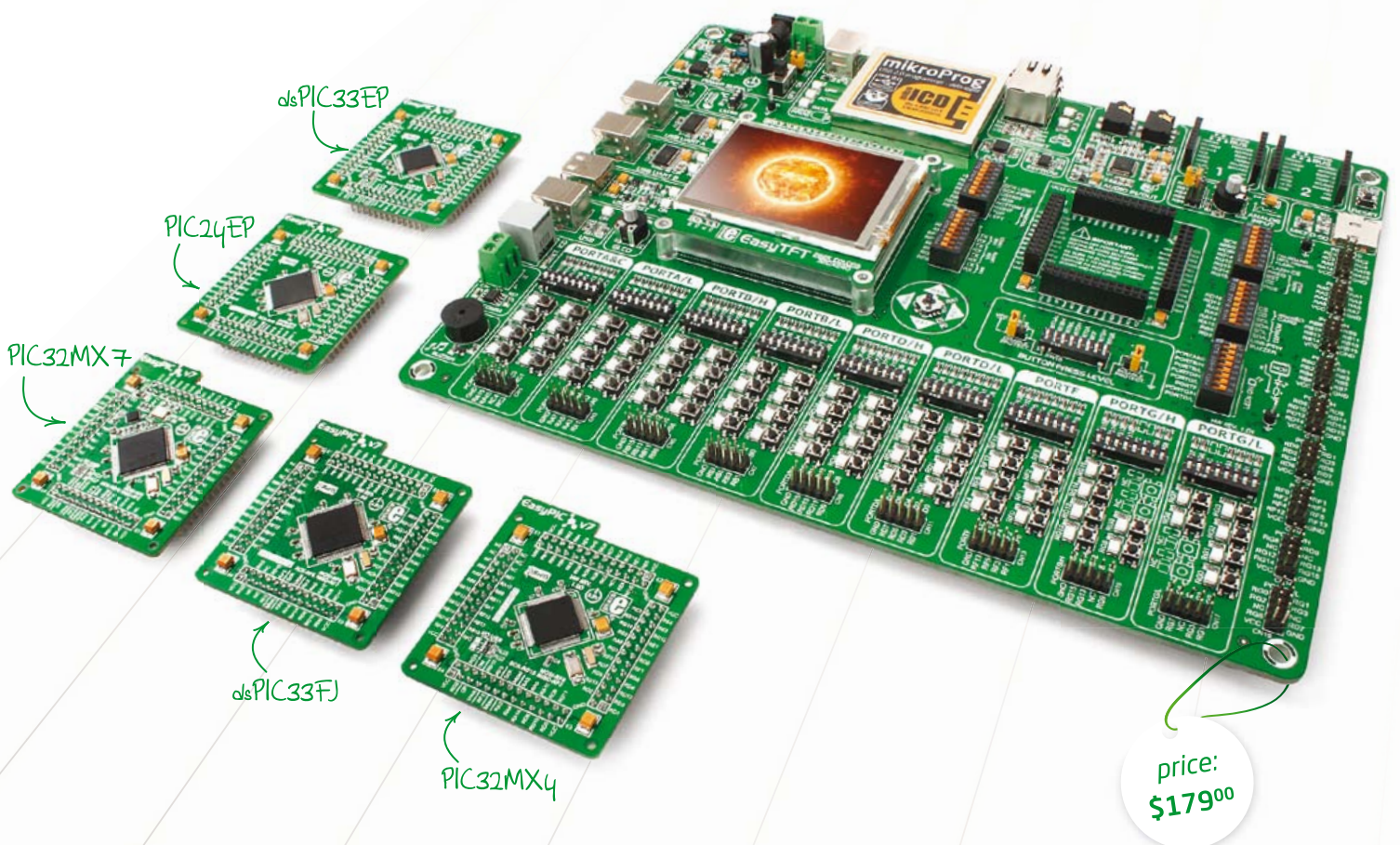
16-bit and 32-bit united

Intelligence is the ability to adapt quickly. Being able to seamlessly change microcontrollers while seeking for the best solution for your project means holding the key to success. That was never easy. Until now.

EasyPIC Fusion™ v7 is the first development board in the world that combines support for three different architectures: dsPIC33®, PIC24® and PIC32®. Just pick a card and get started.

Work smart. Not hard.

EasyPIC Fusion^{v7}





Close encounters of the RFID kind

While engrossed in the making of an e-product it's easy to fall prey to electro-technotunnel vision (ETTV), which, although possibly harmful, also greatly augments the experience of an eye opener every now and then, coming from unexpected corners — from kind non-technical people. The product in this case is a book on RFID (radio frequency identification – device) I am post-editing now for publication in November (I hope). The book goes way beyond the vast body of work already published on practical RFID in Elektor magazine. It is the most exhaustive yet comprehensible work on the technology I've seen thus far, and the authors discuss these wonderful cards and their readers down to the last bit of the checksum system.

The book almost ready editorially, I thought I'd suggest to our book production crew to approach NXP for a supply of a few K of sponsored RFID cards to glue in the books, say two with every copy, for customers to experiment with. Although they liked my suggestion, I wasn't able to return to my ETTV desk without answering questions like "What's all this stuff good for then?" I failed to trigger any enthusiasm about RFID technology and bit masking until I mentioned the system's vulnerability to hacking by digital pickpockets. Like "so if I stand close enough to Donald Trump I can read all of his personal data?" Scam or not, this gave my fellow workers an immediate warm connection to the book and its scope. RFID was no longer nerd stuff or Boys Toys, and the book "great if only I understood the third chapter". A few days later while on my way to Boston to celebrate Circuit Cellar magazine's 25th anniversary I experienced my coat being confiscated by Customs at Amsterdam airport. It produced an odd blip on their Xray (I think) machines, and a kind but strict officer told me "Sir it's an anomalous sub-frequency response that needs investigation" and I could collect my coat after the return flight. As it turned out, an active "quality control" RFID tag got sewn into the coat lining; no harm done except I was without a coat in New England early October weather. Vulnerable, for sure, but not to hacking.

Happy reading,

Jan Buiting, Managing Editor

elektor

6 Colophon

Who's who at Elektor.

8 Industry

A monthly roundup of all the latest in electronics land.

10 Electrical Storm Proximity Measurement

Design considerations behind a lightning detector IC recently introduced by ams.

14 Welcome in elektor-dot-labs

Join the welcome tour of the Elektor laboratories. Who's doing what there, why, and how?

16 Farewell 7805 & 7905

Grateful for their accomplishment, we now institutionalize the 7805 and 7905, and welcome their switch-mode replacements.

22 Embedded Linux Made Easy (5)

This month we set about doing A-D conversion, I/O and generally getting our Linux board onto the Internet highway.

30 Nixie VU Meter

Your high end tube amplifier really deserves this warm & retro power indicator built with two IN-g bargraph Nixie tubes.

36 A Library for the ElektorBus

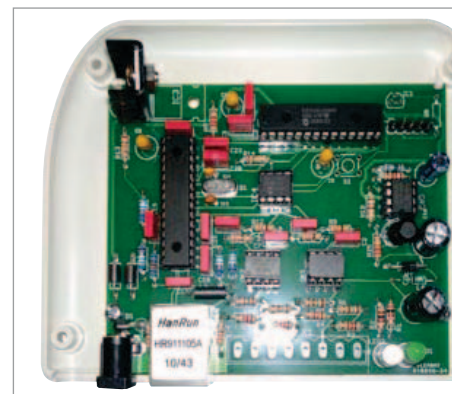
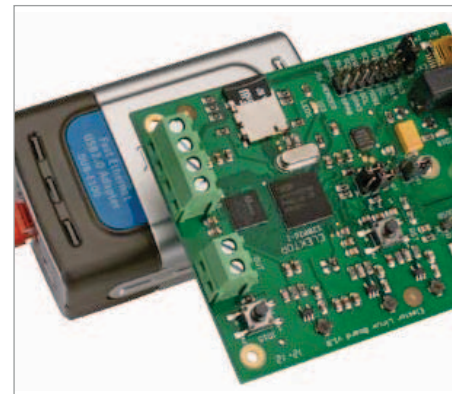
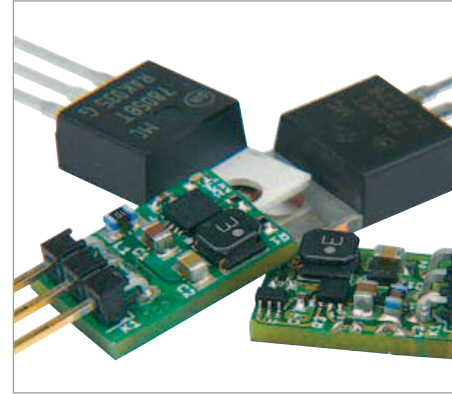
The ElektorBus being a platform it can benefit from a library of code pieces for everyone to set up his own comms methods.

44 E-Labs Inside: Budget Rigol Spectrum Analyzer

This spectrum analyzer with built-in tracking analyzer caught our attention mainly because of its stunning price/performance ratio.

45 E-Labs Inside: Beep-beep-beep who's there?

Meanwhile in the labs, trying hard to identify a mysterious source of signals in the 433 MHz ISM band.



CONTENTS

Volume 4
November 2012
no. 47



16 Farewell 7805 & 7905

Linear voltage regulators of all shapes and sizes are used in countless electronic circuits. The classic 7805 and 7905 devices are not exactly champions in energy efficiency, due to their linear mode of operation. All superfluous power between the input and output pins of these regulators is simply converted into heat. Surely we can do better nowadays — more energy efficient and greener? That's what this article is about. The message is clear: down with power dissipation, up with switching!



22 Embedded Linux Made Easy (5)

The Elektor Linux board is proving to be a big hit. The board's versatility makes it an ideal learning tool and platform for Linux application development. In this installment we set about reading digital and analog signals then we hook up a network adapter and build a small web server which generates dynamic HTML pages. Using this we can read the status of remote LEDs amongst other things and display the information in a browser.



30 Nixie VU Meter

This circuit is designed to embellish a (tube) amplifier. It supplies delightful light effects rather than a calibrated readout for output power or decibels. At the heart of the circuit is a PSoC from California that's busy all the time driving two Nixie tubes... from the Ukraine.



54 Electricity on the Web

Reducing each individual's carbon footprint has been high on the Global Agenda for the last couple of years. There are many ways to achieve that, but first and foremost you need to know how much electricity is passing through your meter and emptying your wallet 24/7/365. Why not let the worldwide web and some clever technology help you keep tabs on your electricity consumption?

46 Arduino on Course (3a)

We're now ready to use the Arduino Mega ADK board to connect the real world to your Android smartphone — or is it the other way around?

54 Electricity on the Web

Big chance your electricity meter is So-compatible, which is reason enough to build this openSource gateway for the openEnergy platform.

58 SDN — Software Defined NIC

Dirt cheap! A NIC and any old webcam, add a microcontroller and you have an IP network camera.

64 Design Tip: PT100 Simulator.

66 Electronics for Starters (9)

Here we introduce oscillators and their basic design methods using transistors of course.

70 ICL78M6610+PSU & ADE7953 AFE ICs

Raymond's pick of the month in terms of components.

72 Retronics: HP-35: A Pocket Revolution (1972)

It's hard to contend that this product was every engineer's delight when it hit the market 40 years ago.
Series Editor: Jan Buiting

76 Gerard's Columns: TANSTAAFL: NOT

A column or two from our US columnist Gerard Fonte.

78 Hexadoku

Elektor's monthly puzzle with an electronics touch.

84 Coming Attractions

Next month in Elektor magazine.

The Team

Managing Editor:	Jan Buiting (editor@elektor.com)
Deputy Editor:	Thijs Beckers
International Editorial Staff:	Harry Baggen, Eduardo Corral, Wisse Hettinga, Denis Meyer, Jens Nickel, Clemens Valens
Design Staff:	Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser
Membership Manager:	Shannon Barraclough
Graphic Design & Prepress:	Giel Dols, Jeanine Opreij, Mart Schroijen
Online Manager:	Daniëlle Mertens
Managing Director:	Don Akkermans

The Network



Our international teams

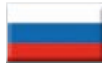
 **United Kingdom**
Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com

 **Spain**
Eduardo Corral
+34 91101 93 95
e.corral@elektor.es

 **India**
Sunil D. Malekar
+91 9833168815
ts@elektor.in

 **USA**
Hugo Vanhaecke
+1 860-875-2199
h.vanhaecke@elektor.com

 **Italy**
Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it

 **Russia**
Nataliya Melnikova
+7 (965) 395 33 36
Elektor.Russia@gmail.com

 **Germany**
Ferdinand te Walvaart
+31 46 4389417
f.tewalvaart@elektor.de

 **Sweden**
Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

 **Turkey**
Zeynep Köksal
+90 532 277 48 26
zkoksal@beti.com.tr

 **France**
Denis Meyer
+31 46 4389435
d.meyer@elektor.fr

 **Brazil**
João Martins
+55 11 4195 0363
joao.martins@editorialbolina.com

 **South Africa**
Johan Dijk
+27 78 2330 694 / +31 6 109 31 926
j.dijk@elektor.com

 **Netherlands**
Harry Baggen
+31 46 4389429
h.baggen@elektor.nl

 **Portugal**
João Martins
+351 21413-1600
joao.martins@editorialbolina.com

 **China**
Cees Baay
+86 21 6445 2811
CeesBaay@gmail.com

No. 47, NOVEMBER 2012

ISSN 1947-3753

Elektor (ISSN 1947-3753) is published monthly (except for one issue in July/August) at \$40 per year, Canada add \$15 per year; by Elektor International Media LLC, 4 Park Street, Vernon, CT 06066, USA. Phone: 860-875-2199, Fax: 860-871-0411. www.elektor.com

Elektor is also published in French, Spanish, German and Dutch. Together with franchised editions the magazine is on circulation in more than 50 countries.

Subscriptions:
Elektor USA, P.O. Box 462228, Escondido, CA 92046.
Phone: 800-269-6301
E-mail: elektor@pcspublink.com
Internet: www.elektor.com

Head Office:

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

US Advertising:

Strategic Media Marketing, Peter Wostrel,
2 Main Street, Gloucester MA 01930 USA.

Membership Counter

We
now have

284600

members
in

83















countries.

**Subscribe now to the world's most comprehensive
magazine on electronics and microcontrollers!**

Sign up at www.elektor.com/members



Supporting Companies

	AP Circuits www.apcircuits.com 71		Front Panel Express www.frontpanelexpress.com 65
	Beta Layout www.pcb-pool.com 29		HuMANDATA www.hdl.co.jp/EL/ 21
	Cadsoft www.element14.com 88		Microchip www.microchip.com 2
	Cleverscope www.cleverscope.com 71		MikroElektronika www.mikroe.com 3
	DesignSpark www.designspark.com 35		Pico www.picotech.com/pco487 63
	DLP Design www.dlpdesign.com 29		Pololu www.pololu.com 69
	ExpressPCB www.expresspcb.com 9		Saelig www.saelig.com 61

Not a supporting company yet?

Contact Peter Wostrel (peter@smmarketing.us, Phone 978-281-7708, Fax 978-281-7706)
to reserve your own space for the next edition of our members' magazine

Phone: 978-281-7708, Fax: 978-281-7706
E-mail: peter@smmarketing.us
Advertising rates and terms available on request.

Copyright Notice

The circuits described in this magazine are for domestic use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, DVDs, software carriers and article texts published in our

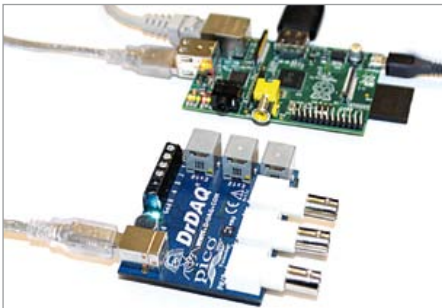
books and magazines (other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning and recording, in whole or in part without prior written permission from the Publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept

responsibility for failing to identify such patent(s) or other protection. The submission of designs or articles implies permission to the Publisher to alter the text and design, and to use the contents in other Elektor International Media publications and activities. The Publisher cannot guarantee to return any material submitted.

© Elektor International Media b.v. 2012 Printed in the USA

DrDAQ samples some Raspberry Pi

Pico Technology's DrDAQ compact single-board data logger adds 17 I/O channels to your Raspberry Pi. Now your Linux application can have access to a 100 kHz oscilloscope, arbitrary waveform generator, 4 digital I/Os (2 with pulse-counting input and PWM output), 24-bit RGB LED, built-in light sensor, temperature sensor, microphone and sound level sensor, resistance measuring input and pH/redox sensor input. There are also 3 inputs for Pico's own sensors or for custom devices that you can build yourself. DrDAQ requires just a single USB connection for power and data.



When connected to the Raspberry Pi single-board computer, DrDAQ forms a powerful data logging system that can be integrated into your custom Linux application. Pico Technology has released a Debian driver and C++ example code for free download. The example code displays a simple text menu that allows you to capture data, control the digital I/O pins, set up the signal generator and drive the LED.

Download the driver and example code, and read the latest Raspberry Pi news, on the Pico forum at www.picotech.com/support/. You can see the example code in action in a post by Andrew Back on the Design Spark blog (designspark.com).

If you don't already have a DrDAQ data logger, you can buy one from Pico or its distributors for only £99 RRP.

www.picotech.com (120605-1)

Digilent Analog Discovery design kit

Digilent Analog Discovery is a new mixed-signal test and measurement tool that targets the needs of students in undergraduate engineering classes.

Digilent, Inc. announces the release of the



Analog Discovery, a new mixed-signal test and measurement tool that targets the needs of students in undergraduate engineering classes. Developed in conjunction with Analog Devices, the Analog Discovery combines a dual-channel oscilloscope, a two-channel waveform generator, a 16-channel logic analyzer and many other instruments into a USB-powered, low-cost device. Costing less than a textbook and smaller than a deck of cards, the USB-powered Analog Discovery lets students design and test analog and digital circuits in a variety of settings, without the restrictions of working only in the lab.

The Analog Discovery works with the free WaveForms™ software that offers intuitive interfaces to the oscilloscope, waveform generator and other instruments, as well as advanced features like Fourier transforms, Bode plots, and cross-instrument triggering. A parts kit and a large collection of freely-posted teaching and learning materials are also available to help students. The parts kit contains a solderless breadboard, jumper wires, more than 150 passive components, more than 20 IC's from Analog Devices, and an assortment of diodes, transistors, and other devices. Teaching and learning materials include a free on-line text book with exercises and design projects, a series of 40 video lectures suitable for a university-based Circuits class, tutorials, and reference designs and projects.

Studies have shown that students learn more, learn faster, and retain information longer when they have access to portable design kits they can use at home to augment lectures, homework, and lab-based projects. University professors recommend the Analog Discovery Design Kit for aspiring engineering students and often incorporate it into their lessons. The Analog Discovery Design Kit provides a diverse array of opportunities to both professionals and students to decentralize and individualize engineering education.

Digilent Analog Discovery Design Kit is built around 14-bit, 100 MSPS+ data converters from Analog Devices, and offers two 100

MSPS, 5 MHz differential oscilloscopes, two 100 MSPS, 5 MHz waveform generators, two fixed power supplies, 16-channel logic analyzer, 16-channel digital pattern generator, trigger in and trigger out for linking multiple instruments, USB cable for power and data transfer, signal probe wires, and the freely downloadable WaveForms™ Software.

Digilent has also forged an agreement with Designsoft, the producer of the popular TINA circuit design and simulation tool, to offer the student edition of TINA to academic customers for just \$6.95. By combining TINA's powerful design and analysis software with the Analog Discovery, students and professors can build world-class teaching and learning systems for less than the price of a textbook. The kit costs USD 199 (Academic pricing: USD 159), the associated Analog Parts Kit, USD 59 (USD 49 when purchased with Analog Discovery).

www.digilent.com à Analog Design (120605-11)

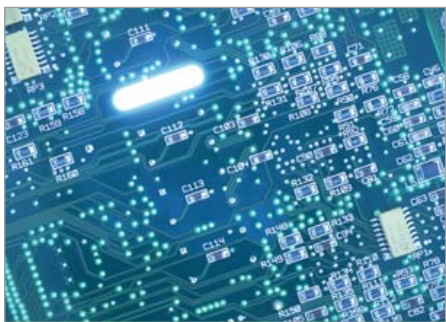
Economical RoHS and ESD compliant solder station

In the area of soldering technology, German specialized distributor reichelt elektronik offers a complete product range of high-performance devices and accompanying accessories. In addition to soldering stations for the lead-free soldering, this also includes soldering fume extraction devices, desoldering and hot air stations, soldering tips, tin solder, flux as well as much more.

The digital soldering station type LF-3000 from Taiwanese manufacturer Xytronic is one of the best-selling products in this segment. From a technical perspective this mainly includes the innovative high frequency heating element for very short heating times and highest regulating precision as demanded by professional users.

With a performance range of up to 90 W and a wide temperature range of 100 to





520°C, the Xytronic soldering station is ideal for a lead-free soldering in accordance with RoHS standards. Due to a zero-voltage control circuit that is electrically isolated from the Ac power line, even electrostatically sensitive components can be soldered safely and damage free. An equipotential bonding socket furthermore enables ESD compliant operation. The soldering iron that is supplied as standards works with 36 V low voltage and is equipped with an internally heated soldering tip with integrated temperature sensor for a high regulating precision of $\pm 3^\circ\text{C}$.

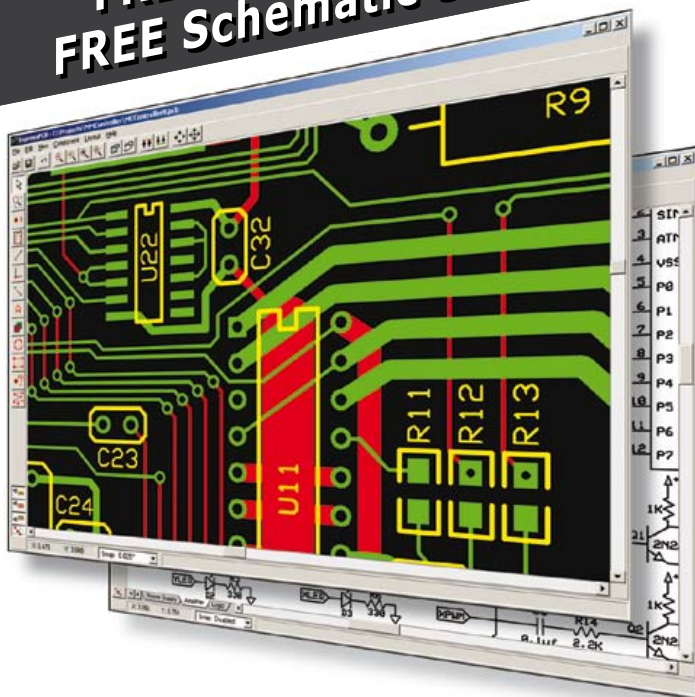
www.reichelt.de (120605-III)

HiPer Simulation™ AFS and enhanced T-Spice HiPer Silicon™ v15.23

Tanner EDA has released version 15.23 of its full-flow analog and mixed-signal design suite: HiPer Silicon™. The addition of HiPer Simulation™ AFS to version 15.23 gives designers added capabilities for front-end design flow, including schematic capture, dual circuit simulators and waveform probing. HiPer Silicon version 15.23 includes Tanner Analog FastSPICE (T-AFS), which integrates the Berkeley Design Automation Analog Fast-SPICE Platform with Tanner EDA's S-Edit™ schematic capture and W-Edit™ waveform analyzer. With HiPer Simulation AFS, two Spice simulators deliver the ultimate in performance and productivity, even for large netlists. T-Spice provides fast, accurate analysis while T-AFS delivers accuracy with run-times 5x to 10x faster than traditional Spice simulators, on a single core. Users can drive the T-AFS simulator directly from S-Edit, getting the speeds and accuracy necessary for nanometer design. Simulation results are displayed automatically in W-Edit for viewing, measuring, and analyzing interactively. For additional information on T-AFS or a

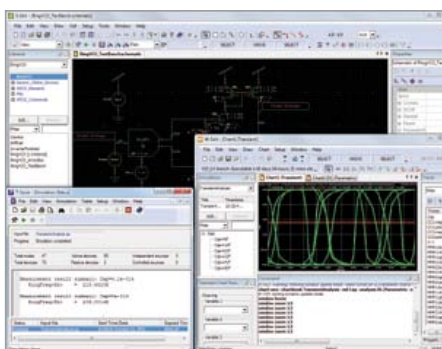
Advertisement

\$51^{For 3} PCBs
FREE Layout Software!
FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

expresspcb.com



product datasheet, please see the website below. As always, Tanner EDA offers a free 30-day evaluation.

Version 15.23 also adds new TCL commands to S-Edit, supporting greater functionality. And T-Spice now supports the HiSIM-HV model. Integration with Berkeley Design Automation transient noise analysis capability allows users to simulate realistic device noise effects for all circuits, especially non-periodic circuits such as sigma-delta ADCs and frac-N PLLs.

www.tannereda.com/tafs (120605-VI)

Electrical Storm Proximity Measurement

By Ruggero Leoncavallo,
System Engineer, ams AG

The US maintains detailed weather and accident reports that provide a long-term record of the risk to human life posed by lightning. Since 1940, approximately 30% more people have been killed by lightning than by hurricanes (according to the US's NOAA National Weather Service). Accurate weather forecasts nowadays help people to take precautions when there is a high risk of storms, and fatalities number fewer than 50 per year in the US on average. But every death is one too many, and in addition the injuries and equipment losses caused by lightning strikes (detailed at www.struckbylightning.org) exact a considerable toll on those unlucky enough to live in a region prone to violent storms. In addition, Chinese scientists have warned that global warming is likely to intensify extreme weather patterns, and severe storms in recent years may be a prelude to this (China Meteorological Administration, July 30, 2007).

Yet the death and damage attributable to lightning is largely avoidable, if people have sufficient warning of an approaching lightning storm. According to the so called "30-30 rule", if the time between the lightning and the thunder is less than 30 seconds, people should get under cover for at least 30 minutes. And if you hear thunder, you are probably already in danger. The human senses are not well equipped to perceive the onset of a lightning storm. A delay of 30 seconds between lightning and thunder corresponds to a distance of about 10 km (6 miles) from the strike (since the speed of sound in air is about 300 m/s); this is about the furthest distance at which humans can hear thunder in a quiet environment without any physical obstacles baffling the sound. In case of physical obstacles and/or a high level of ambient noise (such as traffic, crowds) this distance can be reduced to just a few kilometers.

Exacerbating the problem, it is typical for lightning to strike the ground not vertically but on a diagonal, over a horizontal span

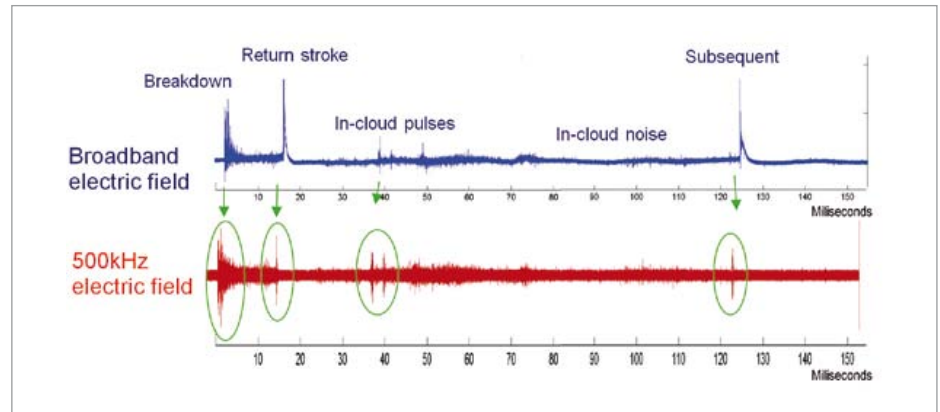


Figure 1. Comparison of broadband and narrowband signals from a single lightning event.

which can stretch as far as 10 km; this is the biggest limitation of the 30-30 rule. Since people can only hear thunder at a maximum distance of 10 km, it is clear that reliance on unaided human senses poses a considerable risk to both life and property.

Electromagnetic propagation (EMP) in lightning

As early as the 19th century, Alexander Stepanovic Popov noticed that it was possible to detect lightning using a simple AM radio receiver; this was the first electrical system capable of predicting a storm.

In fact, lightning emits electromagnetic energy from very low frequencies up to X-ray bands. The intensity of the EMP phenomenon displays $1/f$ behavior: the emissions are at their strongest at low kHz frequencies, and weaken as frequency rises. Using a simple system with an amplifier, down-mixer and a low-pass filter, Popov was able to hear the signal produced by lightning.

Similar technology is still in use today in personal lightning detectors (or 'lightning counters') sold to consumers. Although the American Meteorological Society does not recognize the reliability or value of these portable devices, lightning counters can, in the right conditions, detect lightning within a small area. But these rudimentary devices are of limited use, since they can

not estimate the distance from the head of the storm, nor can they reliably differentiate lightning from sources of interference such as microwave ovens, fluorescent ballasts, motors, car engines and camera flashes. Furthermore, as those systems are based on discrete solutions they are not optimized for current consumption and the battery life is limited to a couple of weeks.

What is needed to provide consumers with a reliable and timely warning to protect themselves is a personal device that accurately estimates the distance from a storm over a distance of 30 km (19 miles) or more, and that reliably distinguishes lightning signals from other sources of EMP.

Use of narrowband systems in lightning detection

There are two kinds of lightning: cloud-to-ground and intra-cloud. In terms of electromagnetic analysis, the huge currents generated in storms produce wideband signals across a large spectrum. Monitoring such a wide frequency range is next to impossible with a portable, consumer device. Fortunately, since Popov's experiments it has been known that a narrowband system can pick up signals from lightning. But how accurate is this narrowband signal?

In fact, lightning is a complex combination of several different events: breakdown, return strike, in-cloud activity, and

Using simple narrowband RF technique

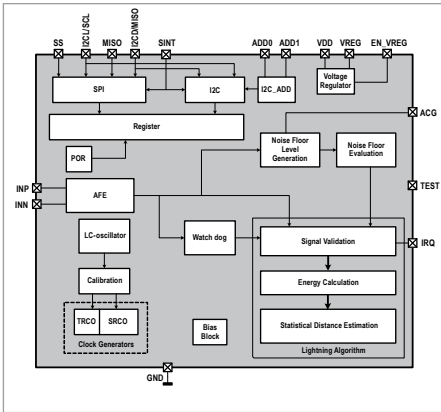


Figure 2. Block diagram of AS3935 Storm Detector.

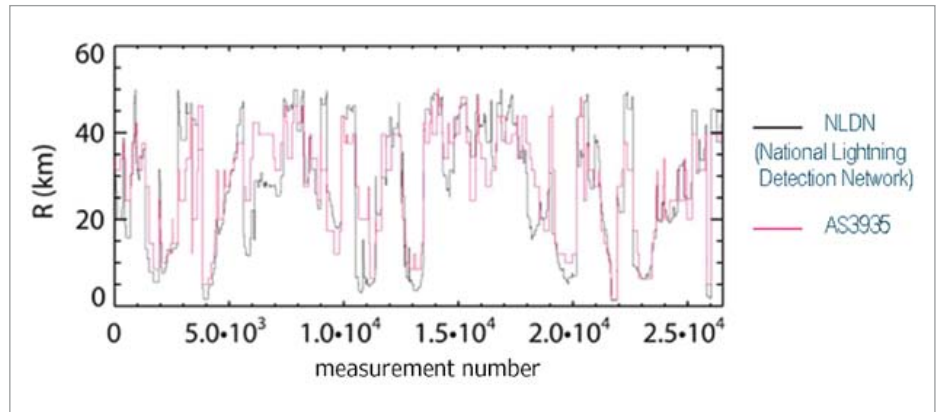


Figure 3. Comparison of lightning monitoring outputs from the AS3935 and the gold-standard NLDN data. This shows that the output from the AS3935 closely matches that of the NLDN's sophisticated radar-based systems. The horizontal axis represents the number of lightning strikes.

subsequent strike. The scientific literature (Le Vine, D.M., 'Review of Measurements of the RF Spectrum of Radiation from Lightning', 1987) establishes that it is possible to detect lightning with a narrowband system, even if some detail in the shape of the signal is lost. As **Figure 1** shows, there is a close match between the broadband electric field and the signal picked up with a narrowband receiver at 500 kHz.

Le Vine's paper also shows that lightning's emissions peak at around 5 kHz and drop as a function of $1/f$. So if the center frequency of a narrowband receiver is set too high, the received signal strength falls to the point at which it is hard to distinguish it from other, interfering sources of EMP. Signal strength is much greater at a low frequency; at very low frequencies, however, the antenna becomes too bulky to be accommodated in a portable device. There is therefore a trade-off between signal strength and device size, with the most suitable frequencies being between a few hundred kHz up to a few MHz.

Pattern matching software produces reliable distance measurements

In line with Popov's findings, then, a low frequency receiver can sense the emissions from lightning strikes. But the bigger challenge in producing a reliable and useful

lightning detector is:

- to reject signals from other emitters;
- to estimate accurately the distance from a storm.

Now, ams (formerly: 'austriamicrosystems') have developed suitable technology which is implemented in its AS3935 Storm Detector IC. This technology, which is effective for both cloud-to-ground and intra-cloud lightning, uses algorithms which analyze incoming signals and compare their shape to the typical shape of a lightning strike's waveform. Exhaustive effort has gone into tuning the algorithm so that it provides an excellent balance between rejection of interference from other emitters, and recognition of genuine signals emitted by lightning. A dedicated hardwired algorithm implemented in the AS3935 is also able to derive accurate distance estimations from analysis of the energy of the signal detected by the IC's RF front end.

Figure 2 shows a block diagram of the AS3935. Like the Popov system, it monitors the LF bands (500 kHz – 2 MHz) to detect the strong $1/f$ signature characteristic of lightning. The system includes an Analog Front-End (AFE), which amplifies the input signal picked up by the antenna and transfers it to the baseband, while filtering it for back-end numerical elaboration. The

lightning algorithm block consists of three stages: signal validation, energy calculation and statistical distance estimation.

The first block checks the pattern of the incoming signal and rejects man-made noise. As shown in **Figure 1**, a typical lightning signal goes high very fast and declines smoothly after the peak. The flexible algorithm in the AS3935 allows optimization by the end user of the trade-off between lightning detection effectiveness on the one hand and disturber rejection on the other. If the received signal is judged to be lightning, the second block performs an energy calculation. This calculation is then analyzed in the last block, which makes an assessment of the distance to the head of the storm based on data collected during the whole storm. The distance estimation is provided in kilometers with a variable resolution between 1 km (0.8 miles) and 4 km (2.5 miles). (The estimate becomes more accurate as the storm draws closer to the instrument.)

The performance of the AS3935 has been tested by the Florida Institute of Technology at Melbourne, US and compared to the official lightning data provided by the National Lightning Detection Network (NLDN) in the US, which is regarded as the gold standard for lightning monitoring (see **Figure 3**).

(120541)



Elektor is poised for the future and ready to rejuvenate itself, starting next month. Our rock solid foundation will remain the level of respect for what the joint team has achieved over the past years. Our readers, website visitors and clients can continue to expect service, reliability and expertise. Elektor holds its position as one of the leading portals, platforms and forums for electronic engineers all over the globe. The means to reach our readers, as well as the accessibility of our services, are upped to match today's requirements and beyond.

By the Elektor Editorial Team:

Jan Buiting (Elektor UK & USA), Jens Nickel (Elektor Germany), Harry Baggen (Elektor Netherlands) and Denis Meyer (Elektor France)

Elektor today is much more than a just these printed pages. Still, ink is where we started from. Many members of the Elektor community consider themselves a subscriber first and foremost. Some have been with us for ages, others for just a few months.

Many of you buy the magazine from bookstores if a project or article is announced that matches their particular interest. Others may remember Elektor magazine as their vehicle to a professional career in electronics many years ago. We are equally happy to guide readers if they are students, or inform and support them in their professional capacity.

Along the way Elektor's services have expanded vastly, by popular request or simply prompted by time. We are sure you have seen our website at least once, or bought a PCB or kit from our shop. We would not be surprised to see at least one copy of 30x Circuits on your bookshelf, or find your name on the list of participants to one of our webinars or international design contests. We may even have published one of your projects for all & sundry to enjoy and learn from.

Meanwhile the electronics industry has discovered what's cooking at Elektor. Manufacturers are actively seeking to contact you by way of us, for beta testing, sampling, competitions and information exchange. In some case, even for job opportunities.

Increasingly electronics enthusiasts get to know Elektor by way of media other than the magazine that has been the foundation of our name and fame.

You may not realize it, but you are part of a 300 Kperson network spanning 80 countries. Our websites welcome visitors from 243 countries.

Summarizing, Elektor has developed into an international community of electronics aficionados. Indeed, way beyond the limits of printed paper. Consequently we will no longer use the terms "reader" or "subscriber" and switch to "member" instead. To underscore that the times are well past when we reached our members with a printed-only product, and that the Internet offers fantastic opportunities for us to employ for our services, we are introducing the Elektor.COMMUNITY.

Here's the bill of materials and the schematic!

elektor  **labs**

A new dedicated website called Elektor.LABS is available where you can leave proposals for projects, present your own project to peers, follow its progress, as well as reel in finished and tried & tested stuff. A beta version of this website has been online for some time under the name Projects. Elektor.LABS is Elektor's throbbing heart, where projects are being proposed and get developed, by you and the community — where knowledge is shared, enjoyed and acquired.

elektor  **community**

At www.elektor.com a single portal (homepage, central website) exists called Elektor.COMMUNITY from which you can access our shop Elektor.STORE, our archive Elektor.FILES and a number of FORUMS. Use this portal to trace old articles and components, order

(WAY) BEYOND PRINTED PAPER



books or kits, extend your membership, and more!
As before, you can access all sections directly too.

elektor magazine

As of now the printed magazine will be called Elektor.MAGAZINE, appearing 10 times per year. Although it has a restyled layout, a new logo and new sections, Elektor.MAGAZINE continues to bring you the best-of-market number of circuits and projects. In addition to the extra thick Projects Generator edition to cover the summer months, you will receive another 'jumbo' magazine at the start of the year. In green fashion the magazine is also downloadable in its entirety.

elektor post

The scope of our new mail service called Elektor.POST is beyond that of the current E-weekly newsletter, which remains in place to deliver newsletters on a regular basis, videos, offers and invitations to webinars and competitions. Elektor.POST now adds an exclusive two-weekly project that will automatically reach you by email. That's 25 extra projects per year, delivered by email.

For whom?

The traditional subscription is replaced by a membership, comprising:

- Eight editions per year of the restyled and restructured magazine with the traditional number of circuits and projects enhanced and extended with new sections and elements.
- Two editions per year of the 'jumbo' version of Elektor.
- A GOLDCARD membership card that comes as a USB card loaded with a solid amount of goodies
- Elektor Year Volume stored on your membership card.
- A minimum of 10 percent discount on all products and services in Elektor.STORE.
- Printed circuit board availability for every Elektor project; with a discount of 10 percent minimum for members.
- Direct access to Elektor.LABS
- Elektor.POST sent to your email account.


Alternatively, we offer the all-paperless GREENCARD membership card, which delivers all products and services, including Elektor.MAGAZINE, online only.

If your current subscription is about to expire, you will receive a personal message in which all changes and options are explained in detail. With that message comes an offer for your new membership that's hard to refuse.

Welcome in elektor labs !

Hello! Nice to see so many of you here for this tour of Elektor.LABS. Today .LABS is open to everyone, but if you want to come back later, you will need a membership card or an invitation. Then you can visit Elektor.LABS whenever you like. My name is CPV and I will be your guide.

Please follow me.

Elektor  LABS is the place where the real electronics action is. Forget accounting or management, .LABS, or dot-labs as we pronounce it, is the throbbing heart of your favorite electronics community better known as Elektor. The people who work in .LABS are not only as attractive as our secretaries, but they also know way more about electronics.

Here on my right for instance (your left) we have Ton G. as in "Gee Ton, that sounds amazing!" Although he doesn't like to brag about it, Ton is a highly gifted audio amplifier designer and he literally overflows with knowledge and experience. If you have a question, ask Ton. We do too.

Over there in the far corner we have Luc or Lucky Luc. Please all say hi to Luc. [Hi Luc!] We call him Lucky Luc because he always gets his projects working. Don't ask me how he does it, he just does it. Luc is a kind of quiet guy full of hidden depths littered with electronics nuggets. If you talk to him gently he will give you some of them.





Then we have Raymond or Raving Ray as we like to say. Contrary to Ton and Luc, Raymond is always raving about new technologies and chips. He sucks up product information like a sponge and spills wild ideas like a regular BP oil well in the Gulf of Mexico. Looking for a new field-programmable doohickey? Ask Raving Ray.





www.elektor-labs.com

Now if you look this way you will see Sunil, Sunil and Sunil. They are from India. Well, actually, they are *in* India. Our Indian S3-Express produces electronics projects with a slight hint of curry. That's how we like it, spicy electronics to keep things interesting. Since you now have .LABS access you can talk to one of the Sunils whenever you like, just leave them a comment on the website.


Sorry, what was that? Ah, you thought that Elektor  LABS was just an obscure lab hidden away in a castle? You couldn't be further from the truth! Elektor  LABS is *many* obscure labs hidden away in many garages, bedrooms, attics, companies, schools, universities and you name it all over the world. Elektor  LABS is not geographically limited, it is worldwide. You see, there we have Ivan from St. Petersburg, Russia, who is a specialist in microcontroller-based sports applications. And over there are Gina and Nina from Lima, Chile, who are very much into wearable electronics. They just did a poncho that... excuse me? Lima is in Peru? Oh, OK, and you are? Paikan from Japan. Hi Paikan. Besides the capitals of the world what are your other interests? Anything electronic? You would like some help with the development of a wind speed meter without moving parts? Excellent. For that you should go to the Elektor  LABS website and publish your project or proposal. Write down a short description of what you want to do or what you are doing so that other people can understand it and then can respond to it. Don't forget to add illustrations — people like photos and drawings and they do make things more attractive.

The Elektor  LABS website is designed to help bring people together.



Members with similar interests or that happen to know how to solve your problem can contribute to your project. Together you can create, develop and finish simple or complicated designs. Did you browse the .LABS website? I seem to

remember having seen a solid-state wind speed and direction meter. You should check it out; it is a really cool project. And there are many more interesting projects on all kinds of subjects.



Visitors of the Elektor  LABS website can also rate projects. All you have to do is click a star on the project's main page. It is crucial to rate projects, because popular projects will get picked up by our

editors who will eventually turn it into a magazine article with the project owner's help. And you know the great thing about that? If your project is published in Elektor Magazine you get paid for it! Not only will you see your name in print in the renowned Elektor Magazine, you will also be able to take your partner out for dinner to celebrate it without worrying about the exorbitant prices in the wine list! You can even buy yourself some cool shades and avoid being recognized. Get published, get famous, get rich!







But not only projects with five-star ratings can make it into the magazine. Not everybody likes Lady Gaga, the Spice Girls or the Beatles, right? Some people like more obscure stuff. So if our editors think that a project is really interesting even though it doesn't attract that much attention, they might still decide to take it to the magazine. Anyone can



become an author, anyone can make money from his or her passion, anyone can get elektorized. Elektor  LABS is not just about popular electronics, Elektor  LABS is about electronics in all its shapes, flavors and appearances.

Some of these shapes and flavors are electronics contests and product giveaways. As you may have noticed from our magazine and websites, Elektor organizes special activities on a regular basis in which you can participate and win prizes. And now that we are on the subject, currently we have a giveaway going on in the PSoCaMorph analog music synthesizer project. Here active contributors can win a PSoC 5 Development Kit just by being clever and by helping the project forward. By the way, this project is moderated by the famous filter expert Kendall Castor-Perry. If music synthesis is your thing, you should definitely check out the PSoCaMorph fast. Be clever and win!

This ends our guided tour of Elektor  LABS. As you have seen, Elektor  LABS is much more than just an electronics lab; it is a way to get the most out of your passion. When you have your own key to Elektor  LABS you can stay or come back whenever you feel like it. If you want to discuss something with one of the people mentioned before, or with someone else active on the website, just post a contribution or a comment below the project that interests you. We definitely encourage you to do so.

Oh, one last thing. When you leave, don't worry about the lights or the solder station. Elektor  LABS is open 24/7 all year round.

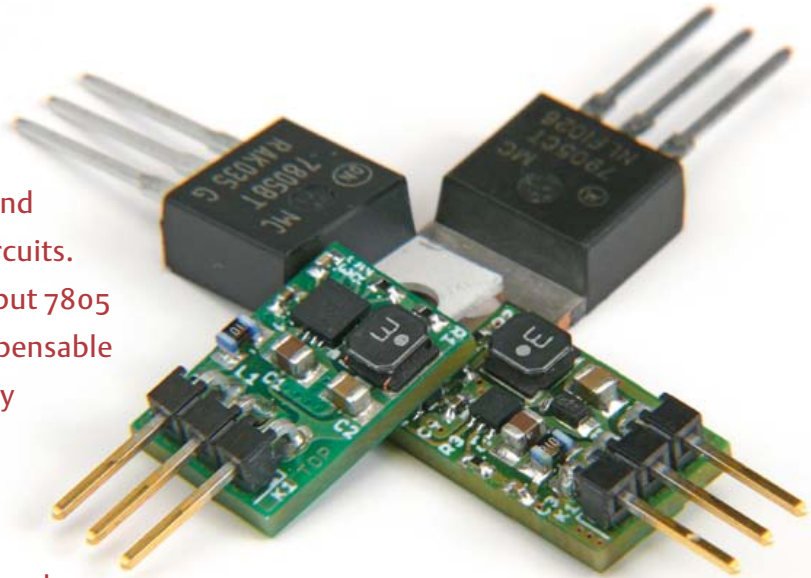
(120608)

Farewell 7805 & 7905

Switch-mode replacements do a much better job

By Raymond Vermeulen (Elektor Labs)

Linear voltage regulators of all shapes and sizes are used in countless electronic circuits. Classic devices such as the positive-output 7805 and the negative-output 7905 are indispensable in most designs, but they are not exactly champions in energy efficiency, due to their linear mode of operation. All superfluous power between the input and output pins of these regulators is simply converted into heat. Surely we can do better nowadays — more energy efficient and greener? That's what this article is about. The message is clear: down with power dissipation, up with switching!



The inception of this project came one day when my colleague Ton Giesberts dropped by my desk with the idea of making a switch-mode version of the widely used 7805 voltage regulator. My first response was to point out existing commercial solutions, but they were either a bit too large or could supply only a small amount of current. I thought it must be possible to do better, and the result is the switch-mode positive voltage regulator described in this article. It is based on a standard buck converter design, cast into a 7805-compatible package.

Shortly after this, our Dutch editor Harry Baggen had the idea of also making a matching switch-mode negative-voltage version. Although I was aware of positive-to-negative voltage regulators, regulating a negative voltage to something less negative was a different story. After a bit of research on the Internet I turned up an old National Semiconductor application note describing the 'negative buck' topology. The negative voltage regulator is based on this operating principle.

Positive voltage regulator

As previously mentioned, this circuit uses a buck converter, also known as step-down converter. This is a switching circuit based on a diode, an inductor and a capacitor (**Figure 1**). When the switch is closed, energy is stored in the inductor. When the switch is opened, the energy in the inductor is transferred through the diode to the capacitor and the connected load. The output voltage depends on the duty cycle (on/off ratio) of the switch. The longer the on time, the closer the output voltage approaches the input voltage. With this simple design, the output voltage depends on the load connected to the circuit. To prevent this, practical switch-mode voltage regulators employ feedback. This is usually done by comparing the output voltage with the desired voltage and using the difference to adjust the duty cycle.

Buck converters are widely used in modern electronic equipment, and nowadays many semiconductor device manufacturers sell ICs specifically designed for this purpose. They contain the necessary control and

protection circuitry in addition to the switching components. For our application we chose the Texas Instruments TPS62150, a synchronous step-down converter that can supply 1 A and operates at a fairly high switching frequency of 1.25 MHz. This allows the dimensions of the peripheral components to be kept fairly small. **Figure 2** shows the internal block diagram of this IC. If you now look at the schematic diagram of our 7805 replacement in **Figure 3**, you will see that the diode of the basic circuit in **Figure 1** has been replaced by an internal MOSFET in order to minimize diode losses. There's not much more that needs to be said about the circuit, since nearly everything aside from the inductor and a few resistors and capacitors is integrated in the IC. An extra LC filter stage (L1/C1) is included on the input to suppress any noise that might be present. Resistor R3 is included to keep the filter from oscillating, which is theoretically possible but actually unlikely because the equivalent series resistances of L1 and C1 should be large enough to prevent oscillation. As you can see from the block diagram, the buck converter IC contains several control

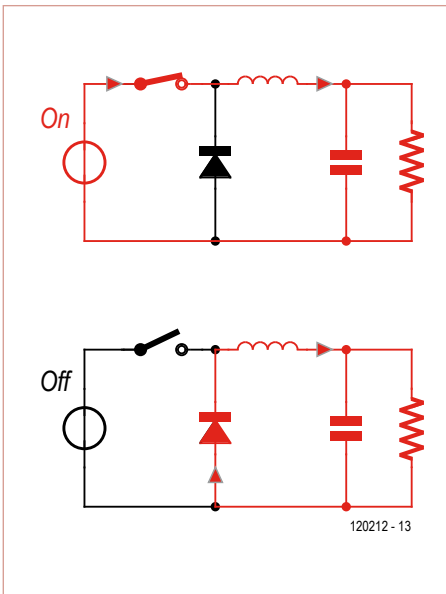


Figure 1. Block diagram of the buck converter topology.

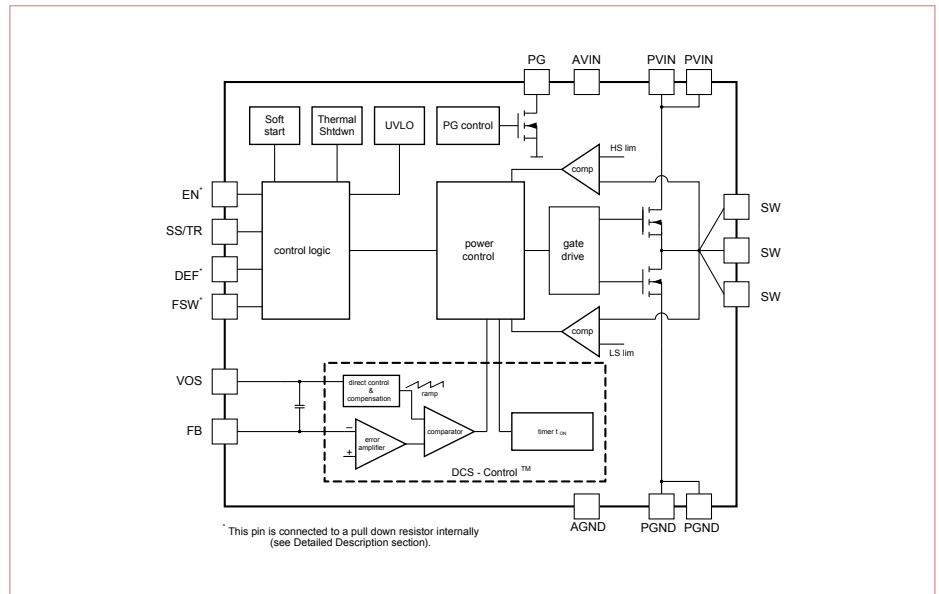


Figure 2. Internal block diagram of the TPS62150.

circuits that precisely regulate the output voltage, so that only minimal undershoots or overshoots occur in response to fast load changes ('load steps').

The components have been specifically chosen with an eye to the size of the overall circuit. The dimensions of the TO-220 package are rather loosely specified and vary from one make to the next. We chose values roughly in the middle for our design, with a package size of $15.5 \times 10.1 \times 4.75$ mm. Here the space normally occupied by the metal tab is part of the circuit board. This means that the device cannot be screwed to a heat sink like a normal 7805, but that is not necessary due to the low power dissipation of the switch-mode voltage regulator. Alternative resistor values for output voltages other than 5 V are listed in **Table 1**.

The PCB layout is an important factor for switch-mode voltage regulators. **Figure 4** shows the PCB layout for the positive voltage regulator. The basic rules generally applicable to PCBs of this sort are that circuit loops with large current variations should be kept as small as possible, while circuit loops with large voltage variations should have as little copper area as possible. Since our device must anyhow be made small, these rules fit nicely with our mechanical constraints.

In terms of performance, the switch-mode regulator can hold its own against a normal 7805: the maximum input voltage is 17 V

Technical specifications

Positive voltage regulator

- Output voltage +5 V (adjustable with two resistors)
- Input voltage range 6 to 17 V
- Maximum output current 1 A
- Maximum output ripple 80 mV_{pp} at 17 V in and 1 A out
- Dimensions and pinout identical to standard 7805

Negative voltage regulator

- Output voltage -5 V (adjustable with two resistors)
- Input voltage range -6 to -17 V
- Maximum output current 0.85 A
- Maximum output ripple 36 mV_{pp} at -17 V in and 0.85 A out
- Dimensions and pinout identical to standard 7905

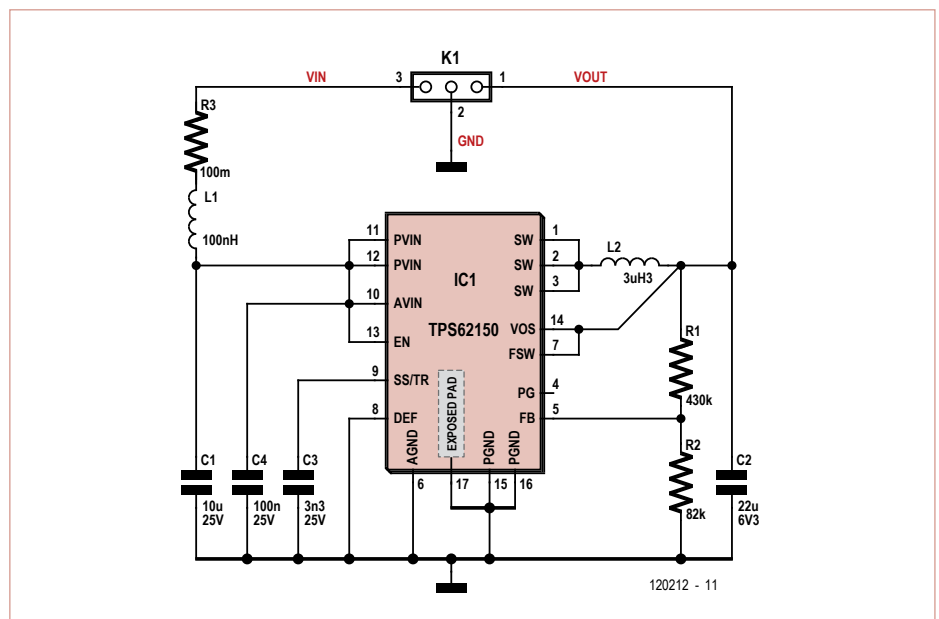


Figure 3. Schematic diagram of the positive voltage regulator.

COMPONENT LIST

Positive voltage regulator

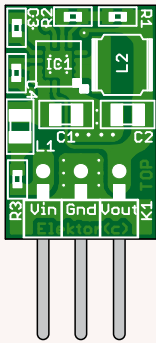


Figure 4. Double-sided PCB layout for the positive voltage regulator. (200%)

Resistors

R1 = 430kΩ 1%/0.1 W, SMD 0402
 R2 = 82kΩ 1%/0.1 W, SMD 0402
 R3 = 0.1Ω 1%/0.1 W, SMD 0402

Capacitors

C1 = 10μF 25 V, X5R, SMD 0805 (e.g. TMK212BBJ106KG-T)
 C2 = 22μF 6.3 V, X5R, SMD 0805 (e.g. JMK212BJ226MG-T)
 C3 = 3.3nF 25V, X7R, SMD 0402
 C4 = 100nF 25V, X5R, SMD 0402

Inductors

L1 = 100nH, SMD 0805 (e.g. MLZ2012DR10D)
 L2 = 3.3μH, 4x4 mm (e.g. NRS4018T3R3MDGJ)

Table 1. Resistance values for various output voltages (positive voltage regulator).

U_{out}	R1	R2
5.0 V	430 k	82 k
3.3 V	256 k	82 k
2.5 V	174 k	82 k
1.8 V	102 k	82 k
1.5 V	72 k	82 k
1.2 V	41 k	82 k

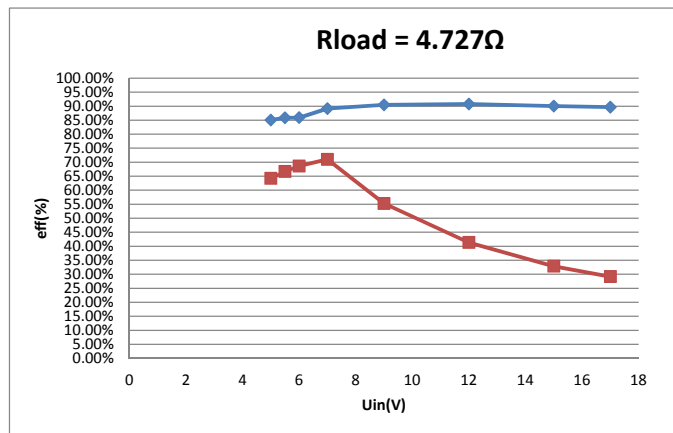
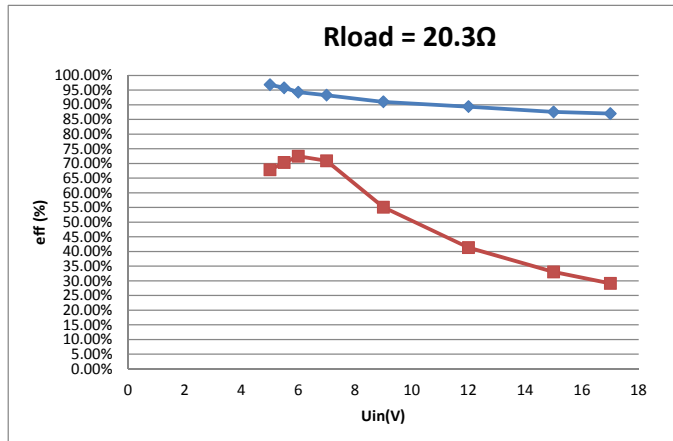


Figure 5. Measured efficiency of the positive voltage regulator at 250 mA (a) and 1 A (b) output current. Blue = switch-mode voltage regulator; red = standard 7805.

and the output can certainly deliver 1 A. The standard capacitors usually present with a 7805 configuration can be left as they are with an existing PCB design. However, you can omit them in new designs without any problems.

Figure 5 shows the efficiency of the switch-mode positive voltage regulator at various

input voltages and output currents. At low input voltages the efficiency is around 95% for output currents up to 250 mA. It drops to around 85% at the maximum allowable input voltage. At the maximum output current level, the efficiency is roughly 90% over virtually the entire input voltage range.

The response of this circuit to a sudden change in load, or load step, is especially interesting. For this we put together a test setup that switches between 10% and 85% of the maximum load at a rate of 50 kHz with 50% duty cycle. The measured output voltage is shown in Figure 6. The large spike is mainly due to the location of the test probe and the leads. The test setup does not represent a realistic load situation, but instead a sort of worst case. In normal use the spike at the output will be much smaller.



Figure 6. Response to pulsed load.

Negative voltage regulator

With the positive voltage regulator the circuit design and the component values are fairly close to the description in the

data sheet for the IC used in the circuit, but things are distinctly different with the negative voltage regulator. There are lots of ICs available for switch-mode power supplies that boost or buck the input voltage or invert the polarity, but there are no switch-mode regulator ICs available that convert a negative voltage to a lower negative voltage. However, as mentioned earlier we found a description on the Internet for a circuit that can handle this task.

It is based on a boost converter connected 'the wrong way round', in what is called a negative buck converter topology. If you take a close look at the circuit diagram of the negative voltage regulator in **Figure 8**, you may think that the pin designations are mixed up, since circuit ground is connected to the V_{in} pin of the IC and the negative input voltage is connected to the GND pin. This allows a negative input voltage to be converted to a less negative regulated output voltage.

However, we have a problem here with the feedback circuit, since we have to reference the feedback signal to the GND pin of the IC in this unusual configuration. If we used a normal voltage divider for this, the resistor ratio would depend on the input voltage relative to the V_{CC} pin. This means that the circuit would only work properly at a specific input voltage. We solved this problem by using a MAX4073 current shunt monitor IC for the feedback signal. it converts the output voltage level relative to GND to a voltage referenced to V_{in} .

For the boost converter IC we looked for a device that could handle a high input voltage but was nevertheless small enough for this application. We ultimately chose the Texas Instruments TPS61170 (**Figure 7**). It can handle up to 20 V between V_{in} and GND. Despite the unusual arrangement, virtually the same formulas can be used here as for a standard buck converter. The input and output LC filters from the positive voltage regulator design can also be reused for the negative voltage regulator.

All in all, this yields a design that is small enough to fit on a PCB with the same

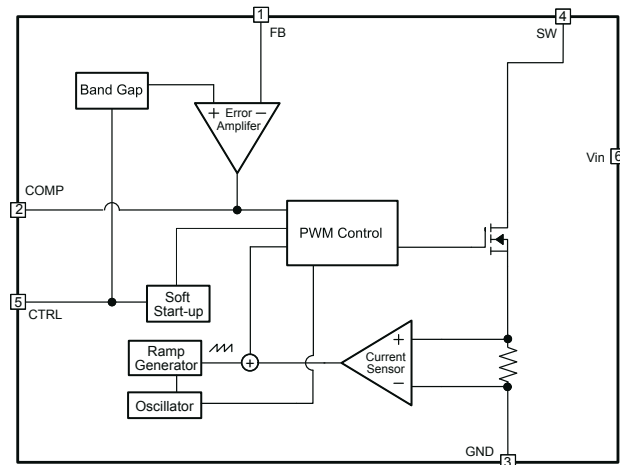


Figure 7. Internal block diagram of the TPS61170.

dimensions as a standard 7905 IC (see **Figure 9**). Here as well, other output voltages can be obtained by adjusting the resistor values for the voltage divider (see **Table 2**).

The performance of this unusual arrangement is very good. As can be seen from **Figure 10**, the efficiency is a good deal

better than with a normal 7905. However, the maximum output current is lower than with the positive voltage regulator. This is due to the built-in current limiting of the TPS61170. In practice, it turns out that the current limit of the negative voltage regulator is somewhat temperature dependent. When it's on the edge of limiting and you

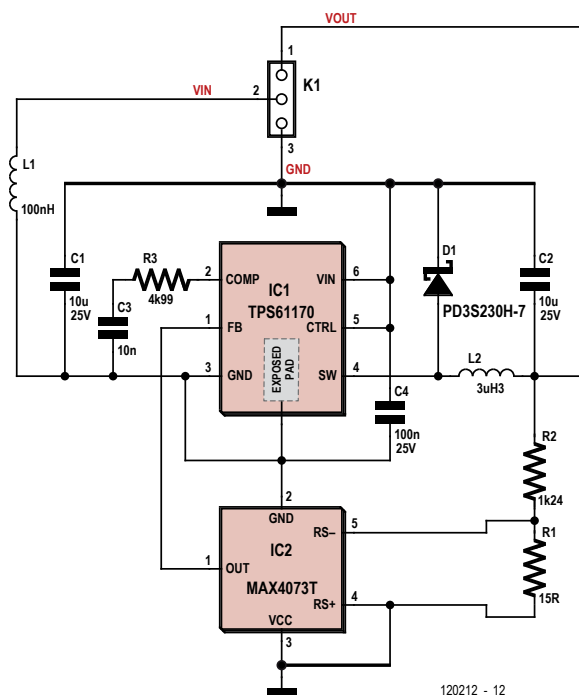


Figure 8. Schematic diagram of the negative voltage regulator.

COMPONENT LIST

Negative voltage regulator

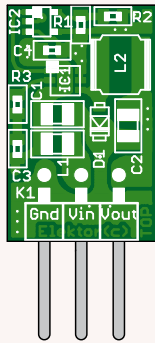


Figure 9. Double-sided PCB layout for the negative voltage regulator. (200%)

Resistors

R1 = 15 Ω 1%/0.1 W, SMD 0402
 R2 = 1.24k 1%/0.1 W, SMD 0402
 R3 = 4.99k 1%/0.1 W, SMD 0402

Capacitors

C1 = 10μF 25V, X5R, SMD 0805 e.g. TMK212BBJ106KG-T
 C2 = 22μF 6.3V, X5R, SMD 0805 e.g. JMK212BJ226MG-T
 C3 = 10nF 25V, X5R, SMD 0402
 C4 = 100nF 25V, X5R, SMD 0402

Inductors

L1 = 100nH, SMD 0805, e.g. MLZ2012DR10D
 L2 = 3.3μH, 4x4 mm, e.g. NRS4018T3R3MDGJ

Semiconductors

IC1 = TPS61170DRVR boost-converter, 6-SON
 IC2 = MAX4073TAXK+T current-sense amplifier, SC-70-5
 D1 = PD3S230H-7 Schottky-diode, PowerDI323

Miscellaneous

K1 = 3-pin pinheader, right angled, 0.1" pitch (2.54mm)
 PCB # 120212-2 (www.elektor.com/120212)

Table 2. Resistance values for various output voltages (negative voltage regulator).

U _{uit}	R1	R2
-5.0 V	15 Ω	1.24 kΩ
-3.3 V	23.2 Ω	1.24 kΩ
-2.5 V	30.1 Ω	1.24 kΩ
-1.8 V	42.2 Ω	1.24 kΩ
-1.5 V	51 Ω	1.24 kΩ
-1.2 V	63.4 Ω	1.24 kΩ

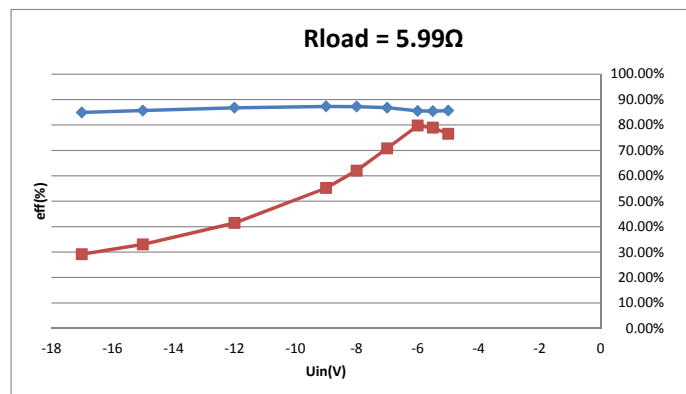
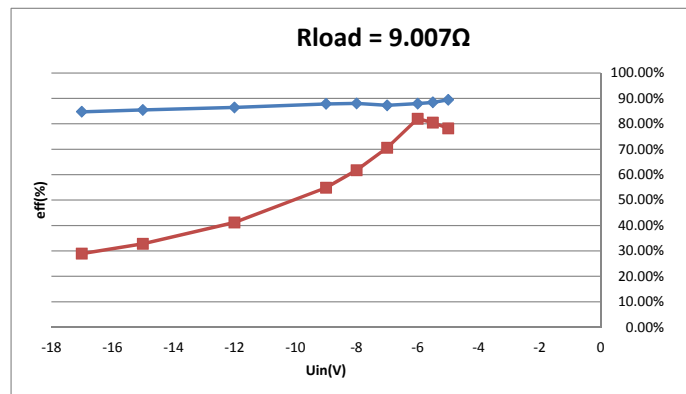


Figure 10. Measured efficiency of the negative voltage regulator at 0.5 A (a) and 0.85 A (b) output current. Blue = switch-mode voltage regulator; red = standard 7905.

blow on it, the output rises slightly again (which actually means it becomes more negative — it's all relative). Here again we also tested the circuit with a pulsed load, and it passed the test with good results. As can be seen from **Figure 11**, the voltage regulator is stable over the entire input voltage range (the figure shows the worst case situation with an input voltage of -17 V).

The switch-mode voltage regulators described here are a good alternative to normal linear voltage regulators. Although they generate a bit more output ripple and their construction is more complex, they are an excellent choice when efficiency and dissipation are important design criteria. (120212)

Internet Links

- [1] TPS61170: www.ti.com/product/tps61170
- [2] TPS62150: www.ti.com/product/TPS62150
- [3] www.elektor.com/120212

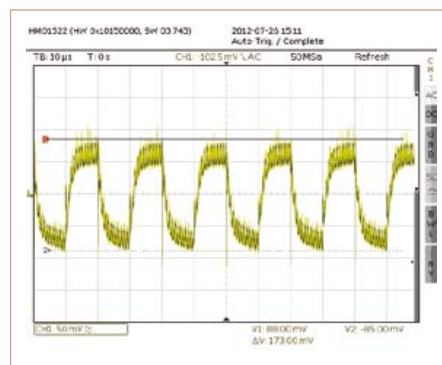


Figure 11. Response to pulsed load at 50 kHz, 60–560 mA, -17 V in, AC coupled.

SAVING COST=TIME with readily available FPGA boards

- Basic and simple features, single power supply operation
- Quality and reliability is provided by years of sales
- Same board size and connector layout – ACM/XCM series
- All stocked items are ready to be shipped immediately
- Over 100 varieties of FPGA/CPLD boards are available
- Customizing speed grade and/or any features are possible
- Free download technical documents before purchasing
- High quality and highly reliable FPGA/CPLD boards from Japan
- Almost all products are RoHS compliance

ALTERA FPGA Board

Cyclone IV E F780 FPGA board

ACM-204 series

Cyclone IV E SDRAM

EP4CE30F29C8N
EP4CE40F29C8N
EP4CE115F29C8N
Credit card size (86 x 54 mm)

RoHS compliant



XILINX FPGA Board

Spartan-6 FGG484 FPGA board

XCM-018/018Z series

Spartan-6 MRAM DDR2

XC6SLX45-2FGG484C
XC6SLX75-2FGG484C
XC6SLX100-2FGG484C
XC6SLX150-2FGG484C
Credit card size (86 x 54 mm)

RoHS compliant



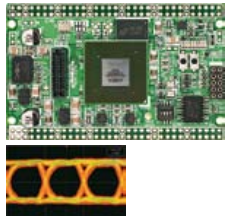
Arria II GX F572 FPGA board

ACM-025 series

Arria II GX DDR2

EP2AGX45DF25C6N
EP2AGX65DF25C6N
EP2AGX95DF25C6N
EP2AGX125DF25C6N
Credit card size (86 x 54 mm)

RoHS compliant



Spartan-6 FGG484 FPGA board

XCM-110/110Z series

Spartan-6 MRAM DDR2

XC6SLX45-2FGG484C
XC6SLX75-2FGG484C
XC6SLX100-2FGG484C
XC6SLX150-2FGG484C
Compact size (43 x 54 mm)

RoHS compliant



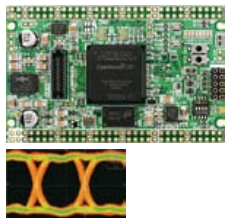
CycloneIV GX F484 FPGA board

ACM-024 series

Cyclone IV GX DDR2 SIF40

EP4CGX50CF23C8N
EP4CGX75CF23C8N
EP4CGX110CF23C8N
EP4CGX150CF23C7N
Credit card size (86 x 54 mm)

RoHS compliant



Virtex-5 FFG676 FPGA board

XCM-011 series

Virtex-5 FRAM SDRAM

XC5VLX30-1FFG676C
XC5VLX50-1FFG676C
XC5VLX85-1FFG676C
XC5VLX110-1FFG676C
Credit card size (86 x 54 mm)

RoHS compliant



Cyclone IV E F484 FPGA board

ACM-107 series

Cyclone IV E MRAM

EP4CE55F23C8N
EP4CE75F23C8N
EP4CE115F23C8N
Compact size (43 x 54 mm)

RoHS compliant



Virtex-5 LXT FFG665 FPGA board

XCM-017 series

Virtex-5 SDRAM RocketIO SIF40

XC5VLX30T-1FFG665C
XC5VLX50T-1FFG665C
Credit card size (86 x 54 mm)

RoHS compliant

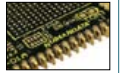


FPGA/CPLD Stamp Module PLCC68 Series

Easy and Quickly Mountable Module

FPGA Module IC socket mountable

- 50 I/Os (External clock inputs are available)
- 3.3V single power supply operation (Voltage converters for auxiliary power supply are built-in)
- Separated supply-inputs: Core, I/O drivers
- JTAG signal
- All PLCC68 series have common pin assignment
- Very small size (25.3 x 25.3 [mm])
- RoHS compliance
- MADE IN JAPAN



XILINX PLCC68 Series

Spartan-6 PLCC68 FPGA Module

XP68-03

Spartan-6 PLCC 68

XC6SLX45-2CSG324C

3.3V single power supply operation
On-board oscillator, 50MHz
RoHS compliant



Spartan-3AN PLCC68 FPGA Module

XP68-02

Spartan-3AN PLCC 68

XC3S200AN-4FTG256C

FPGA internal configuration ROM
Two User LEDs
RoHS compliant



ALTERA PLCC68 Series

Cyclone III PLCC68 FPGA Module

AP68-04

Cyclone III PLCC 68

EP3C25U256C8N

3.3V single power supply operation
On-board oscillator, 50MHz
RoHS compliant



Cyclone III PLCC68 FPGA Module

AP68-03

Cyclone III PLCC 68

EP3C10U256C8N

4Mbit Configuration Device
Two User LEDs
One User Switch(Slide)
RoHS compliant



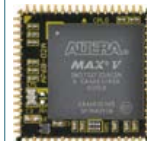
MAX V PLCC68 CPLD Module

AP68-02

MAX V PLCC 68

5M570ZF256C5N

External Clock inputs
On-board Voltage regulator
RoHS compliant



Embedded Linux Made Easy (5)

I/O, ADC, PWM, LAN & Web server

The Elektor Linux board is proving to be a big hit. The board's versatility makes it an ideal learning tool and platform for Linux application development. In this installment we set about reading digital and analog signals then we hook up a network adapter and build a small web server which generates dynamic HTML pages. Using this we can read the status of remote LEDs amongst other things and display the information in a browser.

By Benedikt Sauter [1]

We've already spoken of boot loaders and Kernels and many readers have already taken their first steps with the file system and SD cards (check out the text box 'SD card image'). In this installment we move onto the first 'real world' task. Embedded Linux solutions

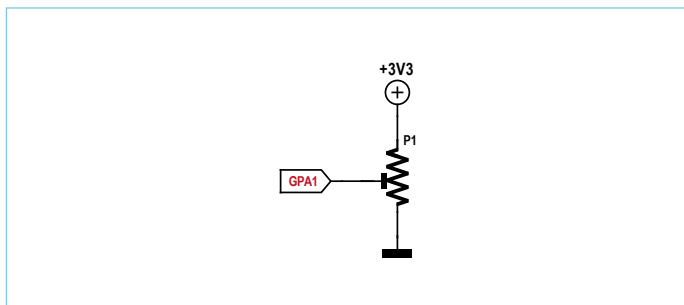


Figure 1. Use a pot to test the A/D converters

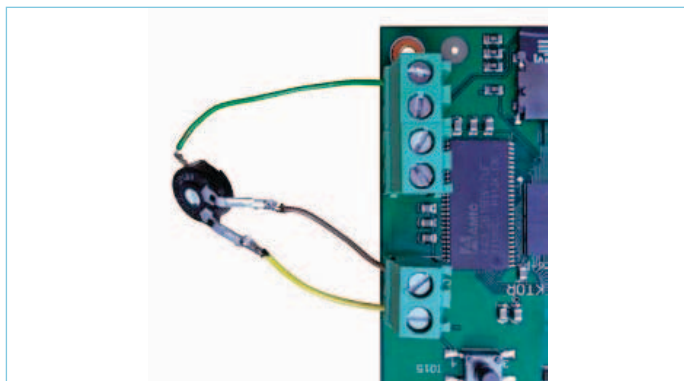


Figure 2. The analog inputs are connected via a connector block

are often found in applications such as process control and data collection. We start by showing how to input and output both analog and digital values. Next we set up a network connection to allow remote access of the board and remote control from a web page.

Digital I/O pins

In the second installment of this series we have already managed to turn LED1 on and off. The LED is connected to the GPIO3 pin of the processor. These GPIO-Pins can be configured as either input or output and also as an interrupt input. The procedure for initializing the I/O pins should be familiar to the majority of Elektor readers by now:

- activate the pins as GPIO;
- initialize the data direction;
- output a value or read in the signal level on the pin.

Under Embedded Linux we can talk to the GPIO-Pins via the device driver from the console.

First we go to the communications folder with the GPIO driver:

```
cd /sys/class/gpio
```

Next we activate any pins connected to LEDs as GPIO (see circuit diagram in [2]):

```
echo 3 > export
```

Now we must activate the pin connected to the pushbutton as a GPIO:

```
echo 15 > export
```

Next configure the pins as either output or input:

```
cd gpio3
echo "out" > direction
```

```
cd ../gpio15
```

```
echo "in" > direction
```

Now we can control the LED (as already shown) using the following command to turn the LED on and off:

```
cd ../gpio3
echo 1 > value
```

```
echo 0 > value
```

The pushbutton status is contained in a (virtual) file called 'value'. The value of which can be read using the cat command:

```
cd ../gpio15
cat value
```

Now you can send commands to control the relay on the board. It is connected to GPIO18 and the pin can be configured as an output in the same way as above.

Analog/Digital converter

The LPC3131 provides four analog inputs with up to 10-bit resolution. The range of measured values therefore lie between 0 and 1023 (or 0 to 0x3FF in Hexadecimal). The 3.3 V supply is used as the voltage reference and also powers the I/O bank.

Reading the converter output value is similar to reading a push button status. The A/D converter has its own driver which can only output the value from one channel at a time. Therefore it will first be necessary to set up which channel is to be read.

For simplicity the A/D converter function can be tested with a simple pot or preset resistor to supply the variable analog voltage. The circuit is shown in **Figure 1**. The track ends are connected between 3.3 V and ground while the wiper connects to pin GPA1 via the terminal blocks.

The setup should look roughly like **Figure 2**. Now the A/D converter can be initialized and successive measurements taken (see **Figure 3**).

During testing it can be tedious to keep entering the same commands. The program 'watch' automates this procedure. With an input of:

```
watch -n 1 cat /dev/lpc313x_adc
```

The tool calls the chosen command once per second. Use Ctrl-C to stop the process.

Next to GPA1 on the Elektor Linux board can be found the ADC channels GPA0 and GPA3 on header J5 (GPA2 is not brought out).

To protect the A/D input (to some extent) from damage by over voltage or over current connect a 10 kΩ resistor in series with the input and also a 3.3 V zener diode down to ground (**Figure 4**).

Generating PWM signals

Many applications such as servo controllers, switched-mode voltage

```
root@gnublin:~# echo 1 > /dev/lpc313x_adc
set res to 10
root@gnublin:~# cat /dev/lpc313x_adc
0x21f
root@gnublin:~# cat /dev/lpc313x_adc
0x11d
```

Figure 3. Initialization of the A/D converters and output of two values.

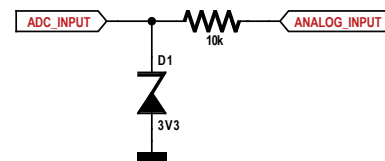


Figure 4. Protection for the A/D converter inputs.

Listing 1: PWM.

```
#include <stdio.h>
#include <stdlib.h>

#ifndef abs
#define abs(x) ((x) < 0 ? -(x) : (x))
#endif

int pwm(int value) {
    FILE* f = fopen(„/dev/lpc313x_pwm“, „wb“);
    fputc(value & 0xff, f);
    fputc((value >> 8) & 0xff, f);
    fclose(f);
}

int main() {
    int value = 0;
    int b;

    while(1) {
        b = abs(63 - 2*value);
        pwm(b * b);
        value = (value + 1) % 64;
        usleep(1000);
    }
}
```

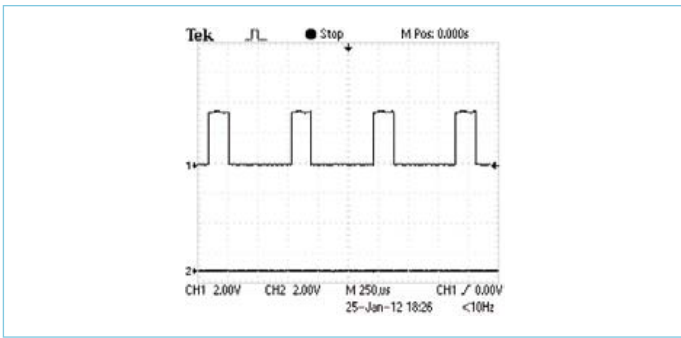


Figure 5. PWM output when a value of 1000 is used as a comparison value.

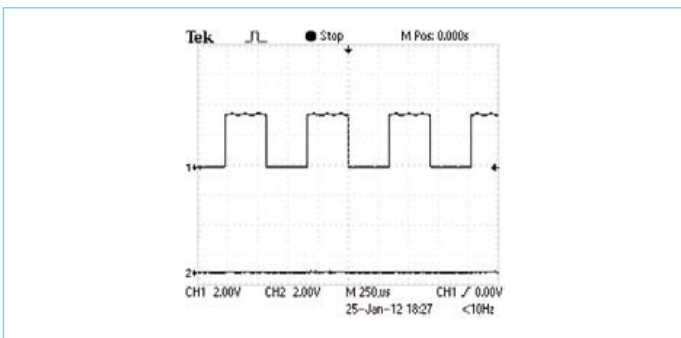


Figure 6. PWM with 50 % mark/space ratio.

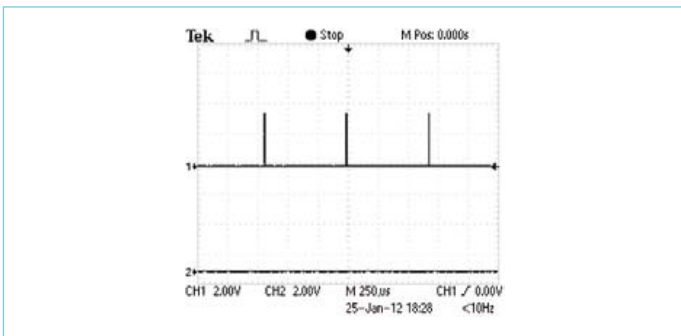


Figure 7. PWM with 1 % mark/space ratio.

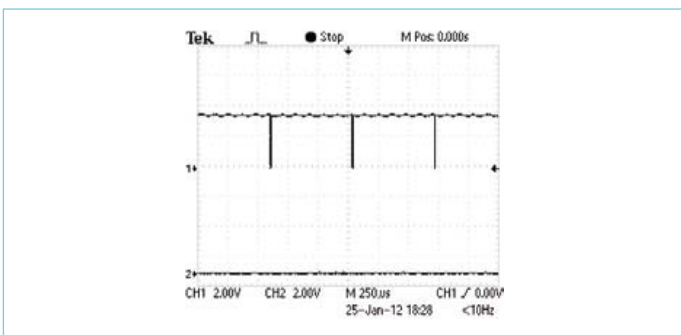


Figure 8. PWM with 99 % mark/space ratio.

generators and digital audio (and much more [3]) need a generator of pulse width modulated signals. The Elektor Linux board outputs PWM signals from header J5. For test purposes use an oscilloscope to view the generated signals.

In PWM mode the controller increments a 12-bit counter on each clock pulse. When the counter value reaches a predetermined value it switches state of the PWM pin from high to low (when the counter overflows it is reset and the bit goes high) The predetermined value can be any 12-bit value i.e. in the range from 0 to 4095. When the value of 0 is specified the PWM output will go low immediately. A value of 2000 gives a square wave with a mark-space ratio of around 50 %.

In contrast to both the I/O and ADC drivers the PWM driver expects a binary input value so it is not simple to use echo or cat because the value supplied will be interpreted as a character (ASCII). We need the assistance of a small help program.

We were able to quickly write this on board using C. We have included a copy of this program (**Listing 1**). In the Home folder, where you will always find yourself after logging in, you will find the file 'pwm.c'. In the code it is necessary to change the name of a device file.

First open the file using an editor on the board:

```
nano pwm.c
```

Navigate along the following line using the arrow keys...

```
FILE* f = fopen("/dev/pwm", "wb");
```

... change the line to read:

```
FILE* f = fopen("/dev/lpc313x_pwm", "wb");
```

Now save the edited version using Ctrl-o and end the editing session with Ctrl-x. The code can be compiled on the PC or directly on the Linux board which also contains its own compiler:

```
gcc -o pwm pwm.c
```

Once compiled (this takes a few moments) it can be directly executed:

```
./pwm
```

The oscilloscope display shows how the mark/space ratio changes.

When a signal with a fixed mark/space ratio is required this can be achieved for example with a small script written in the programming language Python. The file 'pwm.py' can be found in the home folder.

First it is necessary to start the Python interpreter:

```
python
```


SD card image

When experimenting — which we like to encourage! — you can sometimes find yourself backed down a one-way street with no way out. In this situation there may be no alternative but to take a fresh version of the original SD card. For this reason we are offering the SD card contents as an extra download. First download the image from the Elektor web [8] (Download 'SD Card Image', 120180-12.zip).

When the download is complete, unpack the archived files:

```
unzip 120180-12.zip
```

This takes a while before the following message appears:

```
Archive: ../120180-12.zip inflating: Elektor_Linux_Board - Build_New_SD_Card.txt inflating: gnuvmlin.gz
```

Now take the SD to be written to and plug it into the PC or card reader. The system will read the card but we are not interested in this, we just need a 1:1 image of the downloaded file on the memory card. For this it will be necessary to manually configure the card.

The best way is to plug the card in the reader and give the command `dmesg` via the console to find out what the card has been mapped to. The response to this command will be something like the following:

```
[ 1069.427374] sdf: sdf1 sdf2 [ 1069.430857] sd
5:0:0:0: [sdf] No Caching mode page present [
1069.430863] sd 5:0:0:0: [sdf] Assuming drive cache:
write through [ 1069.430868] sd 5:0:0:0: [sdf] Attached
SCSI removable disk [ 1070.002620] EXT2-fs (sdf1):
warning: mounting unchecked fs, running e2fsck is
recommended
```

In the last line there is an indication of the device name that the kernel has given to the SD card (sdf1 in this case).

Now it is necessary to manually unmount all partitions by using...
`umount /dev/sdf1`

... replace `sdf1` with the device name assigned to your card (exactly as the name assigned to the first partition).

Now the downloaded image can be written to the SD card:
`sudo dd if=gnuvmlin.gz of=/dev/sdf`

`sdf` is the description of the whole card as a block device.

NB: The process of writing can take up to 10 minutes.

In the interpreter (already we are in interactive mode) we can load the PWM module (a library of Python functions):

```
import pwm
```

Now it is possible to call the module functions. One of these functions allows direct input of the counter compare value:

```
pwm.pwm_raw(1000)
```

The signal on the oscilloscope should now look like **Figure 5**.

Alternatively the mark-space ratio can be given as a percentage:

```
pwm.pwm(50)
```

the waveform on the screen now looks like **Figure 6**.

Using the command

```
pwm.pwm(1)
pwm.pwm(99)
```

is interpreted as mark then space so produces 1 to 99 % (**Figure 7 / Figure 8**).

Use `Ctrl-d` to close the Python-Interpreter.

Be aware that at the start Python takes a little while to fully initialize but once running it responds quite smoothly.

Network interfacing

In the last installment of this series we have already shown how to interface a USB/UART adapter. Now we install another USB adapter to the Linux board which can connect to an Ethernet network. We will be using an off-the-shelf USB/LAN adapter (**Figure 9**). There

are many different models on the market but they mostly use the same or similar chipsets. The one we are using here is the 'D-Link DUB-E100' [4].

In the last installment of this series we integrated the driver for the USB/UART adapter in the kernel. As we have already shown in this series the kernel can also load a device driver as a module during run time. We will use this approach for the network adapter. The file system already contains many different drivers.

In the case of D-Link adapters it is necessary to give the following command:

```
modprobe asix
```

Now we should see an output as shown in **Figure 10**. There are three drivers in the file system:

```
asix
pegasus
net1080
```

When a different adapter is used try loading different drivers one after the other. To check if the correct one has been found we should see this response to the input:

```
ifconfig -a
```

A response of 'eth0' indicates the driver has been correctly loaded and the network is ready to transfer data.

When all of the drivers fail it is possible to go to 'Device Drivers' -> 'Network device support' -> 'USB Network Adapters' and load the drivers by hand. The drivers can either be compiled into the kernel



Figure 9. A USB/LAN adapter gives the board a network interface.

(recommended for beginners, see [2]) or after the modules have been converted, copied into the file system and then later loaded into the kernel.

All modules can be converted using:

```
make modules
```

The new module can be installed on the card using:

```
make modules_install INSTALL_MOD_PATH=/mnt
```

Instead of /mnt the SD card path should be used here

Now that the network adapter has been recognized it can be given a temporary IP address. It is worth hooking up a PC to the network to check just which addresses have already been assigned before the temporary IP address is chosen. This can be performed in Linux (or

Windows) from the console by issuing the command ping:

```
ping 192.168.0.7
```

When the program reports that no device responded...

```
2 packets transmitted, 0 received, 100% packet loss, time 1006ms
```

... Then this IP address is free to be used. So we give this to the Linux board:

```
ifconfig eth0 192.168.0.7
```

Once this has been set up another ping attempt by the PC should now elicit a positive response. (Figure 11). Optionally the DHCP server can be allowed to allocate addresses automatically (Figure 12).

To ensure the driver is automatically loaded at every start it is necessary to add its name to the '/etc/modules' file. All of the modules in this file will be automatically loaded during Linux boot process. The network IP address is stored in the file '/etc/network/interfaces'. This file already exists in our file system. Use an editor to enter your in-house IP address after 'eth0'. Now each time the Elektor Linux card is started it will be ready to communicate with your local network.

Web server

Now that a network connection is available we can start a small web server to view our first demo page with a browser. In the home folder of the users 'root' is a small script which starts the well-known web server 'lighttpd':

```
root@gnublin:~# ./lighttpd-init.sh
```

Syntax OK

```
root@gnublin:~#
```

Using the browser to visit the IP address mentioned above will show

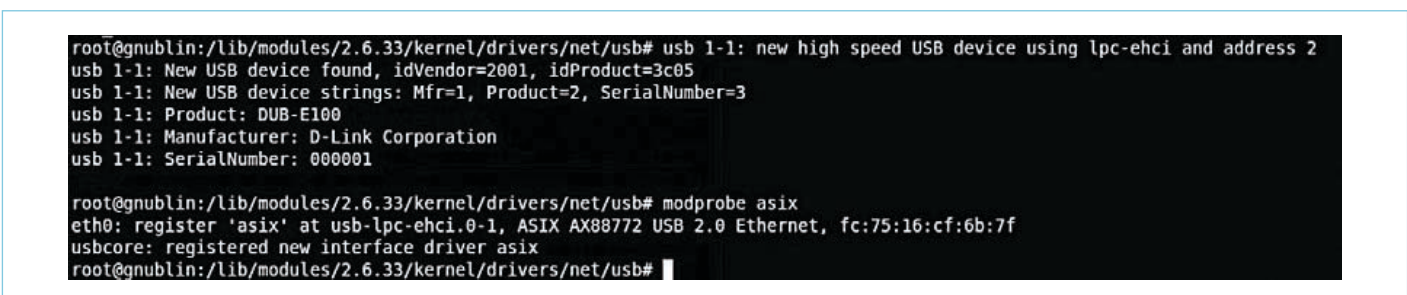


Figure 10. Automatic recognition of the USB/LAN adapter.

```
sauter@pc:~$ ping 192.168.0.7
PING 192.168.0.7 (192.168.0.7) 56(84) bytes of data.
64 bytes from 192.168.0.7: icmp_req=1 ttl=64 time=1.24 ms
64 bytes from 192.168.0.7: icmp_req=2 ttl=64 time=1.50 ms
--- 192.168.0.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.248/1.375/1.502/0.127 ms
```

Figure 11. A reply following a ping.

```
root@gnublin:~# udhcpd
udhcpd (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.182...
Lease of 192.168.0.182 obtained, lease time 864000
Setting IP address 192.168.0.182 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.0.1
root@gnublin:~#
```

Figure 12. The DHCP server automatically assigns the address.



Figure 13. Test page of the web server.

the web site in **Figure 13**.

A web server typically generates static HTML web pages. When we show the status of an LED in a browser for example, the web server must output a dynamically assembled HTML page (depending on the LED status). We require an interface between the web server and an external program that is capable of detecting whether the LED is on or off and can generate a corresponding web page.

The simplest link is called CGI. This ‘Common Gateway Interface’ is an interface which enables the web server to access almost any program. One condition is that it is command-line orientated, and can also be started (with any parameters) from the console. CGI scripts also most commonly return an HTML page. As a CGI program you can use a simple Linux shell script, a PHP or Python program or even a C program.

Switching an LED from the browser

This simple application will show how it is possible to change the state of an LED from a browser. We create a simple script file that the shell (console) can directly execute. First it is necessary to setup the CGI interface on the web server.

In the file ‘/etc/lighttpd/modules.conf’ identify the entry...

```
#include "conf.d/cgi.conf"
```

... Using an editor (e.g. nano or vi) change it to:

```
include "conf.d/cgi.conf"
```

Next, in the file ‘/etc/lighttpd/conf.d/cgi.conf’ it is necessary to edit the entry

```
#alias.url += ( "/cgi-bin" => server_root + "/cgi-bin" )
```

To...

```
alias.url += ( "/cgi-bin" => var.server_root + "/cgi-bin" )
```

Now the web server knows that the files in the ‘/cgi-bin’ folder can be treated as programs (and not as HTML pages to be sent to the browser)

Next for us to finally use a simple shell script as a CGI program it is necessary to identify the region in the file ...

```
cgi.assign          = ( ".pl" => "/usr/bin/perl",
                        ".cgi" => "/usr/bin/perl",
                        ".rb" => "/usr/bin/ruby",
                        ".erb" => "/usr/bin/
eruby",
                        ".py" => "/usr/bin/
python" )
```

... And add the line ‘.sh’ => ‘/bin/sh’

```
cgi.assign          = ( ".pl" => "/usr/bin/perl",
                        ".cgi" => "/usr/bin/perl",
                        ".rb" => "/usr/bin/ruby",
                        ".sh" => "/bin/sh",
                        ".erb" => "/usr/bin/
eruby",
                        „.py“ => „/usr/bin/
python“ )
```

The next step is to create a directory for the CGI programs:

```
mkdir -p /srv/www/htdocs/cgi-bin
```

And lastly to add the program in **Listing 2**, after that the editor can



Figure 14. LED switching in the browser.

be started with the following command:

```
nano /srv/www/htdocs/cgi-bin/example.sh
```

The listing is included in the downloads for this part of the course [5], so you can just copy these to save wear and tear on your keyboard.

In order for the web server to control the LED it is necessary to set the configuration and the data direction from the console:

```
echo 3 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio3/direction
```

From security reasons the web server never runs as user 'root' but we must temporarily allow access to allow the web server access to control the LED:

```
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value
```

Now everyone in the system has access to the LED. This is not an optimal solution but to selectively assign access rights is something we will not go into for the time being. More information on this topic can be found at [6].

Apart from this the web server must be granted appropriate privileges to store log files in a previously generated directory:

```
mkdir /var/log/lighttpd
chown -R lighttpd:lighttpd /var/log/lighttpd
```

Start the web server now...

```
root@gnublin:~# /etc/init.d/lighttpd restart
```

... Then you should get the message:

```
Syntax OK
```

When the browser accesses the previously set up IP address, the page shown in **Figure 14** is opened and the LED status displayed. We have built a little control demonstration with the help of a mini HTML form which in this case only contains a submit button. Pressing the button transfers the form control elements to our web browser. In this case we use this mechanism to inform the web server that it must call the CGI script '/cgi-bin/example.sh'. This toggles LED1 on the board and builds the new web page with the changed status message.

Coming up

In the next installment we will build a more complex HTML user interface which will allow us to control more functions of the board. It goes without saying that this will not be performed from a nice clean user interface where you can't see what's happening under the surface; our control requires a little more intelligence. This can

Listing 2: CGI script to generate a website.

```
#!/bin/sh

if [ "$REQUEST_METHOD" == "POST" ]
then
    if [ 'cat /sys/class/gpio/gpio3/value' == 1 ]
    then
        echo 0 > /sys/class/gpio/gpio3/value
    else
        echo 1 > /sys/class/gpio/gpio3/value
    fi
fi

echo "Content-Type: text/html; charset=utf-8"
echo ""
echo "<html>"
echo " <head>"
echo " <title>Webserver CGI Port 3 (LED)</title>"
echo " </head>"
echo " <body>"
echo " <h1>Control-Panel CGI Port 3</h1>"
if [ 'cat /sys/class/gpio/gpio3/value' == 1 ]
then
    echo " Port: On"
else
    echo " Port: Off"
fi
echo " <br><br>"
echo " <form action=\"/cgi-bin/example.sh \" method = \"POST\">"
echo " <input type=\"submit\" value=\"Click\">"
echo " </form>"
echo " </body>"
echo "</html>"
```

be achieved by a small program running in the background. For the edition after that and then for the final installment of this series (the first edition of the year 2013) we have planned something special: The subjects we intend to cover will be entirely up to you. Go to our special web site [7] and tell us what you would like to see! A good deal of user talk on the project may also be found on our forum [9].

(120182)

Internet Links

- [1] sauter@embedded-projects.net
- [2] www.elektor.com/120181
- [3] http://en.wikipedia.org/wiki/Pulse-width_modulation
- [4] <http://shop.embedded-projects.net/gnublin>
- [5] www.elektor.com/120182
- [6] http://en.gnublin.org/index.php/Permissions_GPIO
- [7] www.elektor.com/linux-feedback
- [8] www.elektor.com/120180
- [9] www.elektor.com/forum/elektor-forums/fields-of-interest/micro-controllers-embedded.1543738.lynx

Advertisement

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

DLP-USB1232H: USB 2.0 UART/FIFO

HIGH-SPEED
480Mb/s

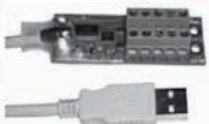


Only \$28.95!

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

DLP-IO8-G

8-Channel Data Acquisition



Only
\$29.95!

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

DLP-IOR4

4-Channel Relay Cable

DLP-TH1b

Temp/Humidity Cable

DLP-RFID1

HF RFID Reader/Writer

DLP-FPGA

USB-to-Xilinx FPGA Module

DLP
Design

www.dlpdesign.com



THE ORIGINAL SINCE 1994
PCB-POOL
Beta LAYOUT

FREE Stencil
with every prototype order

EAGLE order button
pcb-pool.com/download-button
20% off! on your first order

Call Tyler: 1 707 447 7744
sales@pcb-pool.us

PCB-POOL® is a registered trademark of
www.pcb-pool.com **Beta**
LAYOUT

Nixie VU Meter

A warm & retro power indicator for the tube amp

This circuit is designed to embellish a (tube) amplifier. It supplies delightful light effects rather than a calibrated readout for output power or decibels. At the heart of the circuit is a PSoC from California that's busy all the time driving two Nixie tubes... from the Ukraine.



By Harald W. Cintra (Brazil)

Distance is meaningless these days it seems, and borders non-existent. The same with e-age, in a way, considering the PSoC (programmable system on chip) and the IN-9 Nixie tube differ about 45 years in years yet get along really well in an Elektor circuit published in 2012. Let's see how a 1960s 'Sovjet-elektronisjk' device and a 'Silicon Valley' bit of technology from the Internet Age can be coaxed to work together in a synergetic manner — hosted by Elektor.

Why the IN-9?

The IN-9 seems to be the ideal tube for this circuit, being a linear gas discharge tube emitting an orange/pink-ish glowing light. The height of the 'glowing column' is proportional to the current flowing and the anode voltage is around 150 VDC. For sticklers: 'linear' applies to most of the device's range, not all of it, see **Figure 1**. Between 0 and 8 mA everything seems all right at about 10.3 mm per mA but beyond that expect *borsj*, or "*nothing meaningful*" in Oxford English. These tubes can be obtained as NOS (new old stock) devices on the Internet, specifically on a certain auc-

Features

- Column type stereo VU meter
- 2 x IN-9 linear gas discharge (Nixie) tube
- Column colour: warm orange/pink
- Net column height: 85 mm (approx.)
- Audio 'Line' drive level (approx. 0.7 V)
- Cypress PSoC CY8C27443 based
- PSoC device available ready-programmed
- Software lin/log converter
- Onboard 150 VDC supply
- Compact board (65 x 65 mm)

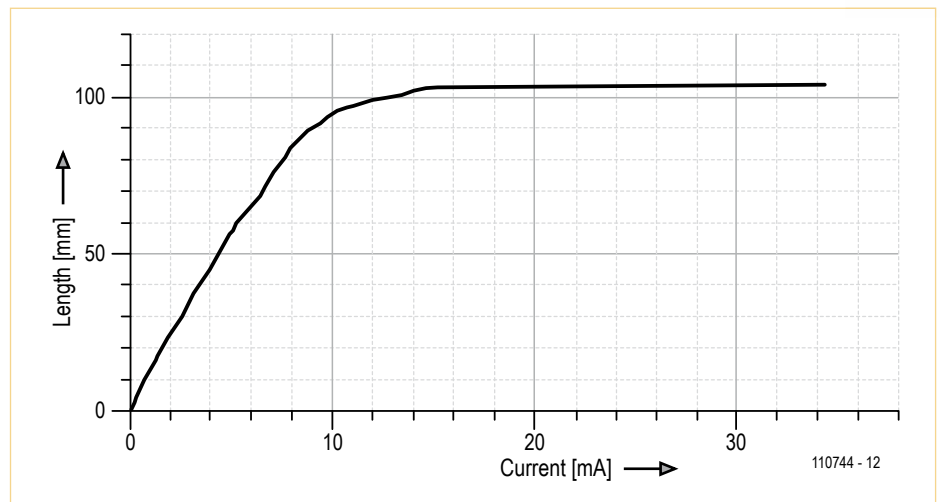
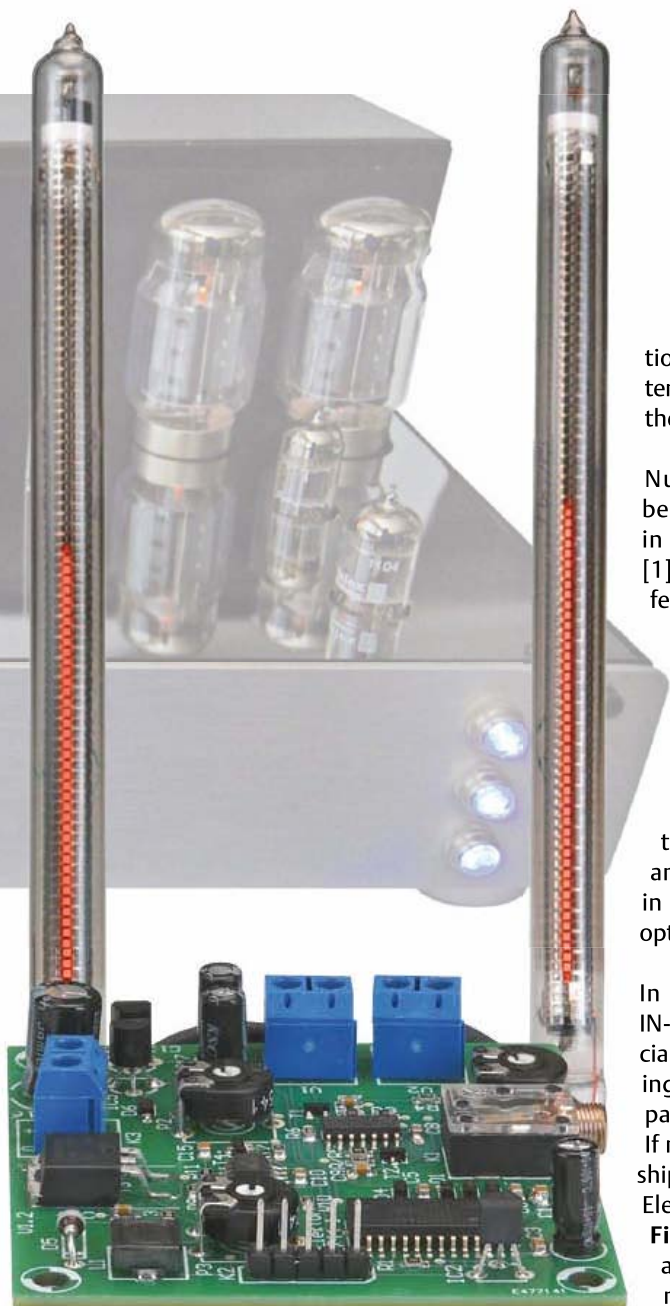


Figure 1. Measured (nope, not "manufacturer specified") relation between IN-9 illuminated column (bar) height and tube current. Stay in the 0-8 mA range and you're fine. Illustration redrawn, permission granted, from an original by Jeff Malins (2004).



tion site sumptuously headquartered in California not too far from the Cypress Corp.

Numerical Nixie tubes have been covered in recent projects in *Elektor* like a thermometer [1] and a Sputnik clock [2] so we feel confident in referring you to those pages for theoretical backgrounders and quick-start guides if you are new to these wonderful devices hailing us from a distant past when tubes ruled electronics. By comparison, it's difficult to imagine a PSoC gladdening an electronics engineer's heart in any way in 2060, but let's be optimistic.

In terms of datasheets for the IN-9, do not expect anything official out there, and rejoice at finding anything resembling printed paper in your shipment of IN-9s. If not, Google is your friend. The shipment that eventually reached *Elektor Labs* is pictured 'as is' in **Figure 2** (Babushka added for amusement). Apart from Ukrainian state of the art packag-

ing you can also see some numerical Nixie tubes, a socket and some driver ICs we ordered at the same time through Ebay (there, we've said it).

Put a PSoC in it

For a VU (volume unit) meter (even one that's essentially a coarse but nice looking sound level indicator) a logarithmic amplifier is in order. Without it, your visual and audible impressions go terribly out of step. It was decided to do this in software instead of hardware and the Cypress type CY8C27443 PSoC eventually got selected — a rare find in *Elektor*.

The CY8C27443 is an excellent processor for this circuit, because we can use all the hidden power of a SoC processor with many options of A/D, D/A, amplifiers and so on, reducing the number of external components to a minimum.

Barring the rectifier and the current source, the units shown in the block diagram in **Figure 3** are contained in the PSoC. From left to right: an input amplifier followed by a rectifier, another amplifier, an A/D converter, a software linear-to-logarithmic table conversion, a D/A converter and finally a current source to drive the IN-9 tube. All this multiplied by two, of course, for L (left) and R (right) channels — yes, tube amplifiers are stereo too these days.

Practical circuit

The circuit diagram in **Figure 4** proves that a PSoC can excel in keeping the dreaded component count to a minimum.

The board is powered by a 12 VDC external power adapter. In order to generate the 150 V anode voltage needed for the IN-9, we build a small step up inverter based in the old faithful NE555 (IC6) wired as an astable multivibrator and using the control voltage input to regulate the output voltage. Components R10, P3, R13 form a voltage divider passing a feedback voltage to the control voltage input of 555 by way of T4. The preset allows you to set the IN-9 anode voltage to 150 V.

We also need a +5 V supply for the PSoC, this is built with a 7805 (IC7), and a +10 V



Figure 2. Nixie tubes and ancillaries packaged "Soviet style". No worries, all parts turned out to work just fine!

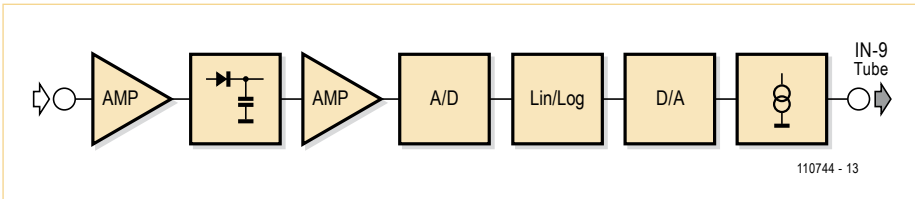


Figure 3. Audio signal chain — most of it is handled inside the PSoC device.

1960s Soviet-elektronisjk meets 2012 Silicon-Valley tech hype

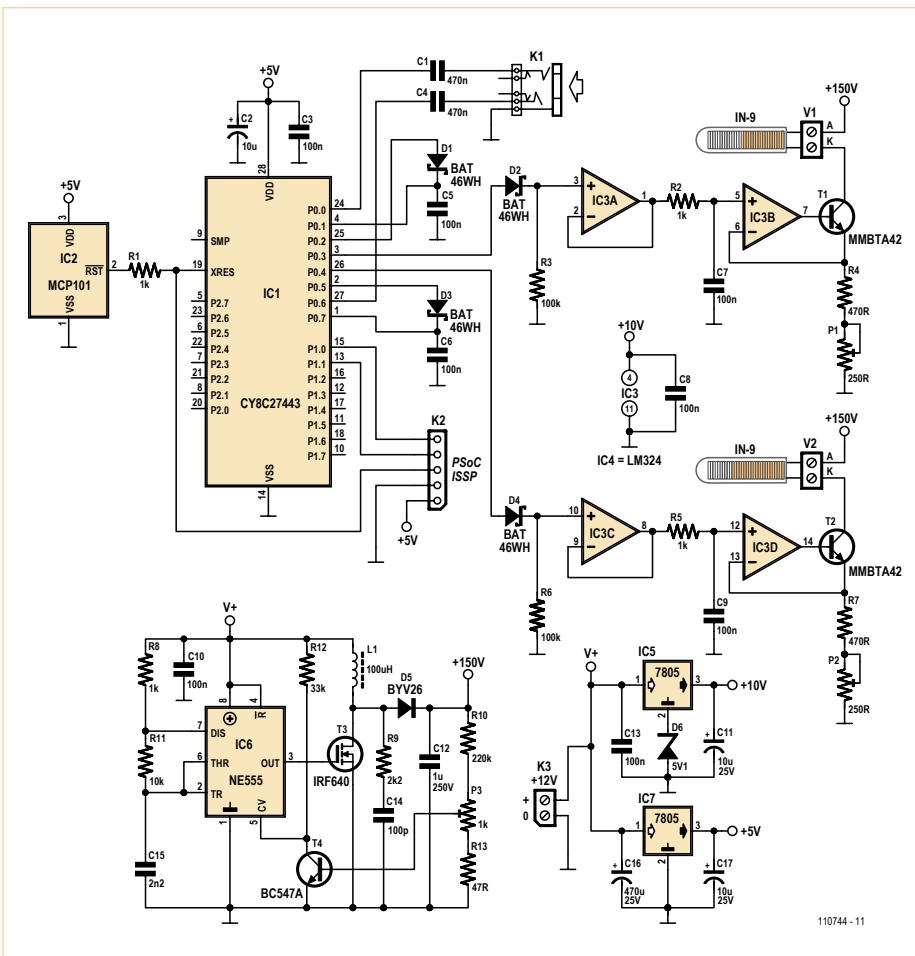


Figure 4. The circuit is a prime example of a PSoC doing all the quasi-intelligent work, as well as keeping the parts list short. However it cannot do anything really useful without the assistance of IN-9 Nixie tubes and an NE555, two components dating back to the 1960s and 1970s respectively.

supply for the operational amplifiers, using another 7805 (IC5) this time with its ground terminal ‘jacked up’ by a 5.1 V zener diode. The PSoC is happy with 5 V only and a reset signal to work properly. A dedicated reset signal generator IC type MCP101 (IC2) is used to guarantee a dead-cert power-on reset.

Looking at the audio path, the Line-level input signal first passes through a decoupling capacitor (C1/C4) and goes direct to PSoC for amplification by a factor of 2. The output of this amplifier is routed to another PSoC pin for rectification by a BAT46WH Schottky (i.e. low drop) diode (D1/D3) and re-enters the PSoC.

Inside the PSoC, this rectified signal is lightly amplified and then arrives at the internal 8-bit A/D converter. The A/D converter output value is entered into a linear-to-logarithmic (lin/log) conversion table, the output of which goes to the D/A converter, still internal to the PSoC.

Back at the component level, through a diode (D2/D4) the PSoC’s D/A output is fed to a buffer (IC3A/C), passes through an R/C filter (R2/C7, R5/C9) and finally drives a current sink built around an LM324 opamp (IC3B/D), a high voltage transistor (T1/T2) and a range adjustment resistance (P1/R4, P2/R7). The purpose of D2 (D4) is to introduce a light threshold to avoid very small signals to cause activity in the IN-9 tubes. The current range presets P1/P2 are adjusted so that a fully glowing bar is obtained at the highest input level.

Returning to the PSoC, **Figure 5** is a serious attempt at showing overview of its internal configuration. **Figure 6** concentrates on the input amplifiers, PGA_2 and PGA_4, and their outputs exiting from the

chip and returning via PGA_1 and PGA_3 for buffering and subsequent routing to DUAL-ADC8 were they get digitized. With some effort you can also see the output of D/A converters routed to output pins. The PSoC design files are available free of charge from the Elektor website [3]. Cypress offer a number of great software tools to design and configure the innards of their PSoC devices. For those of you less concerned about the way a PSoC works or gets programmed, Elektor are offering ready-programmed ICs [3].

Construction

The project is built on the circuit board pictured in Figure 7. It's

a hybrid, i.e. a mix of through-hole and SMD. The SMDs have to be mounted first. The PSoC and the LM324 do not have an unmanageable pin pitch, hence can be soldered by hand with good care and precision using a solder iron with a very fine tip. Various methods exist and these have been described many times in Elektor.

The electrolytic capacitors are all radial through-hole types – be sure to mount them with the correct polarization.

The IN-9 tubes should sit in sockets to prevent any risk of their lead wires breaking off at the 'Murphy Spot', i.e. where they exit from the glass envelope.

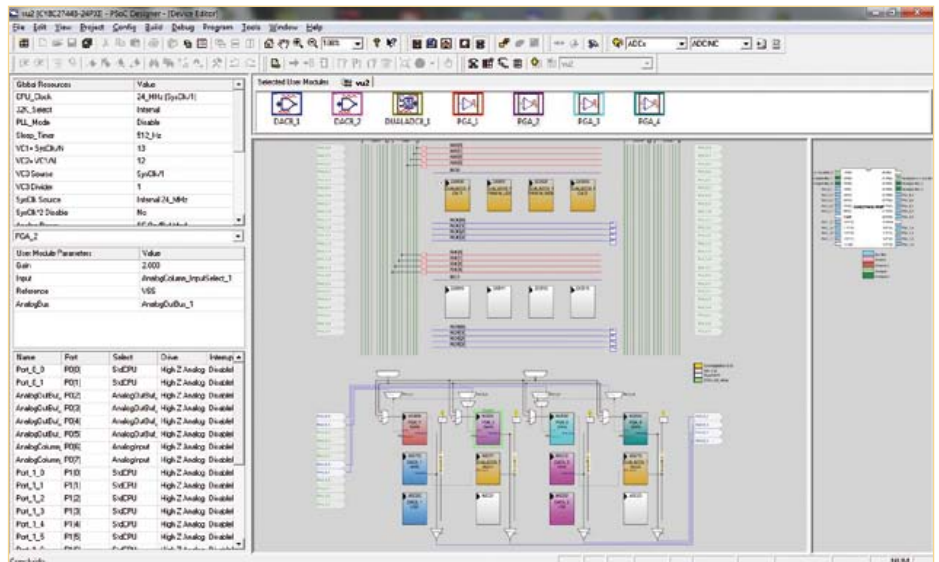
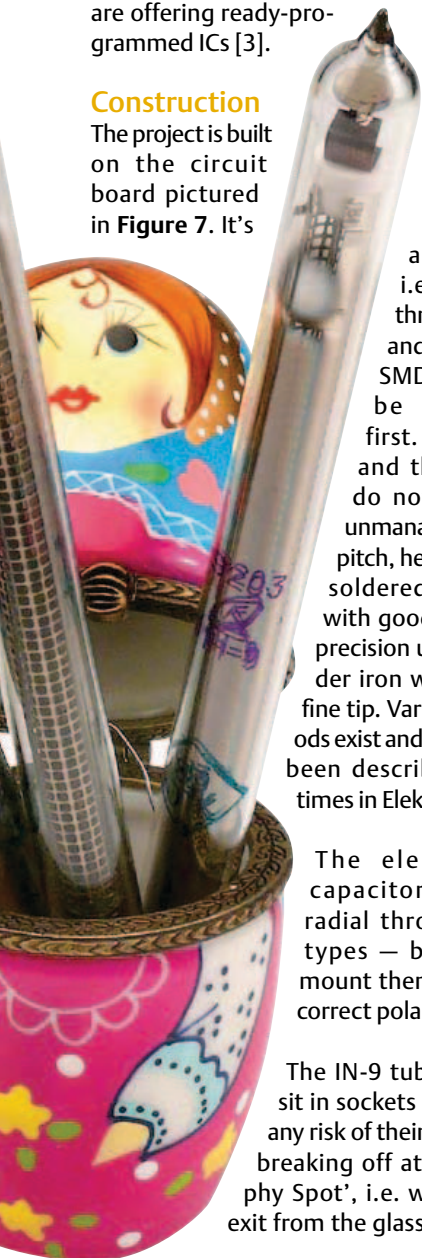


Figure 5. A PSoC is like a sophisticated incarnation of a box of Lego: you get the modules – you decide what's built today – you test – you modify – you show off the finished construction. This image shows the structure programmed to obtain the Nixie VU Meter.

However, if the tubes are secured properly in their final position, their 40-mm long lead wires can simply be extended for connecting to the screw terminals on the board marked V1 and V2, where 'K' is the cathode, and 'A' the anode. Remember, these wires

carry a high voltage and should be rated, secured and protected accordingly.

A short movie showing the Nixie VU Meter in action on a tube amplifier is available on Elektor's very own Youtube channel called

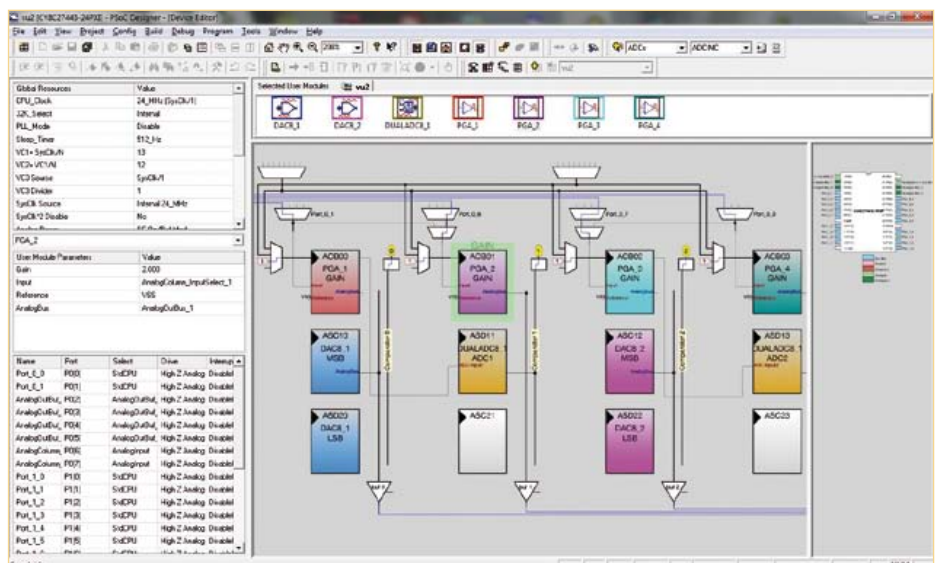
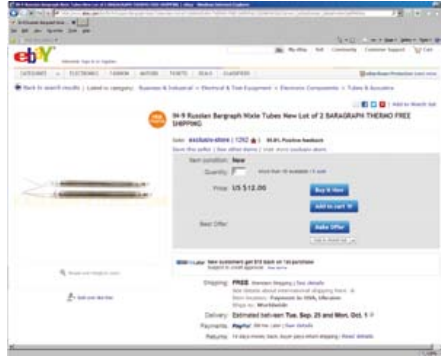


Figure 6. A closer, yet virtual, look at the amplifier blocks inside the PSoC.

‘ElektorIM’ (no typo) [4].
Finally, some friendly yet cautionary advice:

Do not build this circuit unless you understand the dangers of working with high direct voltages. This includes any connection to audio signals in a tube amplifier.
(110744)



Internet Links:

- [1] Nixie Thermometer/Hygrometer, Elektor June 2012, www.elektor.com/110321
- [2] Sputnik Time Machine, Elektor January 2007, www.elektor.com/050018
- [3] www.elektor.com/110744
- [4] www.youtube.com/user/elektorim?feature=results_main

COMPONENT LIST

Resistors

(all fixed Rs SMA shape 0805)

- R1,R2,R5,R8 = 1kΩ
- R3,R6 = 100kΩ
- R4,R7 = 470Ω
- R9 = 2.2kΩ
- R10 = 220kΩ
- R11 = 10kΩ
- R12 = 33kΩ
- R13 = 47Ω
- P1,P2 = 220Ω preset, top adjust
- P2 = 1kΩ preset, top adjust

Capacitors

- C1,C4 = 470nF (SMD 0805)
- C2,C11,C17 = 10μF 16V radial
- C3,C5,C6,C7,C8,C9,C10,C13,C14 = 100nF (SMD 0805)

- C12 = 1μF 250V (SMD shape 1825)
- C15 = 2.2nF (SMD shape 0805)
- C16 = 470μF 25V radial

Inductors

- L1 = 100μH, 860mA, Multicomp MCSDRH73B-101MHF

Semiconductors

- D1,D2,D3,D4 = BAT46WH
- D5 = BYV26
- D6 = BZD27C5V1P 5.1V zener diode
- T1,T2 = MMBTA42
- T3 = IRF640NSPBF
- T4 = BC847
- IC1 = CY8C27443-24SXI, Elektor # 110744-41*
- IC2 = MCP101-475HI/TO

- IC3 = LM324ADT (SOIC-14)
- IC4 = NE555DT (SOIC-8)
- IC5,IC6 = 78L05

Miscellaneous

- K1 = 3.5mm stereo jack, PCB mount (Lumberg 1503 09)
- K2 = 5-pin SIL pinheader, 0.1" pitch (2.54mm)
- K3 = 2-way PCB screw terminal block, 5mm pitch
- V1,V2 = 2-way PCB screw terminal block, 5mm pitch
- 2 pcs IN-9 Nixie tube
- PCB # 110744-1*

* ordering details at www.elektor.com/110744

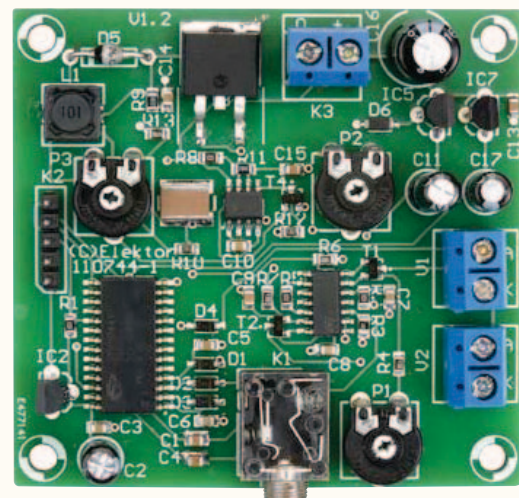
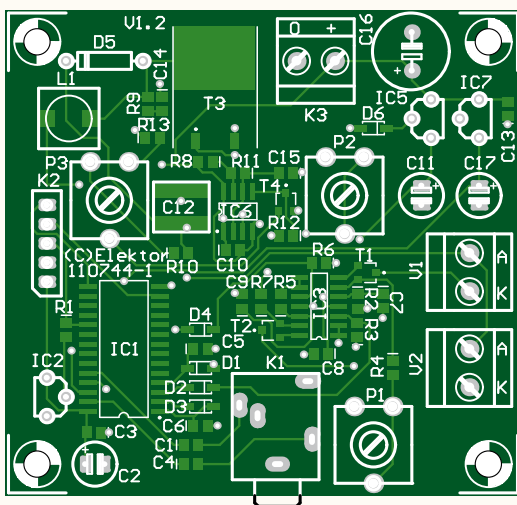


Figure 7. The circuit board designed by Elektor Labs takes a mix of SMA devices (also called SMDs) and through-hole components.



DESIGNSPARK

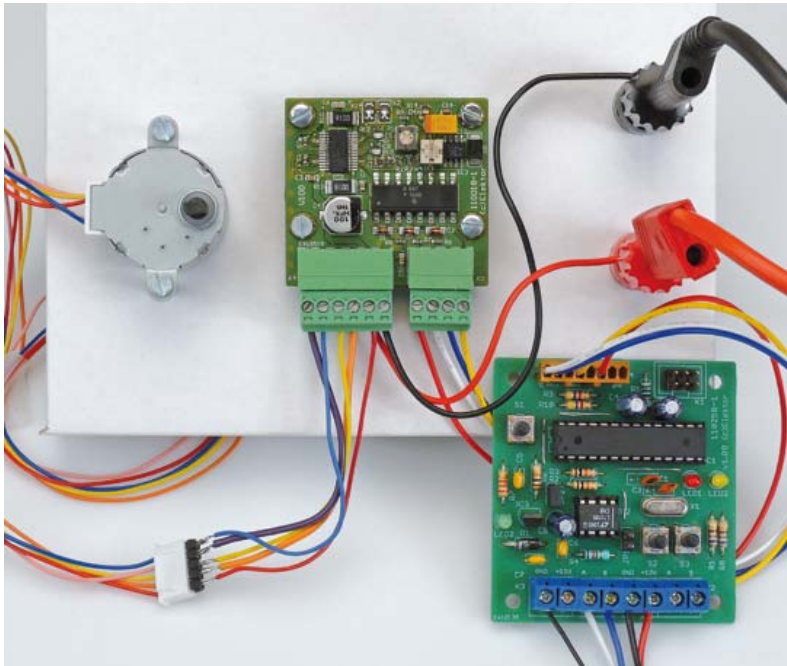
UNIQUE
RESOURCES FROM 

COMING SOON
A NEW ENGINEERING
ECO-SYSTEM

www.designspark.com

A Library for the ElektorBus

The fast track to your own bus firmware



Exactly one year ago we presented software that lets you implement a PC-based controller for your own bus applications quickly and conveniently. Now we are extending this rapid development concept to include the firmware for the microcontroller nodes. The ElektorBus C library relieves developers of the tedious task of dealing with the bus protocol at the bit level. The library even includes its own hardware layer, which makes it independent of your choice of microcontroller. This creates attractive perspectives extending beyond the bus itself.

By Jens Nickel

With tricky projects, simply coming up with a bit of software that actually does what it's supposed to do can sometimes be a major triumph. That's how it was with the ElektorBus. In the course of our series of articles on the bus [1], we first had to develop the protocol functions. For that we usually downloaded firmware to the microcontrollers and used it to test the new functions.

However, in the course of time our expectations grew. For various types of bus nodes, such as experimental nodes with photo-sensors or relays or operational boards, we had to develop specific firmware each time. Although the basic structure was always the same, we first had to laboriously figure out where the code had to be modified in order to do something different, such as transmitting a numerical value instead of receiving one. There was also no getting round a detailed study of the protocol specs when we had to hard-code the bytes of a message to do something like energise or release a relay.

Many advantages

A library can be a real help in such situations. It hides the unchanging basic structure of the bus firmware behind a few function calls, allowing the actual application code to be short and relatively easy to understand. It independently computes the bytes that have to be transmitted on the bus, relieving application developers of the onerous task of bashing individual protocol bits. Finally, it makes the application code independent of the hardware, which means it is independent of the actual bus node boards and microcontrollers. This is important because we keep receiving questions about the possibility of using other microcontrollers, such as PIC devices. Fortunately, the C language has a standardized syntax and the compilers for a wide variety of microcontroller types observe the syntax rules. This makes it possible to write platform-independent application code (**Figure 1**) that can be ported from one type of board to another.

We call the library

Now it's time to leave the theory behind and turn to practical matters. The following lines of code:

Elektor products and support

- Experimental node board (PCB 110258-1 or set of three boards 110258-1C3)
- RS485/USB converter, fully assembled and tested (110258-91)
- Stepper motor driver (PCB 110018-1)

- Free software download

All products and downloads are available via the web page for this article: www.elektor.com/120582



```
TransmitValue(OwnAddress, MasterAddress, 1, 0,
TestLEDStatus);
TransmitValue(OwnAddress, MasterAddress, 2, 0,
SensorValueRaw);
```

put together an ElektorBus message that transmits two numerical values (data units such as this are called *parts*) over two different channels – in the first case the status of the Test LED and in the second case a sensor reading in the range of -1023 to +1023. The instruction

```
SendParts();
```

then transmits these parts over the bus within a message. We no longer need to worry about where the individual bytes must be located in the 16-byte message string or which bits must be set or cleared in the these bytes.

If these function calls seem familiar to you, you have been following our ElektorBus series with close attention. In November 2011 we presented a Javascript library that you can use for developing your own bus controller [2]. At that time we intentionally chose a bus user interface based on HTML and Javascript, so that it can be used on PCs as well as Android devices without any changes.

As much as possible, we have aligned the functions in the new library to the functions in the JSBus Javascript library. This worked out fairly well with some functions, since Javascript has a syntax similar to that of C. However, we had to take the considerably smaller memory of the microcontroller into account. One example of something that is not implemented in the C library is the possibility of collecting parts for different receivers and then automatically coding them to generate several messages. Instead, in this case a maximum of four parts can be put together for sending a message. The library stores these parts temporarily in a simple send buffer in the form of an array.

And it returns our call

There is also a similar buffer for received parts, which is filled by the library when a message addressed to the node has been received. After this the library calls the following function once for each of the parts:

```
void ProcessPart(struct Part part)
{
...
}
```

This function must be defined in the source code of the main bus routine, which is now entirely reserved for the main function and the actual application. Developers can flesh out the function `ProcessPart(...)` with code that responds to the received parts, which could for example be setpoint values for an actuator or some other command from the bus master. Developers can easily access the individual properties of the parts, such as transmitted numerical values, by using an expression such as `part.Numvalue`

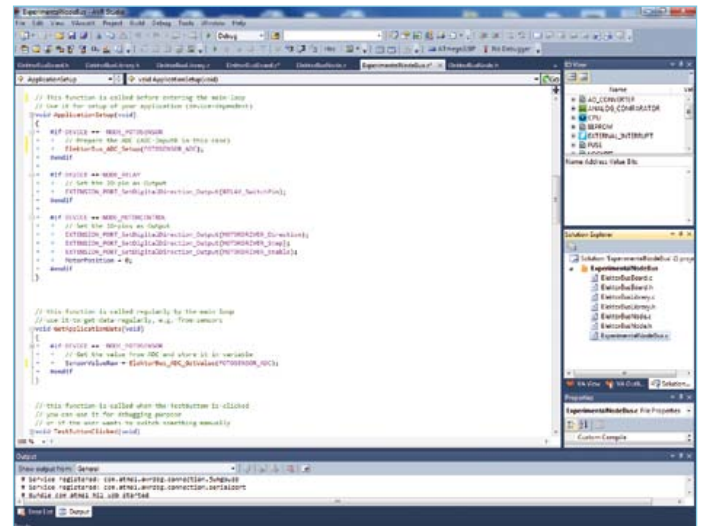


Figure 1. ElektorBus applications can now be developed independently of the type of node board and microcontroller that are used (in this case using AVR Studio 5).

(see the 'Parts' inset).

An illustrative example of application code is:

```
void ProcessPart(struct Part part)
{
if ((part.Channel == 0) && (part.Parttype ==
PARTTYPE_VALUE2))
{
SwitchTestLED(part.Numvalue);
SwitchRelay(part.Numvalue);
}
}
```

Depending on the received numerical value, which in this case can only be 0 or 1, the Test LED on the board is either dark or lit. Another option is to energise or release a relay. The only difference between the functions `SwitchTestLED(...)` and `SwitchRelay(...)` is that the first function is provided by the library (every ElektorBus board should have a Test LED), while the second function is application specific and must be written entirely by the application developer (more about this later on).

There are also other (event-triggered) functions that can be called from the library. For each of these functions, there must be a definition in the main routine or in another project file that includes the ElektorBus C library in the code. The function body can always be supplemented with your own code, but this is not mandatory. The following function is always called when the user presses the Test button:

Listing 1: The main loop of all ElektorBus firmware.

```
int main(void)
{
    ElektorBusLibrarySetup();
    ApplicationSetup();

    while(1)
    {
        GetApplicationData();
        ElektorBusEngine();
    }
}
```

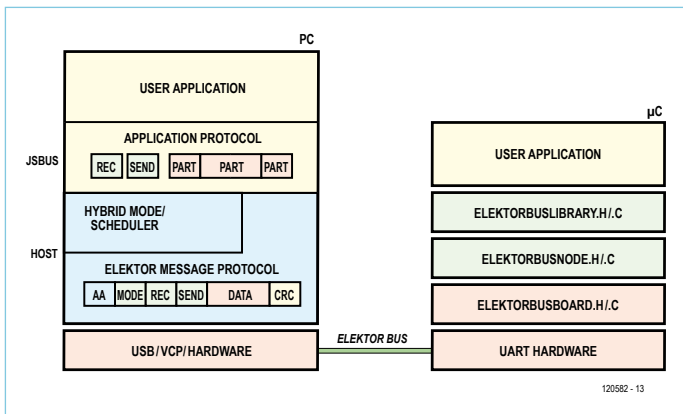


Figure 2. Software for rapid application development. The ElektorBus C library is simultaneously responsible for both protocol layers (message and application protocols). There is also a hardware layer.

```
void TestButtonClicked(void)
{
    ToggleTestLED();
}
```

This example implements a simple debugging feature that can be used to determine whether or not the software is actually running on the node.

The functions presently implemented in the library are listed and briefly described in the ‘Major Functions’ inset.

The bus machine

Nothing has changed in the basic execution of the firmware. First the hardware and the node parameters (such as the address) are initialized, and then the code enters an infinite loop. Only after entering the loop can the node start performing application-specific tasks, such as reading sensor values via the ADC. Next comes the actual bus engine. It initially responds to any messages that have

been received, including messages from the bus scheduler. If the node is a *scheduled* node and the scheduler asks it to send a message, `SendFlag` is set to initiate immediate transmission of a message. Otherwise the node, operating as a *FreeBus* node, transmits only in the *FreeBus* phase and only if it has something to transmit (`SendEventFlag == TRUE`). The algorithm that waits for the message confirmation from the receiver and repeats the transmission if necessary (since collisions can occur in the *FreeBus* phase) is also unchanged (see [3] for the relevant terminology).

All of this is now encapsulated in the library, and the application developer does not need to be concerned with the code for these functions. The developer only needs to know that `SendEventFlag` is set when the function `SendParts()` is called, and that in this situation the message is not sent immediately, but only when the next *FreeBus* phase occurs. A different mechanism is incorporated for periodically queried (*scheduled*) nodes. When such a node receives a message from the scheduler advising that it’s at the head of the queue, the library first calls the function

```
void PreparePartsForScheduledMessage()
{
    ...
}
```

The definition of this function must also be present in the application code. Now the node can collect the current values of the appropriate parts (e.g. sensor values):

```
void PreparePartsForScheduledMessage()
{
    TransmitValue(OwnAddress, MasterAddress, 0, 0,
    SensorValueRaw);
}
```

In this case it is not necessary to call `SendParts()`.

As the library contains the recurrent portions of the firmware, the main loop can be kept nice and short, as illustrated in **Listing 1**. The names of the function calls are self-explanatory. The functions `ApplicationSetup()` and `GetApplicationData()` must be implemented by the application developer. For example, you could initialize an ADC and then periodically read in sensor values.

Hardware functions

You can use the following code to initialize the ADC:

```
void ApplicationSetup()
{
    ElektorBus_ADC_Setup(0);
}
```

The function `ElektorBus_ADC_Setup(0)` initializes the ADC (ADC channel 0). The underlying instructions (specific values must be writ-

ten to specific registers in the microcontroller) are naturally dependent on the microcontroller. Functions of this sort are contained in a separate small library located in the files *ElektorBusBoard.h* and *ElektorBusBoard.c*. By contrast, the source code of the actual ElektorBus C library is located in the files *ElektorBusLibrary.h* and *ElektorBusLibrary.c* (see **Figure 2**).

The reason for this separation is that if someone decides to port their application from one board to another board with a different microcontroller or a different pinout, they can include not only the main source code and the application code in the ported firmware, but also the *ElektorBusLibrary.h* and *ElektorBusLibrary.c* files. They only have to redo the implementation of the functions in the file *ElektorBusBoard.c* file – and with a bit of luck, another developer may have already done this for the board concerned. We can also use the `#define` directives, which are a very nice feature of the C language. The *ElektorBusBoard.h* file contains a series of the directives, such as:

```
#define LED_PORT_OUT PORTD
#define TESTLED_PIN 4
```

Before the actual compilation of the source code, the preprocessor searches the code for the expression `TESTLED_PIN` and replaces each instance with the number '4'. If you (as the developer) always use the defined abstract expressions in your code, the source code of the bus application (and the bus library) can always remain unchanged, regardless of which port pin is used for a particular purpose, such as connection to the Test LED. If you port the application to a different board, it's only necessary to adjust the `#define` directives.

Board specs

A bus hardware library of this sort is especially worthwhile if there is general agreement on a set of basic functions that every ElektorBus board must provide. Along with the previously mentioned Test LED and Test button, we propose the following:

Up to seven additional LEDs connected to a common set of pins (port) named `LED_PORT` (LED1, LED2, etc., where LED0 is the Test LED) and accessible via a register

Up to seven additional buttons connected to a `BUTTON_PORT` (Button1, Button2, etc., where Button0 is the Test button)

A set of pins containing the RX/TX pins and two I/O pins for the *DriverEnable* and *ReceiverDisable* functions for RS485 communication (`ELEKTORBUS_PORT`)

A set of pins whose leads are brought out to a connector for additional hardware (`EXTENSION_PORT`) – this port can comprise up to eight digital I/O pins or optionally up to eight analog inputs

Corresponding hardware-independent expressions for all of these features are defined in the *ElektorBusBoard.h* file. They should be used as much as possible in the application code, for example for controlling the relay in the application-specific function `SwitchRelay()` (see **Listing 2**).

The following expression in that code

Listing 2: Controlling a relay with hardware-independent code.

```
void SwitchRelay(unsigned char relayStatus)
{
    if (relayStatus == 0)
    {
        // Set pin to low
        EXTENSION_PORT_SetDigital_Low(EXTENSION_
            DIGITAL0_PIN);
    }
    else
    {
        // Set pin to high
        EXTENSION_PORT_SetDigital_High(EXTENSION_
            DIGITAL0_PIN);
    }
}
```

```
EXTENSION_PORT_SetDigital_Low(EXTENSION_DIGITAL0_PIN)
```

is a macro that is also defined in *ElektorBusLibrary.h*. It causes digital I/O pin 0 of `EXTENSION_PORT` to be set low, resulting in the release of the relay. As it is unlikely that you will always want to connect the relay to pin 0 of `EXTENSION_PORT` (which is PC0 on the experimental board), you should put your own `#define` directive in the source code of your main routine:

```
#define RELAY_SwitchPin EXTENSION_DIGITAL0_PIN
```

Then you can write:

```
EXTENSION_PORT_SetDigital_Low(RELAY_SwitchPin);
```

If the relay connection is changed to a different port pin during the course of development, you only need to change the previously mentioned `#define` directive.

Node file

We previously mentioned the instruction

```
TransmitValue(OwnAddress, MasterAddress, 1, 0,
    TestLEDStatus);
```

which allows us to access the node address via the variable `OwnAddress`. The bus library also needs to know the node address so that it can decide which messages are addressed to the node. The bus library includes functions for determining the node parameters at software launch time. These functions are located in another set of files named *ElektorBusNode.h* and *ElektorBusNode.c*. At present the node address is entered in the corresponding function as a hard-

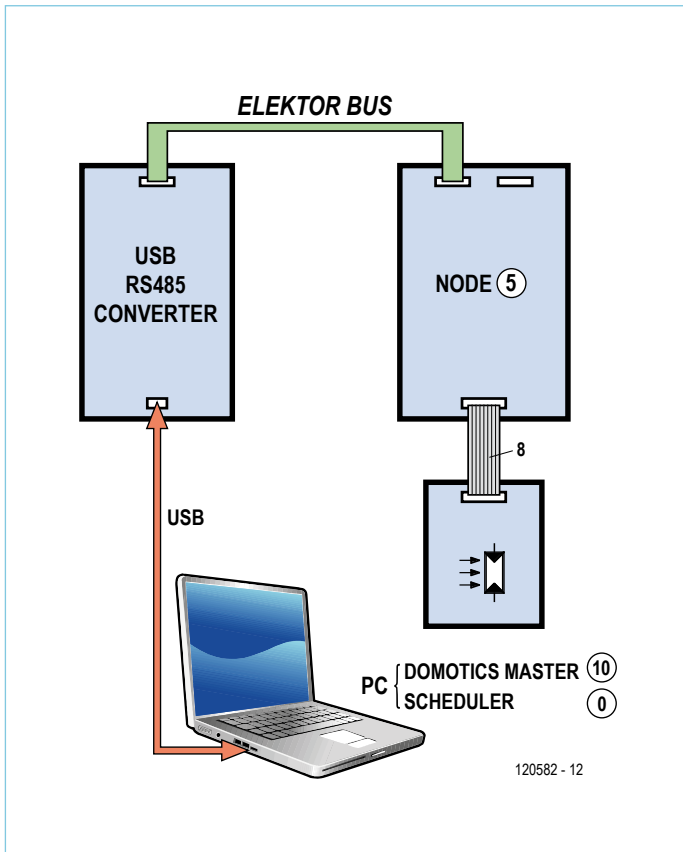


Figure 3. All you need to run the example application is an RS485 to USB converter and an experimental node board, which must be fitted with an ATmega328.

coded value:

```
unsigned char GetNodeAddress()
{
    return 5;
}
```

However, we intend to implement dynamic node addressing at a later date in this project.

Example software

As you might expect, we again have example software [4] for this installment of the ElektorBus series. The project for AVR Studio 5 includes all the files mentioned above, along with an example application. Actually there are three example applications, which you can select as desired by means of a directive at the beginning of the *ElektorBusNode.h* file:

```
#define DEVICE NODE_PHOTSENSOR
```

The node is then assigned the corresponding address (5, 6 or 7) when the software starts up, and the firmware determines whether the node is *scheduled* or not. Of course, you can adapt the addresses in the *ElektorBusNode.c* file to suit your own requirements.

The hardware-dependent files *ElektorBusBoard.h* and *ElektorBusBoard.c* are tailored to the small experimental node board. However, the hex file is too big for the flash memory of an ATmega88, which must be replaced by the pin-compatible (and not significantly more expensive) ATmega328. As the bus library is intended to form the basis for further development, an additional requirement for ElektorBus boards is at least 32 KB of flash memory in the microcontroller; 64 KB or more is preferable. For the example

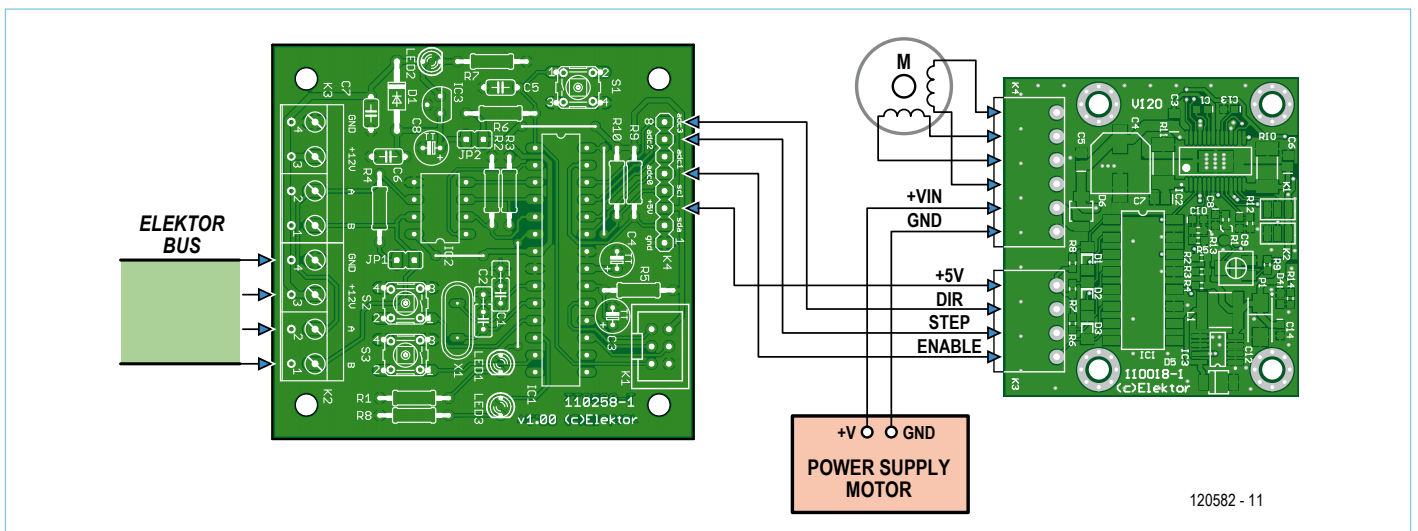
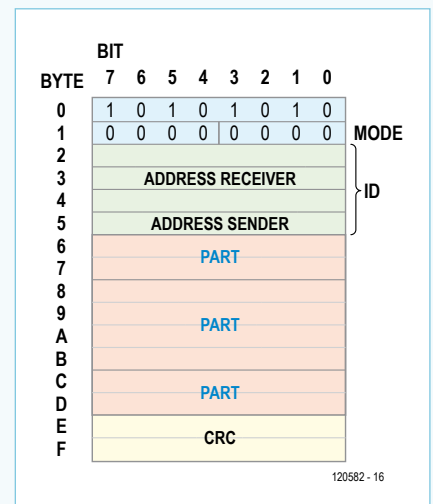


Figure 4. For the third example application, the experimental node board is connected to the stepper motor driver described in the June 2012 issue of *Elektor*. We used a small stepper motor from Ming Jiong [9].

Parts

The ElektorBus protocol specifies that up to four parts (data units) can be transmitted in the eight application data bytes of an ElektorBus message. For example, these parts can be measured values, setpoint values, limit indications or commands for changing the unit or scale. A **part** is represented in the ElektorBus C library by a data structure identified by the keyword `struct`. For each received part, the library calls the function `void ProcessPart(struct Part part)` and passes the data structure corresponding to `part` to the function. The application developer can use an expression such as `part.Numvalue` to access all the properties of a received part:

Valid	Checksum OK? (not yet implemented)
Sender	Transmitter address
Receiver	Receiver address
Channel	Channel number
Setflag	Setpoint or reading?
Ackflag	Acknowledgement message or original message (flag at application level)
Mode	Mode byte of the message (with acknowledge flags at the message level)
Parttype	Type of part, for which the following constants are defined: <code>PARTTYPE_VALUE2</code> , <code>PARTTYPE_VALUE4</code> , <code>PARTTYPE_VALUEFLOAT</code> , <code>PARTTYPE_LIMIT</code> , <code>PARTTYPE_SCALE</code> , <code>PARTTYPE_INTERVAL</code>
Numvalue	Transmitted numerical value (e.g. -1023–1023 for <code>PARTTYPE_VALUE2</code>)
Limit	0 = All OK; 1 = Below lower limit; 2 = Above upper limit
Quantity	Physical quantity (0–127; see [3])
Unit	Unit (0–3; see [3])
Scale	Exponent for scaling (-15 t +15)
Interval	Interval scaling unit for sensor messages (see [3])
Preset	Reserved



software you need only one experimental node board, along with an RS485 to USB converter (see **Figure 3**).

Motor control lite

In order to try out the software, you must connect a photoresistor or a relay to the `EXTENSION_PORT` of the experimental node board (on pin header K4 with 5 V, GND, and PC0–PC6). The circuitry for the photosensor (you can use a potentiometer instead) and the relay extension is described in installment 8 of the ElektorBus series [5]. As a special treat, we have included a third application: a simple controller for a stepper motor. For this you need to connect the Elektor stepper motor driver board [6][7] to the experimental node board as shown in **Figure 4**. Incidentally, this is a sort of sneak preview of an upcoming ElektorBus article in which we will describe a bus-compatible stepper motor driver board. The download [4] also includes a UIBus folder with a very simple HTML file for controlling all of this from a PC. Simply drag it onto the desktop and run `ElektorBusBrowser.exe` [8]. Start by launching the scheduler by means of the corresponding HTML button (**Fig-**

ure 5). After the original firmware has been downloaded to the ATmega328, you should see sensor readings. In addition, the red Test LED on the board should toggle when you press the Test button on the board. The LED status is also shown in the radio button on the PC.

If `NODE_RELAY` is selected in the firmware and a relay is connected, you can energise and release the relay. You can also do this manually on the board with the Test button.

If instead you include `#define DEVICE NODE_MOTORCONTROL` in the firmware, you can cause the stepper motor to move 1,000 steps forward or backward from the PC.

Outlook

There's still a lot to be done, since the bus library is presently only available in an initial version and contains only the most essential functions. Among other things, we should also implement interval functions (for periodic transmission of sensor values from nodes) and in connection with these, functions for the initialization of timers, etc.



Figure 5. A very simple HTML user interface for trying out the three example applications.

This will give us a nice little library that allows us to control a variety of microcontroller functions quite independently of the type of IC that is used. We have already checked the basic feasibility of this approach with an ATxmega 256, which will be fitted on an upcoming Elektor board. This creates possibilities for developing a sort of

Elektor firmware library that can also be used for other projects (not just the ElektorBus). It's a very attractive idea to think that Elektor projects from other electronics enthusiasts could be ported very easily to your favorite microcontroller.

We will keep you posted on this and more at www.elektor-projects.com.

(120582)

Internet Links

- [1] www.elektor.com/elektorbus
- [2] www.elektor.com/110517
- [3] www.elektor.com/Uploads/2012/2/ElektorBusReference.pdf
- [4] www.elektor.com/120582
- [5] www.elektor.com/110428
- [6] www.elektor.com/110018
- [7] <http://rcl.it/schrittmotor>
- [8] www.elektor.com/110708
- [9] www.mingjong.net/index.php?action=prod_detail&p_id=88&lang=en

Major functions in the ElektorBus C library

Functions and variables implemented in *ElektorBusLibrary.c*

`void SetValue(sender, receiver, channel, mode, setvalue)`
 Puts an additional part in the send buffer that sets a target value for an actuator.

`void TransmitValue(sender, receiver, channel, mode, currentvalue)`
 Puts an additional part in the send buffer that represents the actual value of a sensor/actuator.

`void SendParts()`
 Encodes all parts stored in the send buffer into a message and transmits the message.

`void SwitchTestLED(unsigned char ledStatus)`
 Switches the Test LED on (ledStatus = 1) or off (ledStatus = 0).

`void ToggleTestLED()`
 Toggles the Test LED.

TestLEDStatus

This variable holds the status of the Test LED (1 = on; 0 = off).

TestButtonToggleStatus

This variable toggles between 0 and 1 each time the Test button is pressed.

OwnAddress

The address of the node.

MasterAddress

The address of the master node for bus control.

FreeBusPriority

Indicates how the node repeats a failed transmission and after how many *FreeBus* phases.

IsScheduled

Indicates whether the node is scheduled (periodically queried) or not (1 or 0, respectively).

NodeDevice

A number that can be used to distinguish different bus applications (devices) in the same source code (e.g. photosensor = 1; relay = 2).

Functions that must be implemented in the main routine (with or without code)

void ApplicationSetup()

Called before the start of bus communication; can be used for purposes such as initialising an ADC or connected hardware.

void GetApplicationData()

Called periodically; can be used for purposes such as querying connected sensors.

void TestButtonClicked()

Called when the Test button is pressed.

void ProcessPart(struct Part part)

After a message is received, it is decoded into parts (maximum four). This routine is called once for each part. The application can use this for purposes such as responding to a command from the bus master. All properties of the part can be accessed via the variable `part` (see the 'Parts' inset).

void PreparePartsForScheduledMessage()

If a node is *scheduled*, the scheduler periodically requests it to sent messages. When the scheduler receives such a message, the library calls this function. The application code can use this to assemble the current parts to be transmitted, such as readings from a sensor.

Important functions, macros and defines for the hardware layer

void ElektorBus_ADC_Setup(unsigned char ADCchannel)

Initialises the ADC (single-shot, reference voltage = microcontroller supply voltage).

unsigned short ElektorBus_ADC_GetValue(unsigned char ADCchannel)

Causes the ADC to sample an input voltage and return the sample.

EXTENSION_PORT_SetDigitalDirection_Output(Pin)

Sets the mode of a digital pin of EXTENSION_PORT to output. Pin is the physical pin number within the pin set of EXTENSION_PORT (e.g. PC0 with Port C and Pin = 0)

EXTENSION_PORT_SetDigitalDirection_Input(Pin)

Sets the mode of a digital pin of EXTENSION_PORT to input.

EXTENSION_PORT_SetDigital_High(Pin)

Sets a digital pin of EXTENSION_PORT to the High state.

EXTENSION_PORT_SetDigital_Low(Pin)

Sets a digital pin of EXTENSION_PORT to the Low state.

EXTENSION_PORT_Digital_PinStatus(Pin)

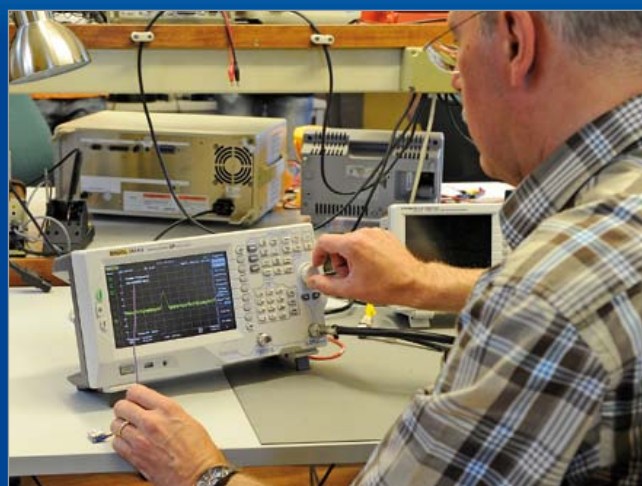
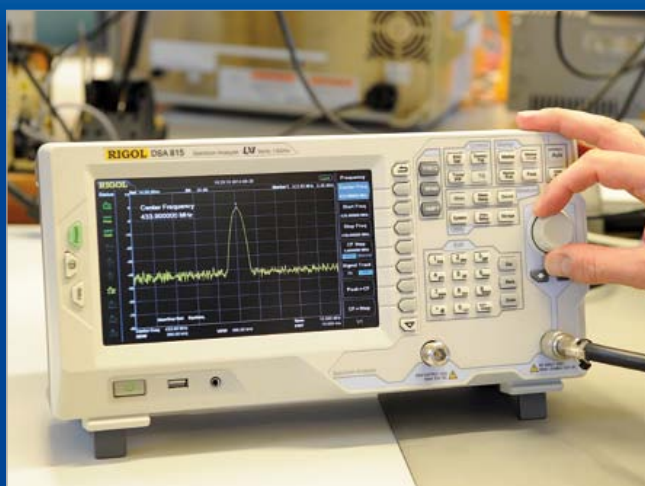
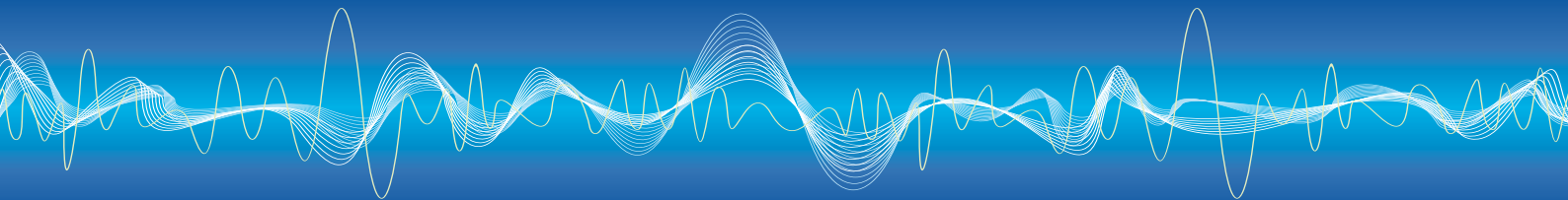
Queries the status of a digital pin of EXTENSION_PORT (==1 High; ==0 Low).

EXTENSION_DIGITAL0_PIN

EXTENSION_DIGITAL1_PIN

etc.

These expressions should be used for Pin instead of numbers, since this makes it very easy to port the code from one board to another board with a different pinout.



Budget Rigol Spectrum Analyzer

By Harry Baggen (Editor, Elektor Netherlands)

“I have this great analyzer, you will have to try it some time. It has an amazing number of features for just over a thousand euro.” This is pretty much the beginning of a telephone conversation I had a while ago with AR Benelux, distributor of, among others, the Chinese brand Rigol. Yes, that did indeed sound very promising: a spectrum analyzer with a range of up to 1.5 GHz and with a built-in tracking-generator and priced around the €/ $\text{£}1,000$ mark (ex VAT) (USA: \$1500). Of course would I like the opportunity to examine this instrument thoroughly, as the former “Elektor bench test” editor I do have a weak spot for test equipment!

An appointment was quickly made and after the production of the September 2012 edition I would have a few spare days to ‘play’ with this instrument, or so I thought. Now, before I realized it I was smack in the middle of the production of the next magazine and the evaluation period for the test instrument was about to expire. On top of it all it happened that Jan Buiting, our only in-house radio amateur, who was keen to take part in the evaluation of the instrument, would be going on holiday. Now, there we were, Elektor Labs engineer Ton Giesberts and the undersigned. How would we deal with this? We are both ‘DC-boys’, which is what radio amateurs would call us; the circuits we normally work on don’t operate above 100 kHz. Nevertheless I wanted to make a serious attempt at evaluating the capabilities of this analyzer. So, we collected several RF circuits that were being worked on in the lab and sat down for the task at hand. We simply started measuring things!

Solid impression

The Rigol DSA815-TG is a spectrum analyzer with a frequency range from 9 kHz to 1.5 GHz. Normally you could easily pay double that for an analyzer with these specifications and that wouldn’t include a tracking generator, while this instrument has that built in as standard. Radio amateurs in particular will drool when looking at all the possibilities that this analyzer has to offer. For its price the instrument can really be considered

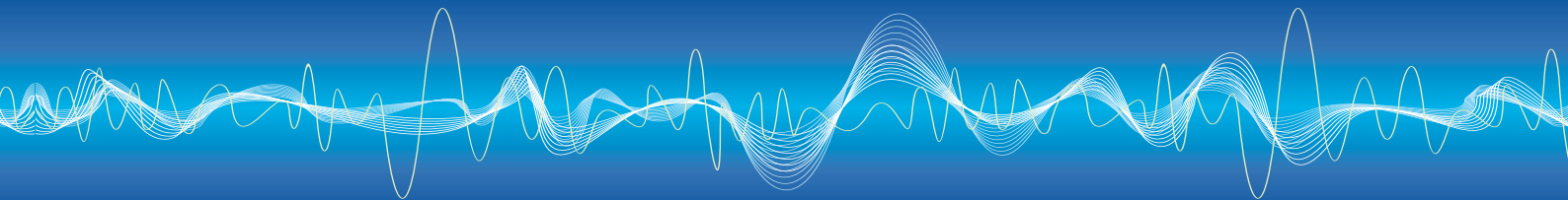
a bargain — finally an affordable analyzer for hobbyists with a slightly longer purse!

The first thing you notice when unpacking the analyzer is its weight. This is, at over 4 kilos (9 lbs), quite heavy, certainly in comparison with a digital oscilloscope at a similar price point. The finish is good, and the appearance is fortunately (unlike many other Chinese manufacturers) very modest with few bright colors. The instrument has both USB-host and USB device connections, so it is possible to plug in a USB memory stick as well as connect the analyzer to a computer. There is even an Ethernet connection which allows you to control of the analyzer remotely over a network.

After switching on, the 8” wide-screen display with a resolution of 800 x 480 pixels turns out to offer an excellent picture quality, it is still clearly readable at large viewing angles. The operating panel is adjacent to the display and has a large number of buttons and a rotary dial. All kinds of settings are indicated on the display around the measuring window. On the right are the menu entries which correspond to an adjacent column of buttons. This design has many similarities with those of other makes of scopes and analyzer. However, every now and then there is the tendency to push on the screen, instead of the adjacent button, but perhaps one day there will be a version with a touch screen. In that respect we are spoiled with the present day mobile phones and tablets. Hopefully AR Benelux will be able to clean the greasy fingerprints off the screen!

Initial measurements

In the beginning you will likely have to struggle through all the menus in order to find what you are looking for. It is not that the menu layout is bad, far from it, but because the instrument has so many features and settings that it is sometimes hard to see the forest for the trees. Though amounting to some 220 pages the accompanying manual on the CD doesn’t offer a great deal of help with this either. Although every setting and function is described in detail, it is still very easy to become lost. However, after working with the analyzer for a few weeks you will be able



to find your way around the menus, this is from experience with the newer scopes in our lab.

As a test, a prototype of a circuit with a small 433-MHz (ISM band) transmitter, for tracking down model airplanes, was connected. The 433-MHz carrier was easily found, the auto-button on the analyzer then resulted in a zoomed-in version of the carrier, positioned in the center of the screen. Switching back to the full analyzer range of 1.5 GHz shows that the second harmonic is only about 12 dB smaller than the base frequency. Hmm, does that meet the standards? We'll have to follow that up.

In particular when measuring the characteristics of filters, is the combination of analyzer and built-in generator ideal. The behavior of a circuit can be quickly established without the need for any other equipment. A normalizing function is very handy here, which lets you compensate for any small deviations in the generator/analyzer combination, before characterizing a filter. The analyzer offers many settings and measuring options when looking at the test results on the screen, such as resolution-bandwidth, sweep time, scale divisions, etc. The display shows a copious amount of information without interfering with the measurements. You can, of course, place various markers for making measurements, and measurement results and settings can be stored and exported. We could continue some more with listing all the features, but we'll leave it at this.

Considering the absence of our resident radio amateur we didn't do any further experiments with the analyzer, but even after this short introduction we have a very positive impression of the DSA815. The comparatively low price, in combination with the many features and built-in tracking generator means that this instrument deserves a prominent place on the wish list of many a radio amateur or RF enthusiast.

p.s. Have a look at the top photo, where the Rigol is next to our own Advantest analyzer. This instrument is at least 20 years old, but the design of the control panel is remarkably similar!

(120560)

Further information:

<http://eu.rigol.com/>
or national distributor

Beep-beep-beep, who's there?

By Thijs Beckers (Elektor Editorial & Labs)

In our Elektor Labs designer Ton Giesberts has been working on a circuit capable of helping to find your lost RC model plane after it made an unfortunate landing somewhere into a corn field or undergrowth. The prototype is built up and running smoothly. But I wouldn't be writing this piece if it wasn't for something strange happening while testing the circuit in our labs. Apart from the receiver functioning correctly in conjunction with the transmitter module, the receiver also picks up a repetitive signal that sounds like three short beeps, with an interval of about a minute or so. When hooked up to a scope, we see the three beeps look like they are Manchester-coded communication signals. Although the receiver is tuned to the 433 MHz ISM band, frankly we are clueless about the origins of these beeps. We surmised it could be the fire alarm sensor installation or the burglar alarm system.

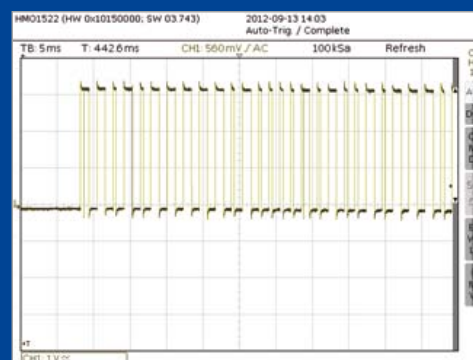
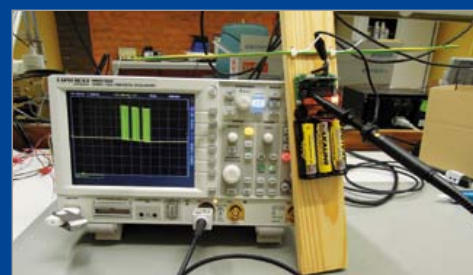
Apart from this, we noticed the receiver picked up the signal generated by car alarm keys as well. When combined with an oscilloscope, this could be 'helpful' when eavesdropping on car alarm systems. Of course the system should be of the older type, where only one-way communication was used (I recently heard of car thieves using a transmitter-equipped RC helicopter they hover over huge parking lots while transmitting bursts of codes from up in the air. A small Wi-Fi camera attached underneath the helicopter provides a nice overview and when there's a car alarm system deactivated by the randomly generated code, the scoundrels simply walk/drive up to the car, open the door and drive away, without having to smash any windows). Another thing we noticed was the noise radiated by TFT screens triggering the LED signal VU meter. It seemed to be originating from the (switching) power supply. Both the screen from the oscilloscope and the computer screen are susceptible. Perhaps a proper metal shielding of the receiver circuit wouldn't be such a bad idea.

In the mean time, our three-beep-per-minute source of interference remained elusive. Perhaps if I could find some time and walk around with a TAPIR [1]...

(120452)

Internet Link:

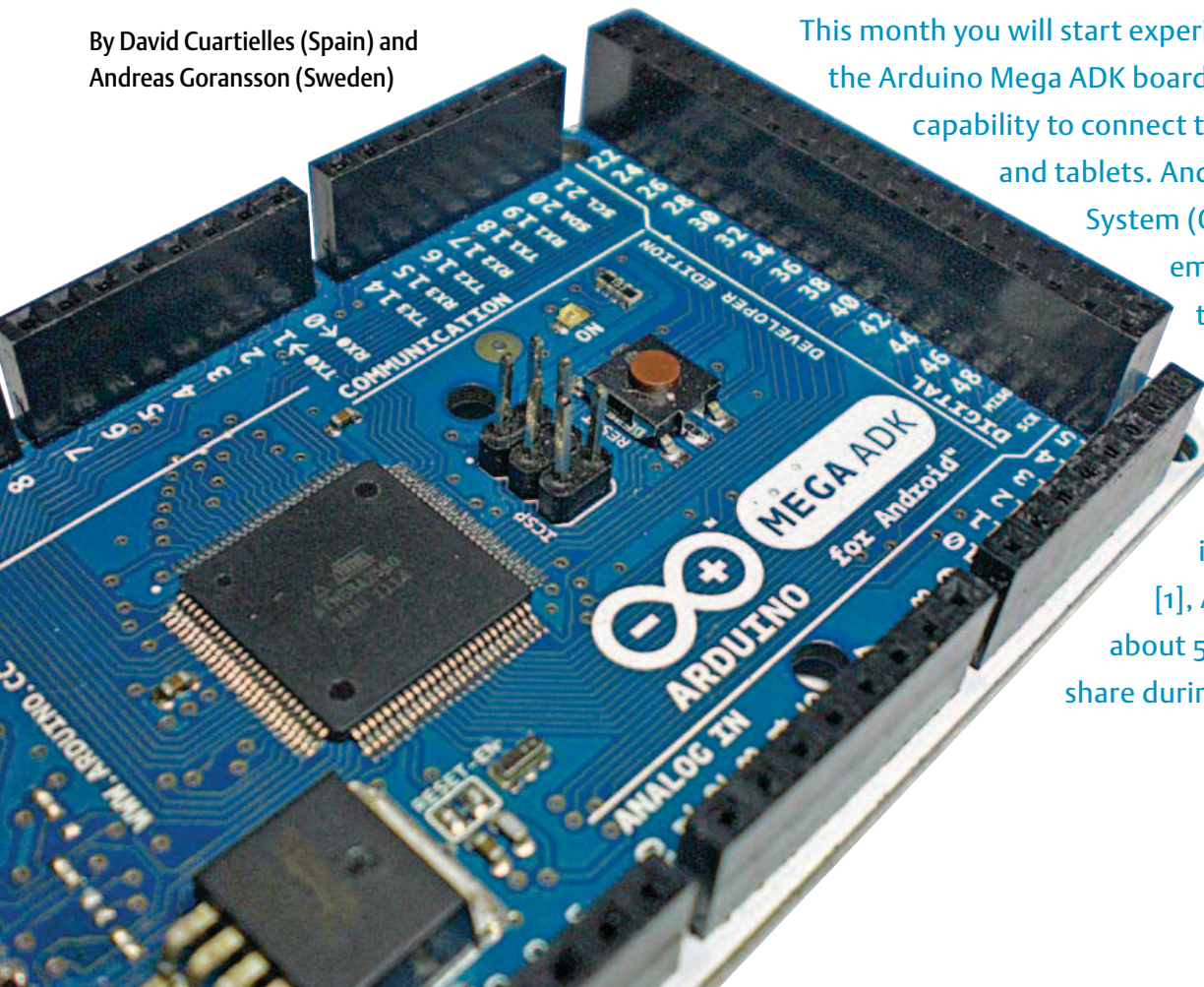
[1] www.elektor.com/120354



Arduino on Course (3a)

Part 3a: Connect the physical world to your Android phone

By David Cuartielles (Spain) and
Andreas Goransson (Sweden)



This month you will start experimenting with the Arduino Mega ADK board, exploring its capability to connect to Android phones and tablets. Android is an Operating System (OS) for mobile and embedded devices that aims at reaching almost every device in the connected world. According to independent sources [1], Android reached about 50% of the market share during 2011.

The Android OS allows anyone to create physical add-ons for smartphones. This article provides an introduction into setting up your system for developing add-ons and Apps connected to them.

Hardware list-o-mania

Here's a shopping list for our experiments:

- Arduino Mega ADK board (pictured above) or Arduino Uno board with USB Host Shield;
- Android smartphone with Android OS 2.3.4 or newer;
- USB cable and microUSB cable.

Note: in this article we will concentrate in how to prepare your system to start developing Arduino applications that will connect to your smartphone. We will give you some pre-made Apps (source included) for you to read/write data from/to the board and for use by, or supplied by, the smartphone.

Tip: before continuing reading the article, we strongly recommend you download all the code used in it, as we are only showing excerpts, due to the length of each one of the examples.

Introduction to Android OS

Android is an operating system (OS) meant for mobile and embedded devices. It is based on Linux and runs a Java-like virtual machine called Dalvik. As in other OSs used in the market of mobile telephony, Android presents a number of differences when compared to Linux.

First of all, as this OS is meant to be used in cellphones, the typical phone operations will have a high priority in the system. For example, unless configured otherwise, a phone call from the missis will stop your device from doing anything else.

Also, all the devices sport a series of pre-assembled sensors like accelerometers, temperature sensors, light sensors, etc.

They offer multiple ways to access the Internet or other devices. Android offers multiple possibilities to do so: GPRS/3G/4G, Wi-Fi, Bluetooth, USB cable, etc.

Finally, by default you will not program native applications, but programs that will run on top of the Dalvik virtual machine. In this way, in terms of syntax your programs will be equivalent to Java. In many cases you can import Java libraries (from source) straight into your phone apps. Android assures a certain degree of portability between ‘terminals’ from different manufacturers. In other words, the same app, once compiled, will most likely work fine for all the different phones as long as they run the same version of the OS.

Android h/w add-ons

Android development team introduced in 2011 the possibility of creating accessory devices (add-on devices) using a series of open source tools. The famous Accessory Development Kit [2] consists of a series of hardware reference designs and the Android Open Accessory Protocol (AOAP).

The hardware designs are derivatives of different Arduino boards. For this article, we’ll focus on using an Arduino Mega ADK, which is compatible with the first version of the AOAP. The code presented in this article should work with later ADK compatible boards (like Arduino Due and the like). It should also be possible to replicate the experiments shown in here using an Arduino Uno and a USB Host Shield for Arduino like the one in **Figure 1**.

One of the main features the AOAP brings into play is ‘App Wakeup’ upon accessory detection. When an accessory is plugged to the phone or tablet, it triggers a call to an application whose name is determined by the accessory. In case the App wasn’t installed in the device, the accessory would also inform about a URL from where to download and install the application.

At the time of writing this article, there were a whole series of boards compatible with the Google ADK besides the Arduino Mega ADK. On the other hand there were no compatible boards with Google ADK2 besides the Arduino Due.

Experimental setup @ the Arduino side

For the sake of simplicity, let’s start by configuring our experimental setup preparing the Arduino IDE enable it to compile AOAP compatible source for the Arduino Mega ADK. Besides having the right hardware, you will need to install:

- Arduino 1.0.1 or newer
- The UsbHost library for Arduino (includes the AndroidAccessory class). If you never installed a library for Arduino before, a how-to follows in the next section.

Thanks to the way AOAP works, once you have the right code running on the Arduino Mega ADK, even if you have no applications in your phone ready to take advantage of the accessory, your Android

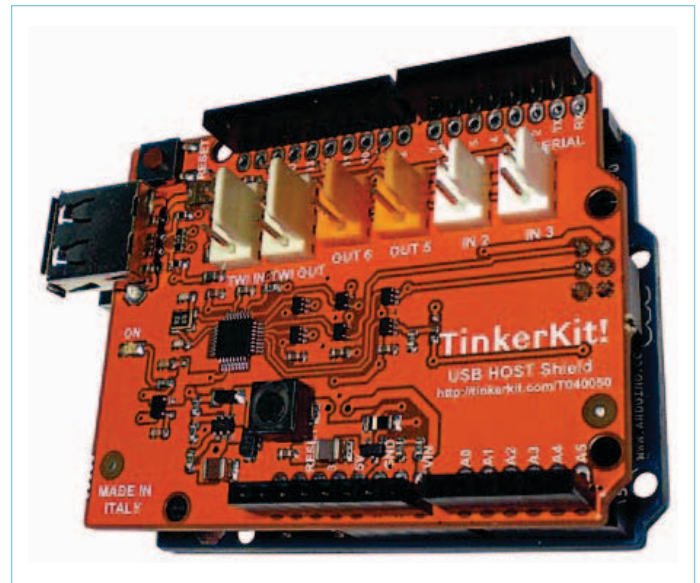


Figure 1. : Arduino Uno connected to a USB Host Shield.

device will connect to the Internet and offer you a website where it will be possible for you to download the App used in this example, as is being done in **Figure 2**.

Note: For this automatic App installation to work, you need to activate the option that allows your phone to install applications from outside the official Google Play Market. Follow Settings → Applications and make sure the option “Unknown sources” is active. Also make sure your phone has a data connection either over Wi-Fi or the phone network to download the App from our servers.

Installing the USB Host library for Arduino

At the time of writing this article the Arduino IDE did not include the USB Host library we are presenting. Therefore when you examine the “Sketch / Import Library” menu you don’t see the library named ‘USB Host’ in the list — you will have to download the library from the official Arduino website hosting this project [3] and install it.



Figure 2. Arduino Mega ADK connected to a PC and to a phone.

Essentially, adding a new library to the IDE is done by creating a folder called 'libraries' inside your sketchbook and uncompressing the file you got from the website [3] directly there. After that, you should restart the Arduino IDE and the library will then show up in the above-mentioned menu. If you had a previous version of this library or one obtained from a different website than the Arduino one, we strongly recommend you uninstall it before bringing in this new version, as they might be incompatible. With the library you will be installing not only the code to execute different commands on the modem, but also a series of examples that will allow you to:

- test whether the ADK mode is working properly;
- send digital/analog values from the board to the phone;
- receive values from the phone into the board;
- debug the different types of USB devices connected to the Arduino Mega ADK board.

The way to access the examples is very simple, just use the menu to navigate through: "File / Examples / USB Host".

Boot an App – prep work

To check whether things are working fine for you, let's make the easiest example possible. You are going to upload a program to your Arduino board that will make your phone download a very simple App called Elektor_MIAU, as well as show something on the screen. (code listing 1)

Once you load this code into your Arduino Mega ADK, you should open the Serial Port Monitor in your IDE to monitor what is happening. When connecting your phone via the micro USB cable to the Arduino board, the Serial port will register something similar to what is shown in **Figure 3**.

At the same time, the phone will detect it has a compatible accessory hooked up and, since you will have no application installed in it for it to handle the data, you will get a message indicating where you can get the proper App from the Internet. You should get something similar to **Figure 4**.

You will notice the URL mentioned in the



Figure 3. Screenshot from the Serial Port Monitor when linking an Android device to your Arduino Mega ADK.

phone's screenshot is the same one as in code listing 1. In other words, you determine in your Arduino code, where the App is located on the Internet. In this case, we have set up a website for you to check all the examples from your phone. You should open the browser in your Android device and see the site as in **Figure 5**.

You will have to download it and install the App by clicking the link underneath the image It will first get downloaded and then effectively loaded into your device. Next you need to click on the file (called "Elektor_MIAU.apk") which will install it. If the App is already in your phone, you will instead get an invitation to load the right application when plugging in the accessory again (**Figure 6**).

Once you have given permission for the App to boot, it will present a GIF image as part of a loop (**Figure 7**). We thought it would be fun to show that you can actually use some graphic capabilities on your Android terminal and stay clear of the corny "Hello World" text on the screen. However, this App is not using any information from the Arduino board, nor is it sending anything back to it — all it does is allow you to check whether your phone/tablet supports Accessory Mode.

Note: It should be mentioned here that it is possible to have more than one App to handle the data produced by an accessory. In that case, the dialog window will offer more than one option to choose from.

Experimental setup @ the Android side

Building Android Apps can be complex. It requires installing a long list of different software packages coming from different locations. We wrote a *Getting Started Guide* revising all the different packages you need for getting the easiest experimental setup possible [3].

Probably the easiest way to develop a simple Android application implies using the open source software tool called *Processing* [4] by C. Reas and B. Fry. This software offers a simplified IDE that allows Java applications to be compiled. The latest version also compiles code for Android phones, and Javascript for websites. We have developed an Add-on tool for the Processing IDE that compiles and uploads code to your Android device for controlling accessories.

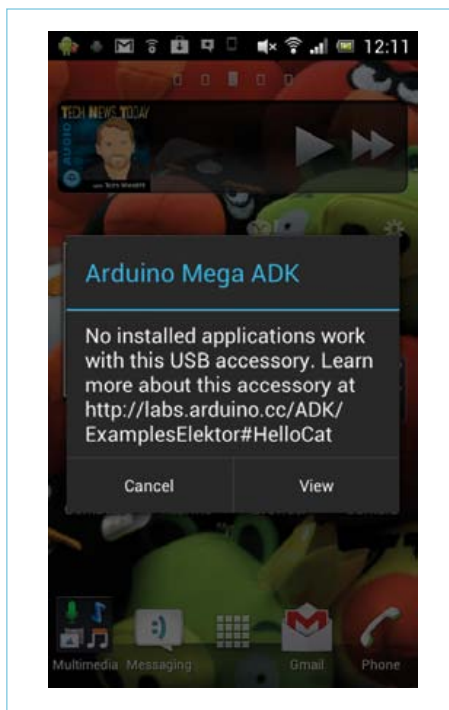


Figure 4. Screenshot taken from an Android phone showing the warning about missing an App to handle an accessory.

(code listing 1)

```
// libraries
#include <AndroidAccessory.h>

AndroidAccessory acc(
  "Arduino, SA",
  "Miau",
  "Arduino Mega ADK",
  "1.0",
  "http://labs.arduino.cc/ADK/
  ExamplesElektor#HelloCat",
  "0000000012345678");

// variable to detect whether the Android device was
// plugged recently
boolean wasConnected = false;
```

```
void setup()
{
  // initialize serial communications
  Serial.begin(115200);
  Serial.print("\r\nStart");

  // initialize the accessory object
  acc.begin();
}

void loop()
{
  if (acc.isConnected() && !wasConnected) {
    Serial.print("Accessory connected. ");
    wasConnected = true;
  }
}
```

Let's summarize what you need to have in place for you to replicate the experiments discussed in this article.

- Android SDK with all the updates up to the latest version of the APIs;
- Processing 2.0a8 or newer;
- The Arduino ADK Tool for Processing;
- Eclipse [5] + ADT plugin (both optional and not used here).

As with almost everything within the world of software, there are many possible tools to write your applications in. The Android SDK is an external tool to your preferred code-editor that will compile,

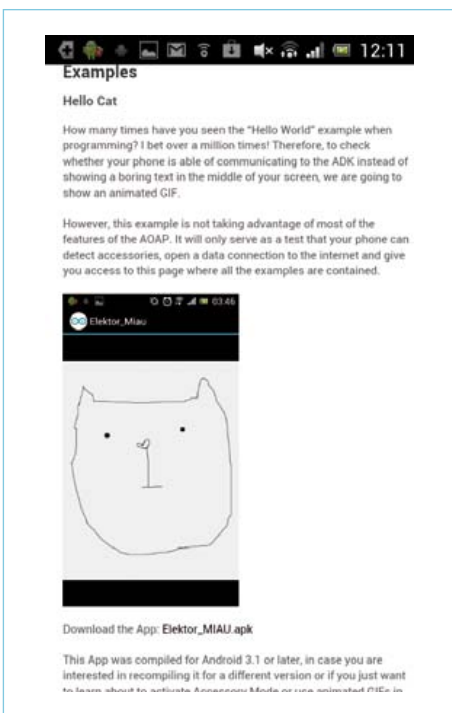


Figure 5. Android phone showing information about the App.

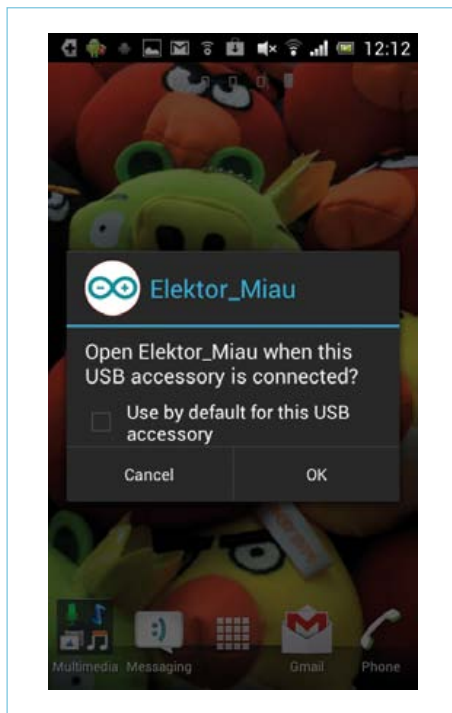


Figure 6. Your Android phone asks for confirmation before loading an App to handle an accessory.

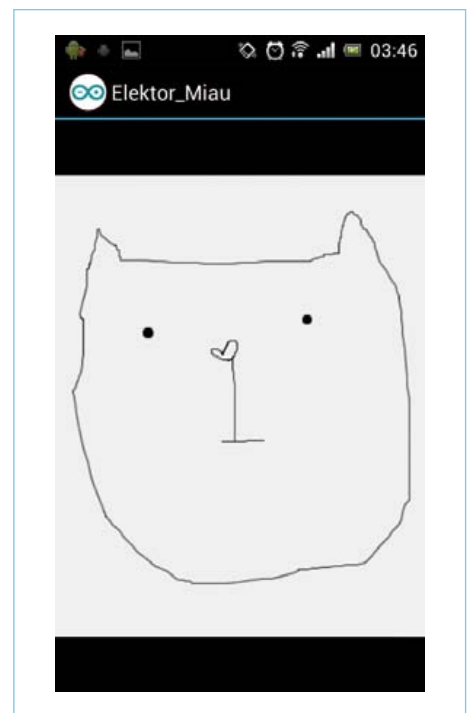


Figure 7. The MIAU App running on Android.

(code listing 2)

```
void setup() {
  // make sure the screen will have
  // fixed orientation
  orientation(PORTRAIT);
}
```

```
void draw() {
  // draw from the center of the shape
  rectMode(CENTER);

  // make a 50x50 pix square using the
  // default color scheme
  rect(width/2, height/2, 50, 50);
}
```

(code listing 3)

```
AndroidAccessory acc(
  "Arduino, SA",
  "Elektor_Rectangle",
```

```
  "Arduino Mega ADK",
  "1.0",
  "http://labs.arduino.cc/ADK/
  ExamplesElektor#Rectangle",
  "0000000012345678");
```

link libraries, simulate, compress and sign your Android applications for a specific version of the OS.

On the other hand, Processing is a really good tool if you are starting to program, and therefore we created an Add-on for it as explained in the *Getting Started Guide* [3]. But if you really want to use a professional set of tools to write Android Apps on, you should consider Eclipse [5], the open source IDE, and the associated tools. Android’s developer site [6] explains how to install the whole Eclipse-based toolchain step by step. I will assume you managed to install Processing and all the other tools without much trouble. All of them are cross platform and should work for virtually any version of your OS. So now we will write the code for Android using Processing.

Your first App

Before even thinking of making an App to control your accessory, let’s make a very simple App using Processing to show something on the phone’s screen. For you to start writing Android Apps you will need to have the Android SDK installed and the Processing software configured to compile Android applications (**Figure 8**). Your IDE should have a green color scheme.

Let’s start by making an App and running it on the device straight away. The following bit of code will show a square in the center of the screen. We will later add the code for the square to change depending on sensor



Figure 8. Screenshot of Processing running Android mode; note the Mode button in the upper right corner.

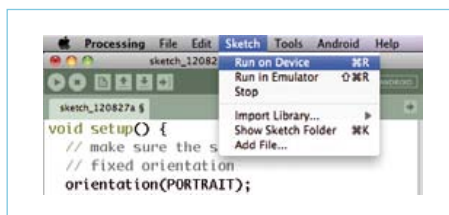


Figure 9. Running the App directly on your phone.

data captured by the Arduino Accessory. (code listing 2)

Processing is a tool aimed at graphic artists and therefore it uses a paradigm where instead of having a ‘loop’, the main function in the program happens to be called ‘draw’. The code you write inside Processing’s IDE is Java, but it hides all the complex operations behind it. To check whether you have everything installed properly, follow the menu Sketch → Run on Device, as in **Figure 9**, this should compile the code and upload it automatically to the phone.

Once the App boots, you should see an image like **Figure 10** on your phone’s screen. The App will boot directly and it will remain there. You will be able to look for it on your App menu and run it as many times as you want to.

Note: At the time of writing, Processing didn’t allow signing Apps. The process of signing an App is what validates it for distribution to other devices. Any Apps created with Processing will only work on the devices you upload them to directly. If you wanted to send your Apps to a friend, you should compile them using Eclipse, which is much more complicated at this point and will not be explained here. For example, the “Elektor_MIAU” application is signed, allowing anyone to download it from the Internet.

Make your App boot upon accessory connection

Now you made a very simple application

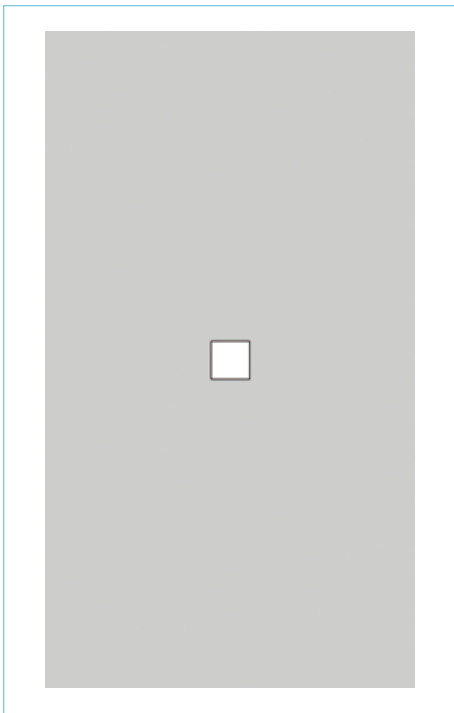


Figure 10. Screenshot App active on the phone.

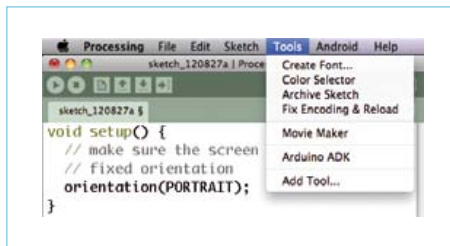


Figure 11. Screenshot of the Tools menu for Processing's IDE.

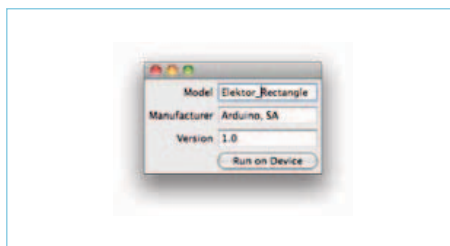


Figure 12. Dialogue window to configure the App to be booted for the accessory.

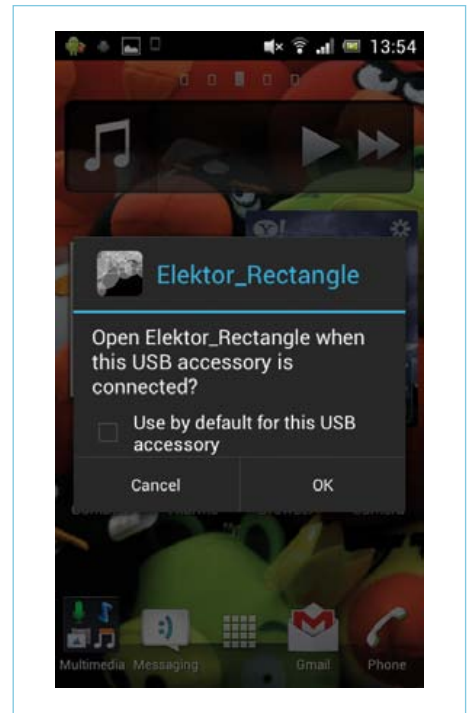


Figure 13. Dialogue window showing the notification for opening your Rectangle App.

that runs on your phone, it's time to make it boot when you plug an accessory to it. Here is when the Arduino ADK Tool for Processing comes into play. If you installed it as explained at [3], no problems arise if you just call it from the Processing IDE. The new tool needed for uploading accessory code to your Android device should show up under the Tools → Arduino ADK menu as shown in **Figure 11**.

When choosing that option from the menu, the IDE will open a dialog window allowing you introducing the name of the app and the manufacturer of the accessory (**Figure 12**). That is what Android Accessories need to identify the App to connect to. You will have to make this match with the code on Arduino, as shown below in code listing 3.

You could now modify code listing 1 where we were calling the HelloCat application to call this new one. (code listing 3)

Note that this is just a snippet of code to modify code listing 1 to make it call your first Accessory App for Android. Having done that and uploaded the code to your Arduino Mega ADK, whenever you plug your phone to the board, it will boot your drawing rectangle App, as pictured in **Figure 13**.

Next month

This was a brief introduction to creating accessories for Arduino and Apps on Android that connect to them. However, this is just an introduction. Follow us next month to see how to send data from Arduino to Android and vice-versa.

(120539)

Acknowledgements

Our thanks are due to the team at Circuits@Home for their first work on the USB Host Library for Arduino and Philip Lindsay for his work on debugging the initial collection of official Arduino Libraries for Android.

About the co-author

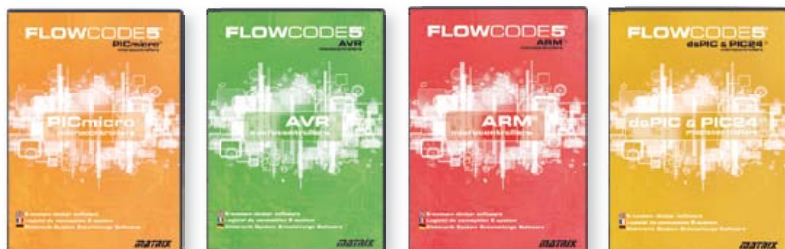
Andreas Goransson is an interaction designer based in Malmö. Currently he is teaching software at Malmö University and he is writing a book about Android and Arduino together with David Cuartielles.

Internet Links & References

- [1] Worldwide smartphone market, by operating system: <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#smartphoneos>
- [2] Accessory Development Kit: <http://developer.android.com/tools/adk/index.html>
- [3] Arduino with Android at Arduino.cc: <http://labs.arduino.cc/ADK/AccessoryMode>
- [4] Processing Project: <http://processing.org>
- [5] Eclipse: <http://eclipse.org>
- [6] Android Guide to installing the Android SDK using Eclipse: <http://developer.android.com/tools/index.html>

Developing and learning

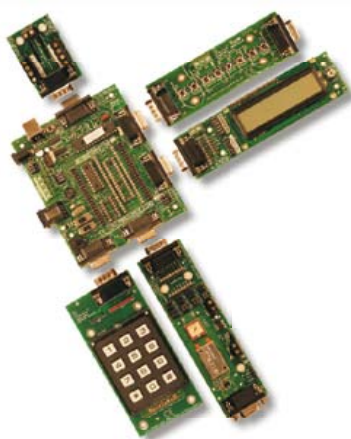
FLOWCODE5



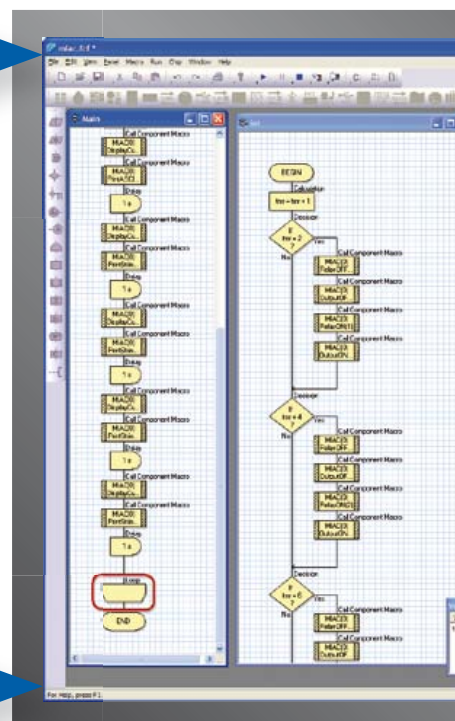
Flowcode 5 is one of the world's most advanced graphical programming languages for microcontrollers (PIC, AVR, ARM and dsPIC/PIC24). The great advantage of Flowcode is that it allows those with little to no programming experience to create complex electronic systems in minutes.

www.elektor.com/flowcode

... for electronics



E-Blocks are small circuit boards each of which contains a block of electronics that you would typically find in an electronic or embedded system. There are more than 40 separate circuit boards in the range; from simple LED boards to more complex boards like device programmers, Bluetooth and TCP/IP. E-blocks can be snapped together to form a wide variety of systems that can be used for teaching/learning electronics and for the rapid prototyping of complex electronic systems. Separate ranges of complementary software, curriculum, sensors and applications information are available.



... for industrial control



MIAC (Matrix Industrial Automotive Controller) is an industrial grade control unit which can be used to control a wide range of different electronic systems including sensing, monitoring and automotive. Internally the MIAC is powered by a powerful 18 series PICmicro device which connects directly to the USB port and can be programmed with Flowcode, C or assembly. Flowcode is supplied with the unit. MIAC is supplied with an industrial standard CAN bus interface which allows MIACs to be networked together.

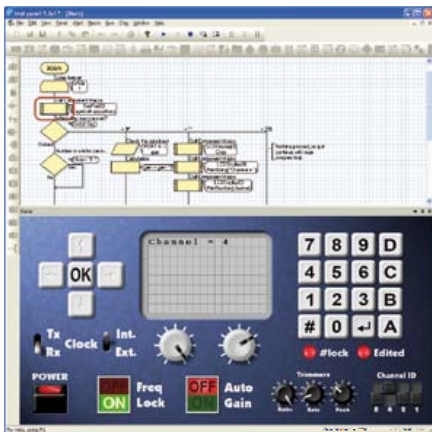
Flowkit

Flowkit provides In Circuit Debugging for a range of Flowcode applications for PIC and AVR projects:

- Start, stop, pause and step your Flowcode programs in real time
- Monitor state of variables in your program
- Alter variable values
- In circuit debug your Formula Flowcode, ECIO and MIAC projects



ng with Flowcode 5



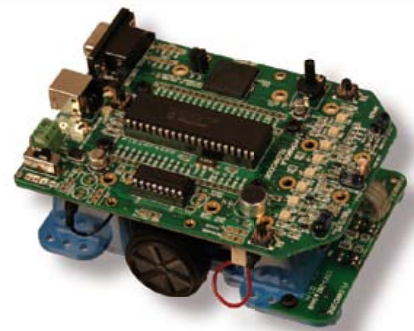
New features in Flowcode 5

Flowcode 5 is packed with new features that make development easier including:

- New C code views and customization
- Simulation improvements
- Search and replace function
- New variable types and features, constants and port variables
- Automatic project documentation
- New project explorer makes coding easier
- Implementation of code bookmarks for program navigation
- Complete redesign of interrupts system allows developers access to more chip features
- Compilation errors and warnings navigate to icons
- Disable icons feature
- Improved annotations
- Improved links to support media
- Support for MIAC expansion modules and MIACbus

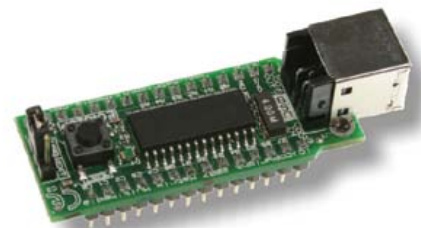
... for robotics

Formula Flowcode is a low cost robot vehicle which is used to teach and learn robotics, and to provide a platform for competing in robotics events. The specification of the Formula Flowcode buggy is high with direct USB programming, line following sensors, distance sensors, 8 onboard LEDs, sound sensor, speaker and an E-blocks expansion port. The buggy is suitable for a wide range of robotics exercises from simple line following through to complete maze solving. E-blocks expansion allows you to add displays, connection with Bluetooth or Zigbee, and GPS.



... for USB projects

ECIO devices are powerful USB programmable microcontrollers with either 28 or 40 pin standard DIL (0.6") footprints. They are based on the PIC 18 series and ARM 7 series microcontrollers. ECIO is perfect for student use at home, project work and building fully integrated embedded systems. ECIO can be programmed with Flowcode, C or Assembly and new USB routines in Flowcode allow ultra rapid development of USB projects including USB HID, USB slave, and USB serial bus (PIC only). ECIO can be incorporated into your own circuit boards to give your projects USB reprogrammability.

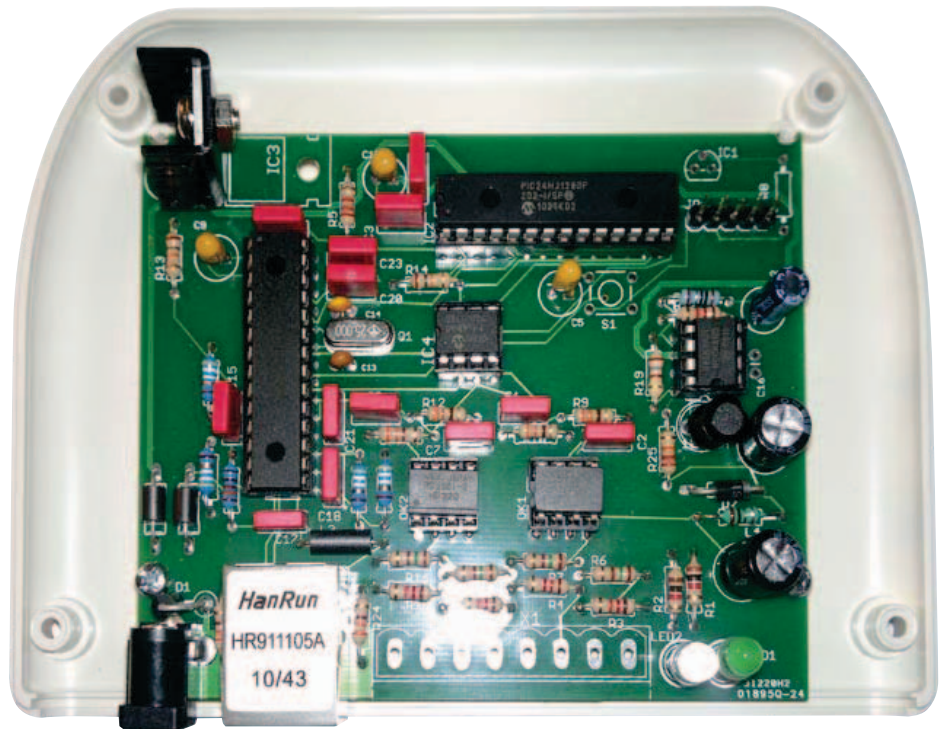


More information and products at:
www.elektor.com/eblocks

Electricity Meter on the Web

An openEnergy, openSource gateway for S0-type power meters

Reducing each individual's carbon footprint has been high on the Global Agenda for the last couple of years. There are many ways to achieve that, but first and foremost you need to know how much electricity is passing through your meter and emptying your wallet 24/7/365. Why not let the worldwide web and some clever technology help you keep tabs on your electricity consumption?



By Søren Mikkelsen (Denmark),
with support from
Hans Henrik Skovgaard (Denmark)

In a standard household you use a mix of different energy sources, where electricity is one of the most used. Some households are in fact powered by electricity only. Households connected to the AC power grid already have a power (electricity) meter, which can be monitored in different ways depending on the technology used. Some old power meters are 'Ferrari' devices with a mechanical counter for human reading only. More advanced power meters exist, capable of sending the power consumption data to the electricity provider. In some cases it can be beneficial to be able to measure power consumption of individual devices in the household. Just recently it has been advised to install sub-metering on heat pumps in order to be able to track their

electrical energy consumption, the ultimate aim being to discover if the heat pump is set up and operating correctly.

S0 interface

Various ways of measuring the electrical power consumption exist. This article will focus on power consumption data using a commercial power meter (or 'meter head') with an S0 interface and a specially designed Ethernet power meter gateway, for home construction.

The S0 interface is defined in the DIN 43864 standard. It describes a current interface for transmitting pulses from a pulsing meter to a tariff-metering device. The maximum voltage that can be applied to an S0 output is 27 volts DC, and the maximum load at the output is 27 mA. The duration of the pulse is typically around 30 ms.

The interface is not only used for electricity power meters but also for water meters and gas meters. The amount of energy (or water, gas) consumed and represented by

a pulse, is defined by the device transmitting the pulses. In the case of electricity that could be 1000 pulses per kilowatt-hour (kWh).

Using an S0 based sub meter usually calls for the meter to be inserted between the AC grid and the consuming device. In most countries this setup requires an authorised electrician due to the risk of working with electricity.

Some power meters installed by electricity providers already have an S0 interface. If the power meter contains an LED that flashes on the front of the power meter, there is a high probability that the meter contains an S0 interface.

Storing power consumption data

Counting the number of pulses is not enough — you also need to have a place to store the data being collected.

Storing the client's consumption data can either be done locally or on servers hosted outside the household (cloud based). The

amount of data and the bandwidth required for storing power consumption are not a big deal by today's standards. Here we've elected to use an open source solution to store and present power consumption data. The solution originates

from the openenergymonitor project [1]. That project has created a system based on PHP and the MySQL database. In terms of availability, you are unlikely to find a better solution. It's open source, so you have full access to the source. But you don't need

be a web-tool expert to use it. You can also use it from a hosted service for a modest fee (www.emoncms.org). But you can also host it on your own server or at any 'web hotel' provider that supports PHP and MySQL.

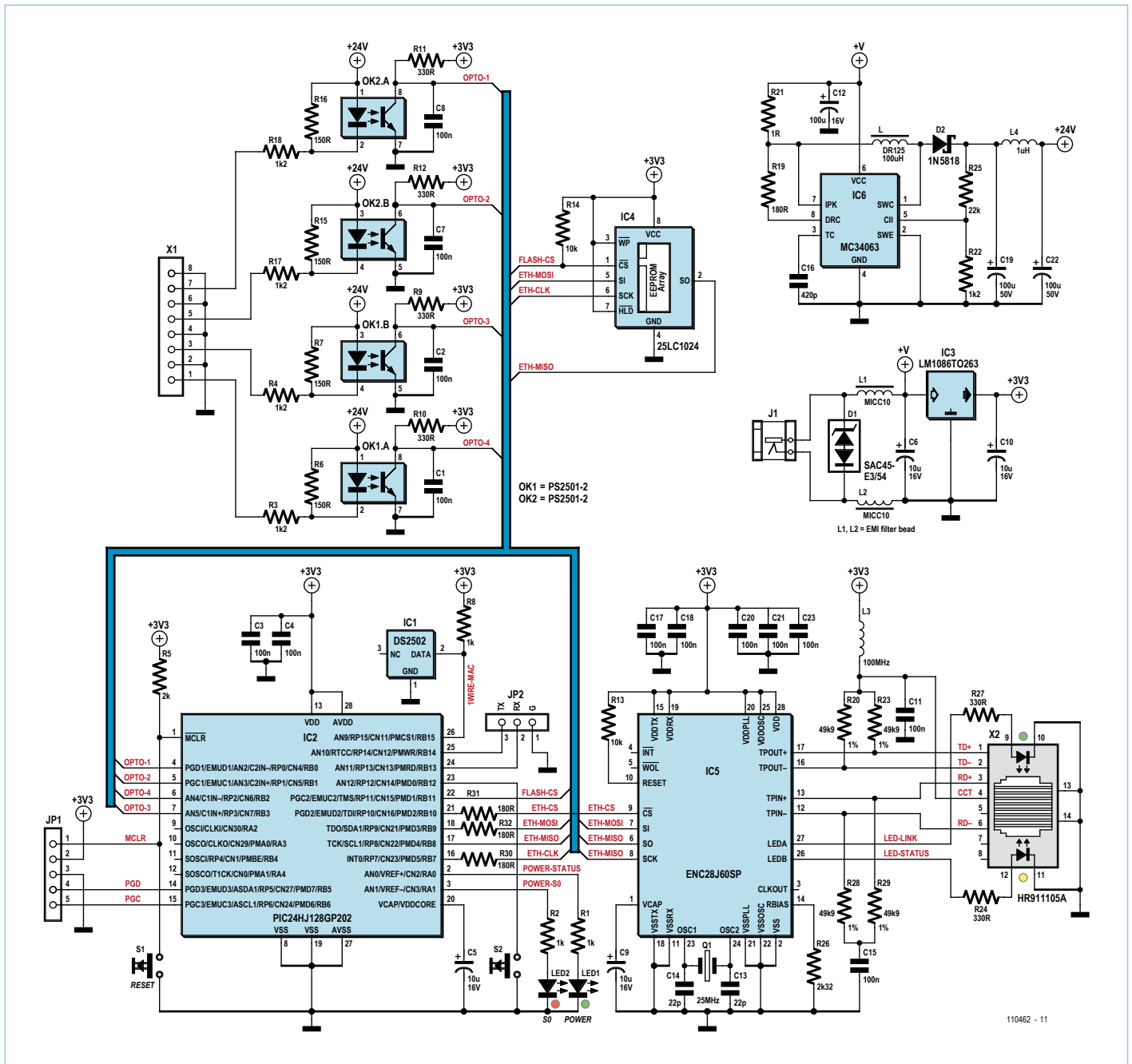


Figure 1. Circuit diagram of the Puls0 gateway for S0-compatible electricity meter modules. Four S0 inputs are provided.

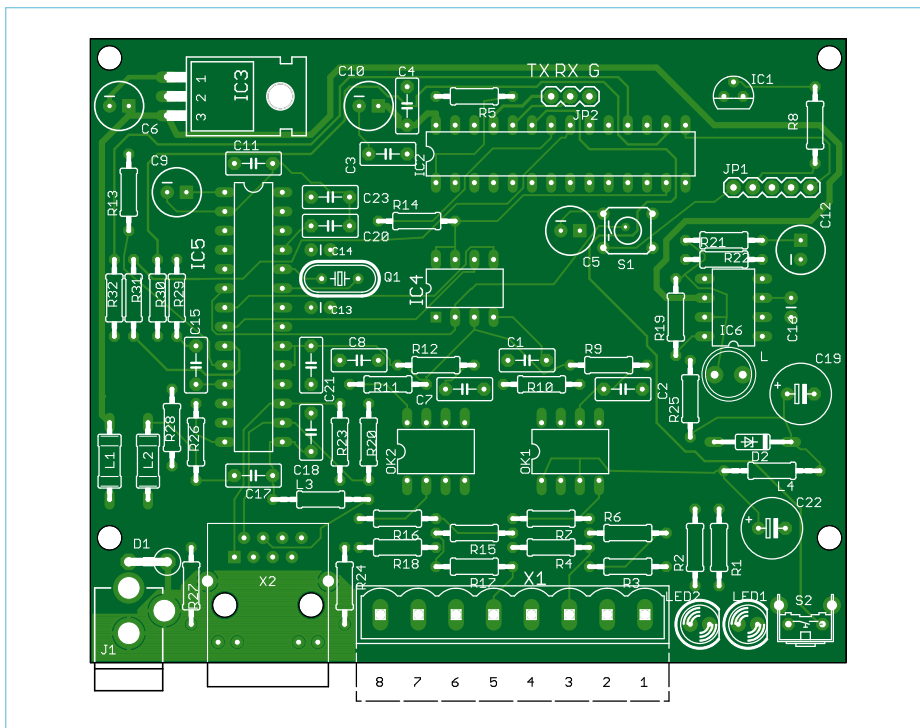


Figure 2. Printed circuit board design for the ‘electricity meter’s gateway to the web’. Remarkably (happily?) there are through-hole components only on the board.

The Puls0 Powermeter Gateway

The initial requirements for the Puls0 gateway were:

- four S0 inputs;
- 5 volt supply;
- Ethernet connectivity;
- OpenEnergyMonitor EmonCMS support;
- Visual indication on received pulses.

The schematic of the gateway appears in **Figure 1**. First off, unlike many other circuits designed today this design is made with leaded components only. This makes it possible for people with big thumbs and weak eyes (the author included) to solder the circuit without significant problems. All the prototypes of this circuit have been manufactured using PCBs from a professional board factory.

The Puls0 gateway consists of the following hardware blocks. Power supply, local storage, microcontroller, S0 interface and Ethernet interface. Let’s have a look at each of these.

The **power supply** around IC3 converts the input voltage to the required 3.3 volts used by the local storage, microcontroller and Ethernet interface block. The Ethernet circuit by itself requires up to 500 mA. An

LM1086-3.3 regulator handles the conversion. A heatsink is also required since the regulator gets rather hot.

The **local storage** block consists of a Microchip SPI based serial EEPROM, IC4. It handles the local software settings, but also contains the web pages used for the user interface. A 1024-Kbit EEPROM is big enough to handle the web pages including graphics. This of course doesn’t have room for fancy flash based configuration pages. But a decent JavaScript /CSS/HTML based UI should be okay too.

The **microcontroller** is a Microchip PIC24HJ128GP202 16-bit, 40-MIPS device (IC2). It has 128 KBytes Flash memory and 8 KBytes RAM. In terms of speed this microcontroller is more than capable of handling the web server and other TCP/IP client functionality.

The **S0 interface** around OK2A/B and OK1A requires a higher voltage (approx. 24 V) to operate. This is provided by a step-up DC-DC converter (IC6). The input voltage to this converter is taken directly from the power adapter supplying the gateway. The DC-DC converter is based on the widely used MC34063 chip used in a billion devices on the globe. Each S0 interface port

is isolated from the microcontroller with an optocoupler.

The **Ethernet interface** is based on the Microchip ENC28J60 Ethernet controller (IC5). The gateway uses an Ethernet connector with built in magnetics. It’s crucial to stick to the one shown here (X2). The ENC28J60 chip is very picky with the type of Ethernet connector used. Not all Ethernet connectors with built in magnetics can be used. The various connectors out there on the market are also not pin-compatible.

Construction and testing

The circuit board design for the Puls0 gateway appears in **Figure 2**. There is no special procedure for assembling the board. It is however a good idea to solder the different function blocks in steps, starting with the power supply. Then verify that the chips in each function block get the right voltage, and that there are no short circuits. Pay special attention to the voltage regulator and the heatsink. The pair have to be mounted carefully to prevent any risk of short circuits.

Once you’re sure the chips get the right voltage, do verify that the Ethernet controller has the right Vcap (pin 14) voltage. This voltage should be around 2.5 volts. If that is the case, plug in an Ethernet cable that’s connected to an active network through a switch. Now the LEDs in the connector should start to flash.

Assuming that there is no short circuit, the circuit is ready for programming.

Adding software to the Puls0 gateway

Without software the Puls0 gateway wouldn’t be of much use. The software is applied to the box in two steps. First, the microcontroller firmware is downloaded. Next the web pages are put into the EEPROM. All software components for the project and the Eagle board design data are available free of charge from the Elektor website [2].

The PIC24HJ128GP202 microcontroller can be programmed using a Microchip PICKIT3 or similar. Remember to keep the circuit

powered by an external power supply while programming.

In order for the S0 Gateway to operate properly the configuration webpages have to be present in the 1024-Kbit EEPROM. The lot can be downloaded with the built-in download functionality in the Powermeter gateway firmware. The built in download functionality is accessed using this URL: <http://192.168.X.X/mpfsupload>.

Using the Puls0 Powermeter Gateway

For the initial setup, connect the S0 output on the power meter to the S0-1 input on the Puls0 gateway box as shown in **Figure 3**. A light duty 2-wire cable can be used as only 24 VDC is carried. The S0 interface supports cable lengths of up to 20 meters (60 ft.).

Now connect the Puls0 gateway's Ethernet port with an Ethernet port on your router or switch. Then connect the 5-volt power supply to the power supply connector on the Puls0 gateway box. The green LED on the back of the Puls0 gateway box will start to flash. Every time a pulse from the power meter is detected the red LED flashes. If no pulses are detected, reverse the S0 cable wires (this could be due to a diode in the output path of the S0 interface).

Locating the web interface

The configuration of the Puls0 gateway is done using a web interface. In order to access the web interface, the IP address of the web-interface has to be known. Currently two ways exist to find the IP address: using a discovery application (Wizard) to find the IP address; using the DHCP IP client list in the Router that provides dynamic IP addresses.

Puls0 gateway configuration

Once the connection to the Puls0 gateway web-interface has been established, the connection settings for the openEnergyMonitor EmonCMS server have to be applied, see **Figure 4**.

You can find detailed instruction on how to set up the EmonCMS datalogging system on the [openenergymonitor](http://openenergymonitor.org) website.

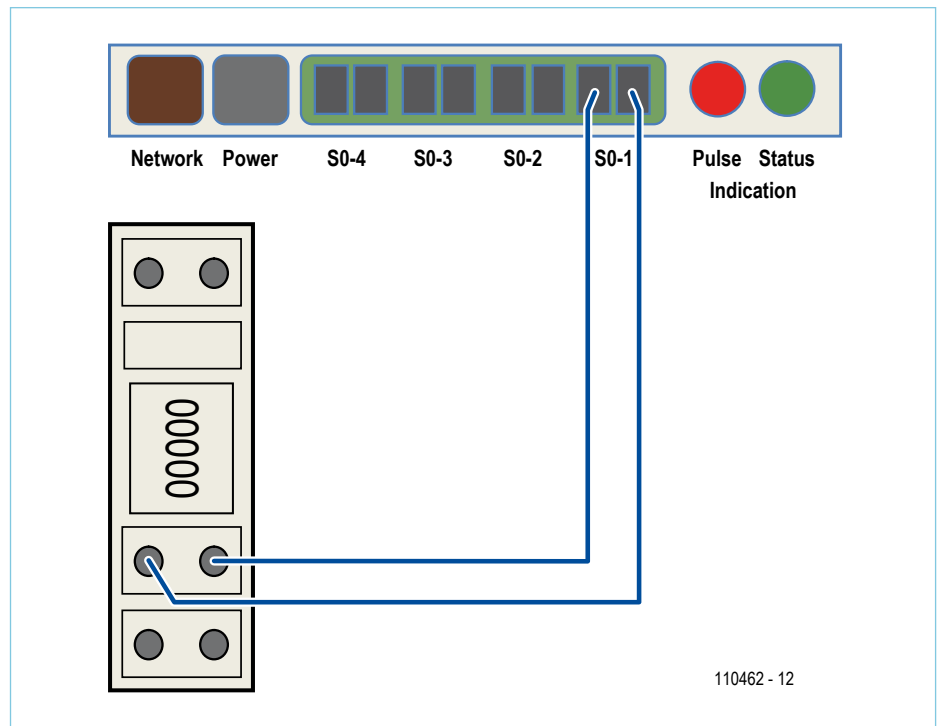


Figure 3. Connection of an S0-compatible meter unit to the gateway by way of a simple 2-wire cable.

Final thoughts

Although this project is targeted to the openEnergyMonitor platform, the firmware could be modified to handle other online data storage services. In terms of hardware however it is difficult to make radical changes without changing to the use of SMD components.

(110462)

Internet Links

www.openenergymonitor.org

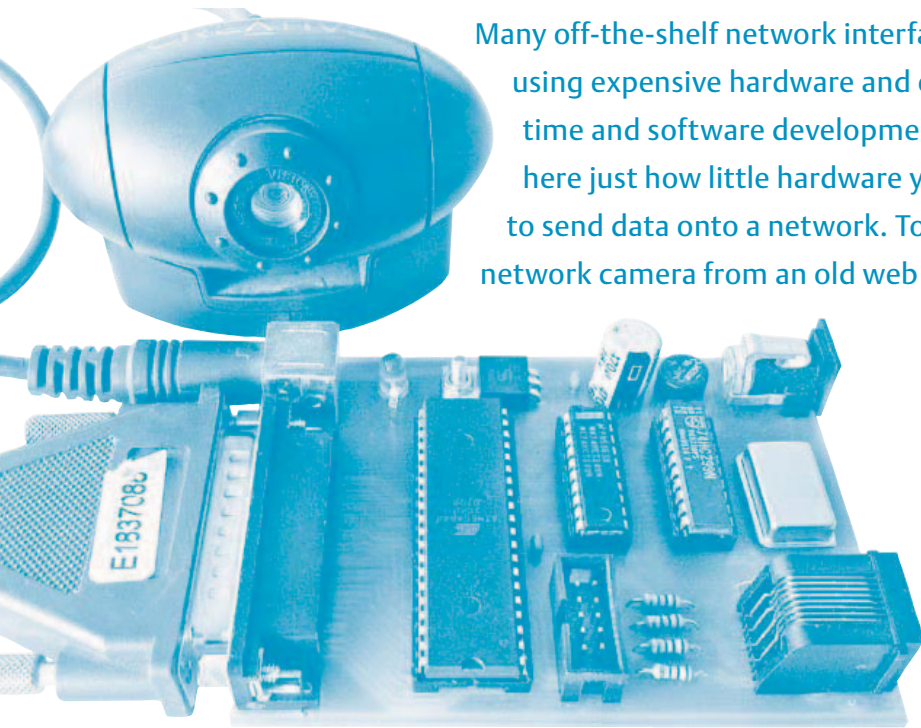
www.elektor.com/110462



Figure 4. OpenEnergyMonitor server configuration.

SDN – Software Defined NIC

Microcontroller Ethernet card



Many off-the-shelf network interface cards (NICs) have the disadvantage of using expensive hardware and often require substantial investment of time and software development before success is achieved. We show here just how little hardware you can get away with if you only need to send data onto a network. To demonstrate we build a dirt-cheap IP network camera from an old web cam.

ing transport layer protocols to build a host to host connection. The most well know example of a TCP protocol is the HTTP which takes care of communication between a web server and a web browser. Some applications use UDP despite the possibility of unreliable transfer. One of these is NFS (Net File System) from Sun microsystems, which allows access of files over a network on UDP.

Packing the packets

Each layer takes the data it is passed by the layer above and embeds it in its own frame. Contained in the frame is information to ensure that the packet arrives at the receiver processor. Some protocols place a header before the data, other append a trailer to the data. An overview showing how each data packet is constructed for transmission over an Ethernet cable is given in **Figure 2**. User data is provided with a header by the application (a web server for example) and forwarded to the TCP layer. This adds an additional header (TCP header) onto the complete packet, this header contains amongst other things a unique number. In addition the TCP header contains a so-called port number which is read during the receive processing to ensure the data arrives at the target computer. At the same time the header may also contain acknowledgement of the reception of a previous packet. The packet is now passed on to the IP layer. This layer also adds its own header to the packet which may for example contain the IP address of the sending and receiving computers. Following this it is passed to the network hardware where the packet is further expanded. In addition to the Ethernet header, which amongst other things contains the MAC addresses of the computers involved, the network card will calculate the checksum according to the CRC32 process and add it as a trailer to the

By Dr. Merten Joost (Germany)

Our series of articles on the SDR AVR featured in the last few editions of Elektor was a good illustration of just what can be achieved with a modest 8-bit AVR microcontroller. You may find it difficult to believe but when clocked at 20 MHz and with the help of very few peripheral components it is also possible to use just such a microcontroller to send Ethernet packets onto a network! To show how this 'Software Defined Network Interface Card (SDN)' can be built it will first be necessary to resort to a little bit of theory.

The TCP Model

Working with networks it is not long before you come across the standard OSI reference model for network communication. In this article however we use the TCP/IP model because it comes closer to reality.

From the diagram in **Figure 1** it can be seen that the model consists of four layers which build on one another. The lowest layer is the network access layer, here the network packets are sent onto the Ethernet cable. Next up is the network layer which ensures that the network packets find their way through the Internet. This network layer establishes a

communication link between two computers connected together via the Internet. On top of this is the transport layer which provides communication between processes on the computers. The best known of these protocols is the TCP (Transmission Control Protocol), which establishes point to point communication between two processes. In contrast to IP (Internet Protocol), TCP guarantees that the packets arrived intact and in the correct order. This is achieved by the receiver sending an acknowledgement onto the network to indicate correct packet reception or otherwise. There are also other protocols that are less well known at this level: The UDP (User Datagram Protocol) differs from the TCP protocol in that it is not secure, it doesn't guarantee arrival or in fact in which order the packets are received. This however gives the opportunity to broadcast data which the TCP does not allow. Broadcasting refers to the sending of a message to multiple recipients on the network. The uncertainty in the data transfer should never be less be put into perspective and packets can be resent if it is determined they were lost.

The highest layer of the TCP/IP model is the application layer, which uses the underlying

packet. Finally a preamble is added. While a software solution to the NIC reduces hardware it will be necessary for the software to construct the header, trailer and preamble for the packet.

The electrical layer

For simplicity the description will be limited to the 10-Mbit Ethernet standard. In its original form [1] coaxial cable was used as the transport medium carrying a differential signal with a voltage swing of ± 0.7 V. The twisted pair cable in use today carries a signal with a level of around ± 2.5 V. Before the digital data is sent over the cable it undergoes a process of Manchester coding according to IEEE 802.3. This process encodes a digital '1' as a rising edge and a '0' as a falling edge (see Figure 3). This coding method indirectly embeds a clock signal in the data signal guaranteeing a signal edge change for each bit of information. Sending a 10 Mbit/s data stream will therefore produce a transmission frequency of 20 MHz. Figure 3 also shows the preamble which precedes the data in each Ethernet packet. The preamble is eight bits long consisting of alternating ones and zeroes except for the last two bits which are both '1' to indicate end of preamble and start of data. The data is sent in 'little-endian' format so that the least significant bit goes first. After the body of data comes a CRC32 checksum consisting of four bytes.

The specification indicates that each Ethernet packet must be between 72 and 1,526 bytes long. Should there be only a small amount of data to send it will be necessary to fill the packet with data until it reaches the minimum length. It is important to be aware of this in the implementation. Connecting a 10 Mbit/s card to 100 Mbit/s capable equipment results in a fall back of the network communication speed to 10 Mbit/s. Network devices are backward compatible so that overall network speed is limited by the slowest device on the net. For identification purposes network devices will send out short pulses at regular intervals when the net is not busy indicating its transmission capability to a partner. At 10 Mbit/s this NLP (Normal Link Pulse) consists of a 100 ns pulse transmitted every 16 ms (± 8 ms). For our purposes this pulse will be generated in software.

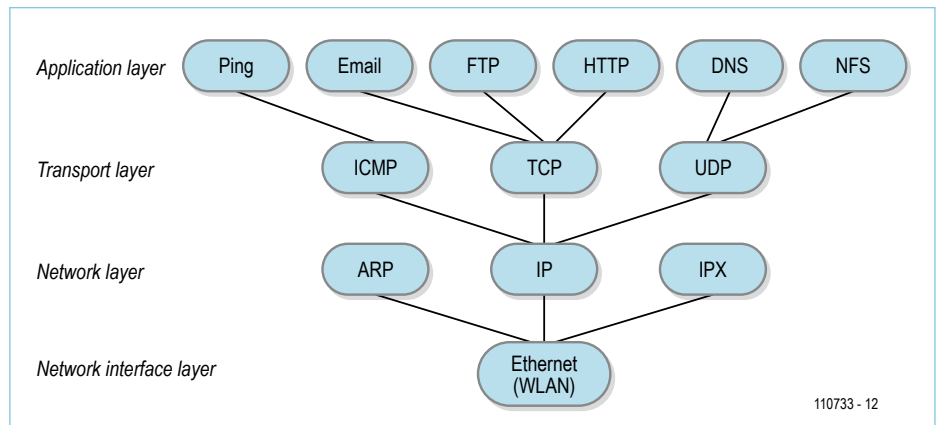


Figure 1. The TCP/IP layer model.

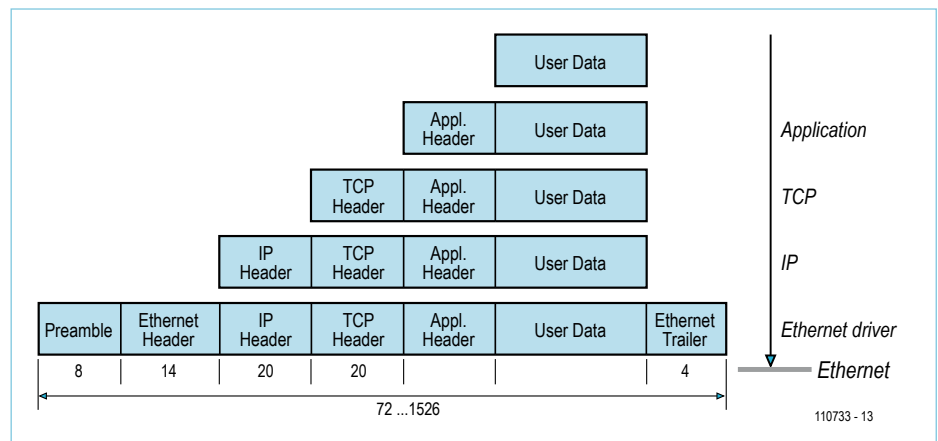


Figure 2. TCP data encapsulation.

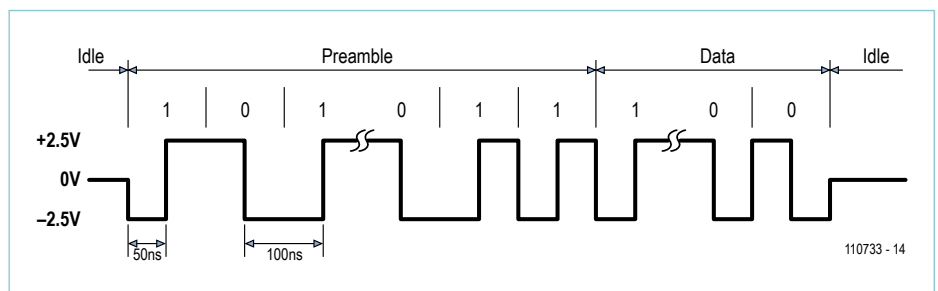


Figure 3. The Manchester coded preamble to the Ethernet packets.

AVR NIC

Comparing the data rate with the processor clock raises the question of how it is possible to send and receive a 20 MHz signal with a processor clocked at 20 MHz? The short answer is that it is not possible.

The simple process of receiving the data stream and storing it to memory requires a faster processor clock. Any attempt to interpret the data at the same time is just out of the question.

Without the possibility to receive data, communication according to TCP is also not possible. It specifies that received packets must be acknowledged, implying the need for a bidirectional capability. The UDP communication protocol however specifies no acknowledge. This introduces a degree of uncertainty in the transmission path because no handshake is implemented but for the same reason achieves a higher communication bandwidth. This allows transmission of streamed video and audio which

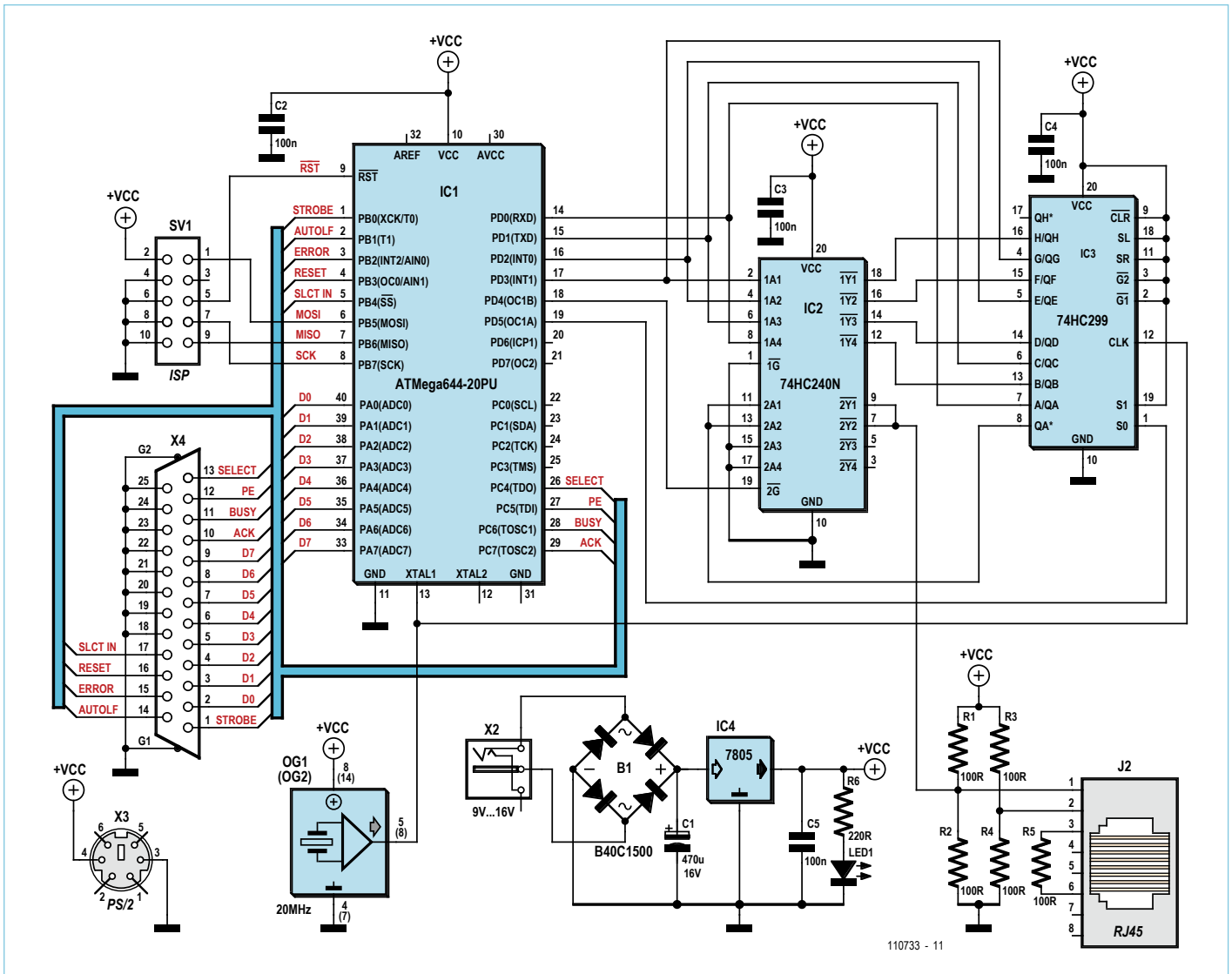


Figure 4. The circuit diagram.

can then be received by any computer connected to this local network.

Just sending the data clocked at 20 MHz is quite an achievement in itself; as well as reading the data from memory in real time it must be Manchester encoded, sent out at the correct time. In addition it is necessary to check the data length and append the CRC32 checksum. All we need from a hardware point of view is an inverting 8-bit buffer (74HC240) with tri-state outputs and an 8-bit shift register (74HC299) with parallel input and serial output – see the circuit in **Figure 4**. The microcontroller outputs data in the form of two nibbles. Using four inverters from IC2 we get eight bits in total, to provide inverted Manchester coding. These eight bits are loaded in parallel to shift register IC3 and clocked out in serial

at 20 MHz. The data stream passes through the inverting buffer where two cells are wired in parallel to provide additional drive current onto the Ethernet cable. This process inverts the signal again so that it now has the correct phase when it appears on the cable. The cable itself is terminated at either end by a voltage divider consisting of two 100 Ω resistors connected between Vcc and ground. This ensures that the quiescent voltage level of the cable is $V_{cc}/2$. A control signal switches the buffer outputs to high impedance when there is no data to be transmitted. When data is to be sent the buffer output is enabled producing a differential signal with a voltage swing of around ± 2 V. There is no requirement for a coupling transformer; the signal simply connects to the cable via an 8-way RJ45 modular connector.

Ethernet calculations

In order to send the packets of data in real time it will first be necessary to store the data together with the entire header and trailer information in an area of RAM so that the microcontroller has nothing to calculate when the packet is transmitted. This means that all the header variables and the checksum must be calculated and stored before transmission.

Figure 5 shows the construction and contents of the header used here. The Ethernet-header consists of three fields. The transmitters MAC address can be freely assigned so long as it is not the same as other devices on the local net. When just one target computer is known its MAC address can be used here. Alternatively the MAC broadcast address can be used (six bytes containing

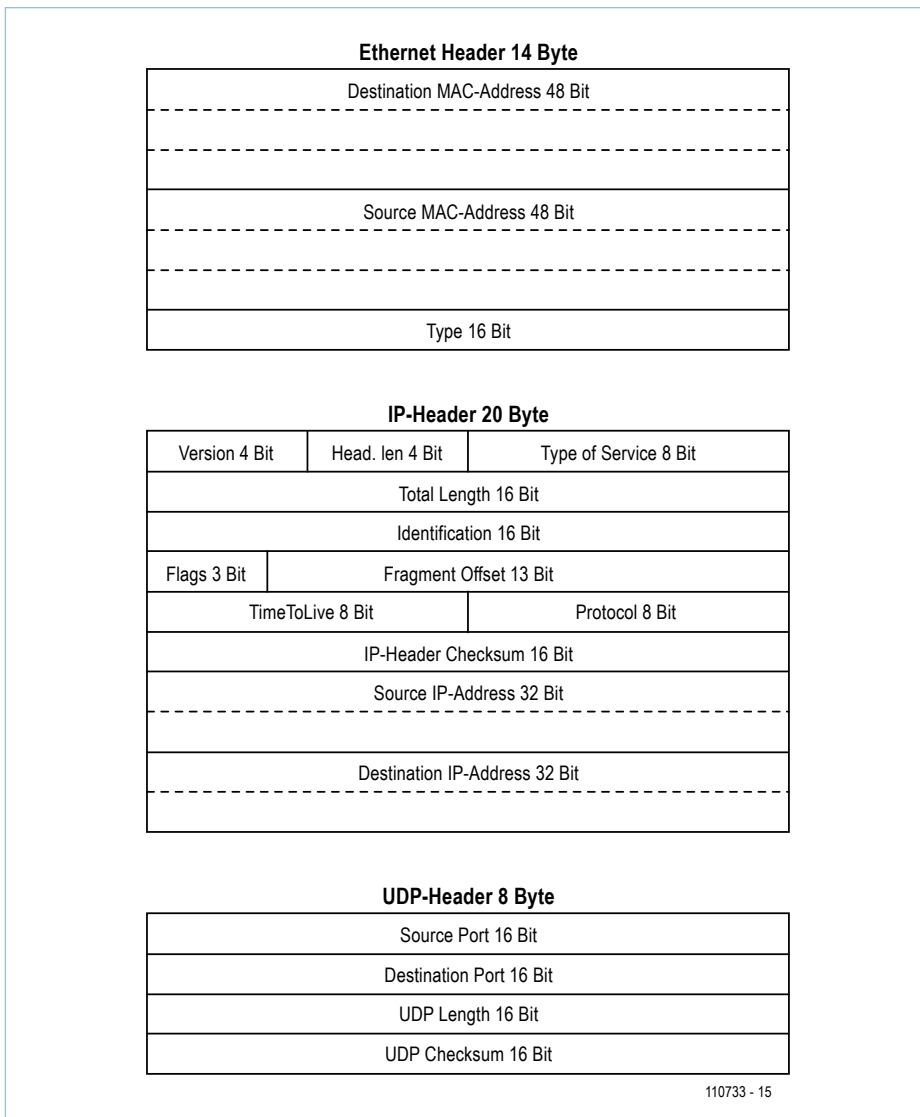


Figure 5. Ethernet header used by the SDN.

0xFF). Using this option means theoretically that any computer in the world could be the receiver. The last header field is the 'type' indicating which protocol is implemented in the frame data. In this case the constant value 0x0800 is used indicating it can be treated as an IP packet.

Some of the IP header fields can be assigned fixed constants. These include among other things the IP version number, the packet lifetime and UDP as the transport layer protocol. The sender and receiver IP address can be constants as they will be dependent on the values used by the local net. For broadcasting the last bits of the corresponding net mask are filled with '1's (the netmask divides the address into the actual network address and the subscriber in the sub-network). Using a network

address of 192.168.1.0 and a net mask of 255.255.255.0 gives the broadcast address 192.168.1.255. There are also fields which need to be calculated for each packet: The 'total length' field for example contains the length of the IP packets (header plus data). The packet length information can be different every time to allow the transmission of different amounts of data. In addition a checksum of the IP header is calculated and contained in the 'IP Header Checksum' field. An option allows each packet to be identified by a consecutive value written to the 'Identification' field.

The ports of the sending and receiving processes are registered in the UDP header. This will generally be a constant value and must correspond with the port of the receiving processes. Port numbers below 1,000 should be avoided because this

BEST SCOPE SELECTION & lowest prices!

WORLD'S SMALLEST

World's smallest MSO!
This DIP-sized 200kHz
2-ch scope includes a
spectrum analyzer and
Arbitrary Waveform Gen.
Measures only 1 x 1.6 inches in size!

XPROTOLAB \$49



IPHONE SCOPE

5MHz mixed signal
scope adapter for the
iPhone, iPad and iPod Touch!
The FREE iMSO-104 app is available for
download from the Apple App Store.

iMSO-104 \$297.99

30MHZ SCOPE

2-ch, 250MS/s
sample rate
30MHz scope
with 8" color
TFT-LCD, AutoScale
& waveform mathematic functions.
Quality FREE carry case included.

SDS5032E \$299



60MHZ SCOPE

60MHz 2-ch scope
with 500MSa/s rate
and huge 10MSa memory!

SDS6062 \$359

100MHZ SCOPE

High-end 100MHz 2-ch
1GSa/s benchscope
with 1MSa memory
and USB port • FREE
scope carry case. New super low price!

DS1102E \$399



100MHZ MSO

2-ch 100MSa/s
scope • 8-ch logic
analyzer. USB 2.0 and
4M samples storage per channel with
advanced triggering & math functions.

CS328A \$1359

20MHZ HANDHELD

Fast & accurate handheld
20MHz 1-ch oscilloscope.
- 100 M/S sample rate
- 3.5 in. color TFT-LCD
- 6 hour battery life
FREE rugged, impact-resistant case!

HDS1021M \$269.95

WWW.SAELIG.COM

Saelig
unique electronics





Figure 6. The author's finished prototype.

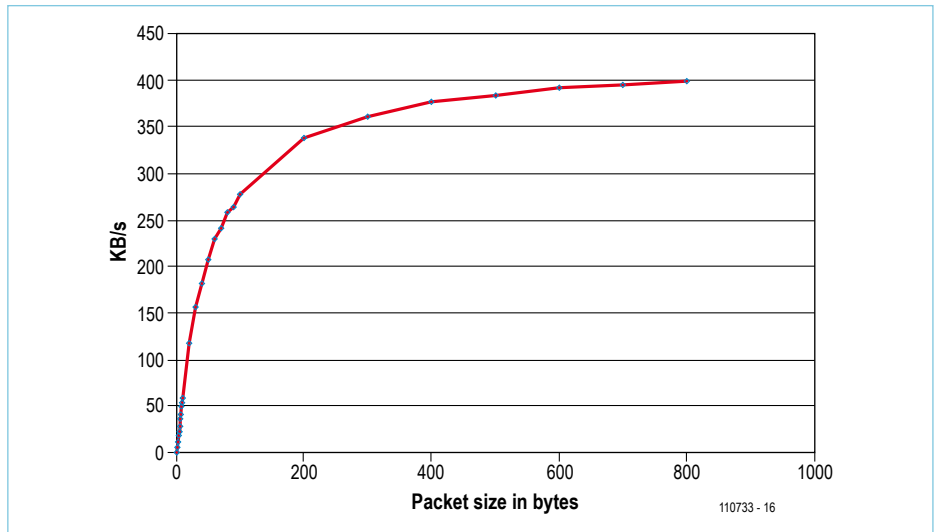


Figure 7. Data throughput is dependant on packet size.

range includes the so-called 'well know ports' where processes would require special privileges to bind a network socket to an IP address. The 'UDP length' field is similar to the 'IP length' field, its value is dependant on the packet size so must be calculated for each packet. The UDP checksum is fortunately optional and is not calculated in this version.

Next the total packet length is checked to make sure it complies with the minimum packet length restriction. When the packet length is found to be too short its size is padded out with additional 'padding bytes' to increase its length. The actual value of the padding bytes is arbitrary. Once this is completed the checksum on the complete data packet can be calculated. The Ethernet preamble is not included in the checksum calculation but the data and padding bytes will be. The resulting CRC32 checksum value is now added to the end of the packet. The source code for the CRC32 calculation routine has been borrowed from the Linux kernel; it was compiled, disassembled and

heavily optimized by hand to improve the routine's efficiency.

A library for the GNU-Assembler has been produced which contains routines to perform all the tasks outlined above and is C compatible. It can be used with the AVR-GCC (in AVR studio 5 for example) to program all the applications. The library has two functions that are useful here; `UDPInit()` expects a pointer value as a parameter which points to an area in memory where the complete Ethernet packet will be stored. The routine stores all the static header information starting at this memory location. It then goes on to initialize the 8-bit timer0 so that every 13 ms an interrupt routine is called which sends the NLP. The data is placed 50 Bytes after the start of the packet storage location. The second function `UDPSend()` calculates the header information from the data packets, corrects the packet length when necessary, calculates the CRC32 checksum and sends the packet to the net. This function expects two pointer values as parameters; the first

points to the beginning of the packets in memory while the second points to the end location of the actual data.

Up and running

The SDN needs just six LSBs from any free port of an AVR microcontroller. Four of these are used to transfer the data value while the other two control the shift register and the tri-state inverter bus driver. The controller is clocked from an external 20 MHz oscillator which also clocks the shift register. The sparse component count with just IC2 and IC3 means that the entire circuit can be laid out on a small single sided PCB. The PCB layout files along with the firmware source files for this project are available to download for free from the Elektor web page for this article [2]. **Figure 6** shows the prototype board developed by the author.

In the AVR Studio environment it is necessary to add the file 'udp.S' in the source files for the project. The listing 'udp_avr.c' is a simple test program which reserves a 54

IP Cam

For test purposes an old parallel port camera was connected to free pins of the microcontroller. The driver code 'cam.c' and header 'cam.h' for the camera was inspired by an old GPL DOS driver which was ported to the microcontroller. The microcontroller reads the camera data line by line and creates a UDP packet for each line, prefixing the corresponding line number. The PC

software 'canvas.exe' takes the UDP packets and displays the image line by line. It achieves a frame rate of approximately two frames per second which is often sufficient for surveillance applications. This is a good demonstration of the capability of the software defined NIC to make an IP capable camera with a hardware cost of under \$15.

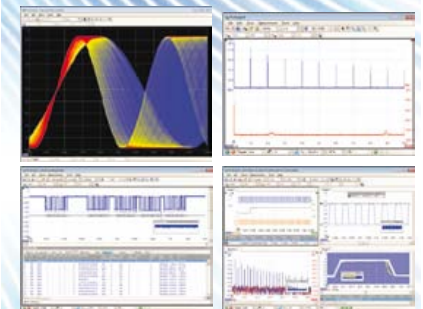
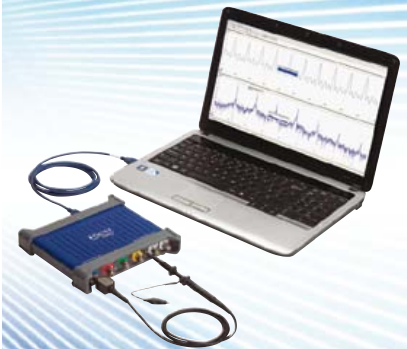
Those of you interested in this demonstration should note that the camera used is a 'Creative Labs Video Blaster WebCam II, Model 1100001424' for parallel port operation, which can be found for sale at on-line auction sites.

NEW

pico
Technology

PicoScope® 4-CHANNEL 3000 Series

200 MHz bandwidth
128 MS deep memory
1 GS/s real-time sampling
Advanced digital triggering
200 MHz spectrum analyzer
Function generator or AWG



ALL MODELS INCLUDE PROBES, FULL SOFTWARE AND 5 YEAR WARRANTY. SOFTWARE INCLUDES: MEASUREMENTS, SPECTRUM ANALYZER, FULL SDK, ADVANCED TRIGGERS, COLOR PERSISTENCE, SERIAL DECODING (CAN, LIN, RS232, I²C, FLEXRAY, SPI), MASKS, MATH CHANNELS, ALL AS STANDARD. FREE UPDATES.



FOR THE FULL PRODUCT RANGE VISIT

www.picotech.com/pco487
or Call: 1-800-591-2796

Listing „udp_pc.c“

```
#define BUFLLEN 800
#define PORT 6668

main()
{
    SOCKET s;
    struct sockaddr_in adresse;
    unsigned char buf[BUFLLEN];
    int num;

    if((s = socket(AF_INET, SOCK_DGRAM, 0)) < 0)
        printf(„no socket %d\n“, s), exit(0);
    adresse.sin_family = AF_INET;
    adresse.sin_port = htons(PORT);
    adresse.sin_addr.s_addr = htonl(INADDR_ANY);

    if(bind(s, (struct sockaddr*) &adresse,
    sizeof(adresse)) != 0)
        printf(„bind failed\n“), exit(0);

    num=recv(s, buf, BUFLLEN, 0);
    // do something
}
```

byte area of memory for the payload. The first 50 bytes are reserved for the header while the last four bytes contain the checksum. In order to send small packets without the need to reallocate memory area every time the send function provides a second parameter which points to end of the transmit data area. When the circuit is connected to a network and a (free) copy of the packet capture software Wireshark [3] is running on the PC, incoming data packets can be viewed in great detail.

PC Source

In order to be able to read data sent by the SDN it is necessary to run software on a PC which can receive the UDP formatted packets. The listing ‘udp_pc.c’ demonstrates in C (GCC under Cygwin) that you need very little code to be able to receive data via a UDP socket. Once the socket has been set up it is only necessary to call the function ‘recv()’. On the PC side you are not confined to the use of the C language, sockets are also used in some other high level languages such as Java or Delphi.

Data throughput

A 10 Mbit Ethernet achieves a data rate of 1 MB/s maximum. **Figure 7** shows the achievable data throughput in relation to the packet length. It can be seen that a data rate of ‘only’ 400 KB/s can be achieved. This is a result of the additional computa-

tional overheads needed to calculate the CRC32 checksum. Measurements indicate that the checksum calculation takes a little more time than is actually required to send the data.

(110733)

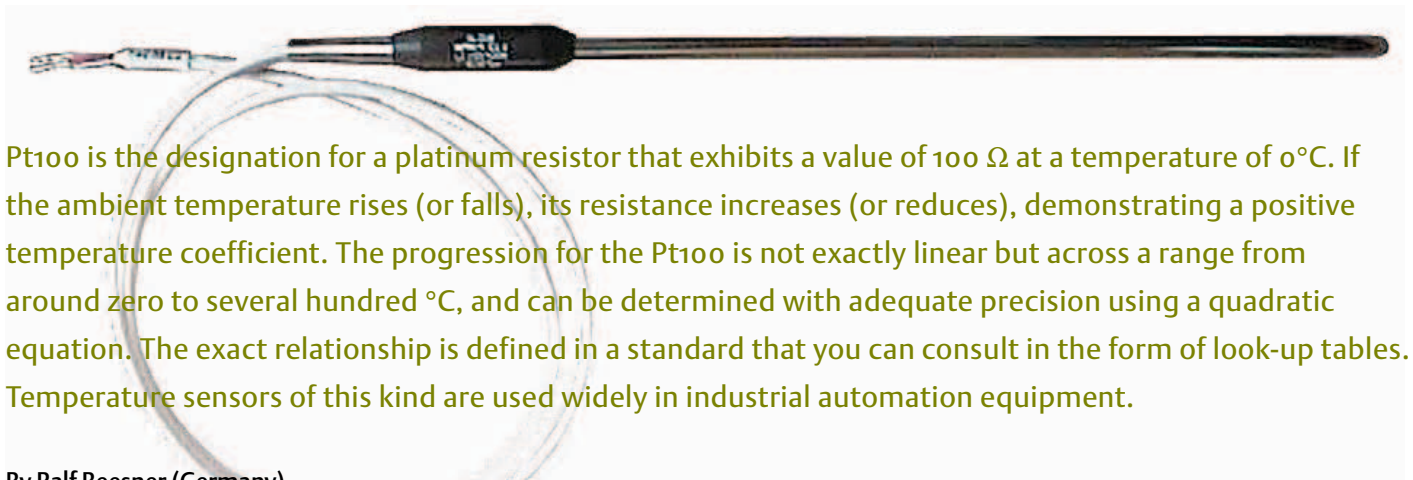
Internet Links

- [1] http://research.microsoft.com/en-us/um/people/gbell/Ethernet_Blue_Book_1980.pdf
- [2] www.elektor.com/110730
- [3] www.wireshark.org
- [4] <http://userpages.uni-koblenz.de/~physik/informatik/techC.pdf>

The Author

Dr. Merten Joost is a lecturer at the university of Koblenz, Germany, and lectures on the subjects of digital electronics, Microcontrollers and Robotics. The contents of this article are the basis of his lecture on microcontrollers [4], he also specializes on the basics of computer networks and efficient microcontroller programming.

Pt100 Simulator



Pt100 is the designation for a platinum resistor that exhibits a value of 100 Ω at a temperature of 0°C. If the ambient temperature rises (or falls), its resistance increases (or reduces), demonstrating a positive temperature coefficient. The progression for the Pt100 is not exactly linear but across a range from around zero to several hundred °C, and can be determined with adequate precision using a quadratic equation. The exact relationship is defined in a standard that you can consult in the form of look-up tables. Temperature sensors of this kind are used widely in industrial automation equipment.

By Ralf Beesner (Germany)

Anyone developing circuits that contain Pt100 elements has to reckon with elaborate test set-ups involving iced water baths and adjustable heat sources, which may not arouse great pleasure. But you can avoid this inconvenience with the circuit presented here, which simulates the behavior of a Pt100 resistor in the temperature range -25 °C to 350 °C.

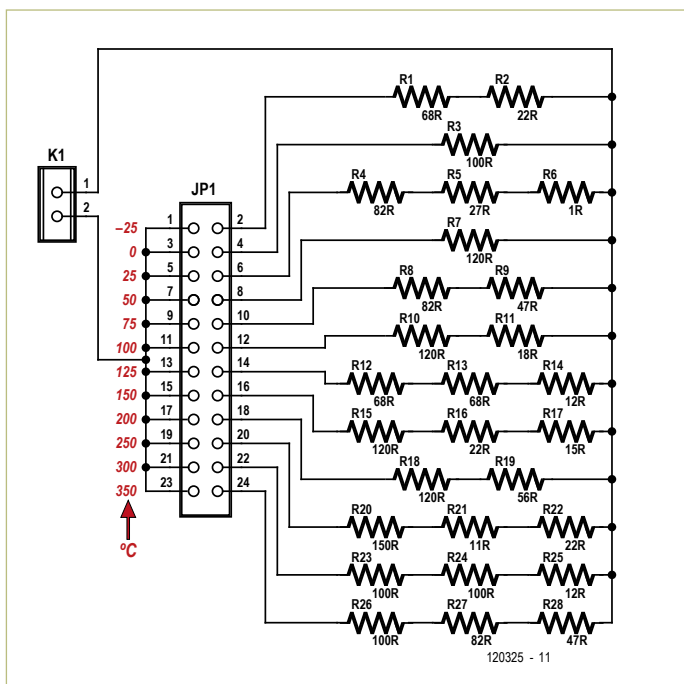
Constructing the circuit is relatively straightforward: terminal K1 connects the simulator to the circuit calling for the Pt100 element. The individual resistor paths represent the resistance value of the

Pt100 at a specific temperature (see values indicated on circuit diagram). A jumper field allows you to select the resistance combination for the temperature required.

Normal off-the-shelf resistors with 1 % accuracy are fine for building our Pt100 simulator. As the standard values of the E24 series do not correspond exactly to the resistor values of Pt100 we use combinations of resistors connected in series to provide an adequately close match. This table gives an overview of the series resistor combinations (R_{Sim}) and the corresponding standard values of the Pt100 standard (R_{Norm}) together with the deviation in each instance.

(120325)

Pt100 is the designation for a platinum resistor that exhibits a value of 100 Ω at a temperature of 0°C.



Resistor values for the Pt100 simulator						
T [°C]	RNorm [Ω]	RSim [Ω]	Individual resistors			Deviation [%]
			1	2	3	
-25	90.2	90	68	22	0	-0.22
0	100	100	100	0	0	0.00
25	109.7	110	82	27	1	0.27
50	119.4	120	120	0	0	0.50
75	129	129	82	47	0	0.00
100	138.5	138	120	18	0	-0.36
125	148	148	68	68	12	0.00
150	157.3	157	120	22	15	-0.19
200	175.8	176	120	56	0	0.11
250	194.1	194	150	22	22	-0.05
300	212	212	100	100	12	0.00
350	223.9	229	100	82	47	2.28

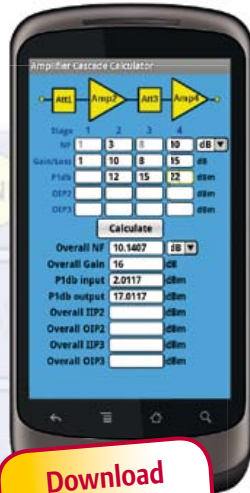
Elektor RF & Microwave Toolbox

➔ for Android

The RF & Microwave Toolbox contains 34 calculation and conversion tools for RF, microwave and electronics in general. Whether you are an RF professional, radio-amateur, astronomer or hobbyist, this app puts some of the most important tools right at your fingertips. Literally!

Highlights:

- Amplifier cascade (NF, Gain, P1db, OIP2, OIP3)
- Field intensity and power density converter (W/m², V/m, A/m, Tesla, Gauss, dBm, W)
- PCB Trace calculator (impedance/dimensions)
- PI and T attenuator
- Antenna temperature (Kelvin)
- EMC (EIRP, ERP, dBμV/m)
- Filter Design (Butterworth, Chebyshev, prototype)
- And much more



Download your app now!



Further information at

www.elektor.com/rf-app

Custom Front Panels & Enclosures



FREE Software



Sample price \$57.32 + S&H

Designed by you using our FREE software, Front Panel Designer

- Cost effective prototypes and production runs
- Powder-coated finish and panel thickness up to 10mm now available
- Choose from aluminum, acrylic or customer provided material
- 1, 3 and 5-day lead times available

FRONT PANEL EXPRESS

FrontPanelExpress.com
1(800)FPE-9060

Fascinated by technology's impact on the future?

Check out Tech the Future!

Computing power and global interconnectivity are pushing tech innovation into overdrive. Pioneering technologies and creative workarounds affect even the couch potato 24/7. Tech the Future reports on technology strides that shape the future — yours included.

www.techthefuture.com

Follow Tech the Future



Electronics for Starters (9)

Sine-wave oscillators

If we only used electronics to process existing signals, we would be missing an important aspect of electronics: generating oscillating signals, as if by magic. Oscillators are important parts of many devices and are used for a wide variety of purposes. For example, they can be used to generate audible signals or test signals for checking out circuits and modules.

By Burkhard Kainka (Germany)

RC oscillators

Everyone knows the unpleasant whistling that can occur with a public address system. It results from acoustic feedback between the loudspeaker and the microphone. The pitch of the tone varies from one situation to the next, and the effect can only be prevented by increasing the distance between the system components or reducing the gain.

In theory, any circuit or system with sufficient feedback can oscillate. The feedback path may be purely electronic, such as feedback from a signal output to an input. A necessary condition is the right phase relationship, which is present with a two-stage amplifier.

The circuit in **Figure 1** is similar to that of a multivibrator, but with adjustable feedback. A multivibrator always generates square-wave signals, but the circuit shown here can also generate sine waves or other wave-

forms. The feedback can be adjusted with the volume control to the point where weak oscillation just starts to occur. The waveform in this situation is usually sinusoidal.

It is also possible to generate an oscillating signal with a single transistor, even though it has a 180-degree phase shift. The required additional 180-degree phase shift can be achieved by connecting several RC networks in series. The phase-shift oscillator shown in **Figure 2** generates a sine-wave signal at approximately 800 kHz, which is ideal for purposes such as practicing your Morse code or providing a test signal for checking out audio amplifiers.

A working phase-shift oscillator can also be built using a BS170 field effect transistor. The circuit in **Figure 3** is designed with very high resistance values and oscillates at a frequency of 10 Hz. It draws a very low operating current of approximately 30 μA .

Ring oscillators

Up to now we have used one-stage or two-stage amplifiers to build oscillators. What

happens if you have a circuit with three common-emitter stages? You would actually expect the feedback to be negative, since the overall phase shift is 180 degrees. However, in practice the circuit oscillates (**Figure 4**). The oscillating frequency rises with increasing supply voltage and can rise as high as 1 MHz.

What is happening here? We basically have a three-stage amplifier with negative feedback and very high voltage gain. However, each of the stages also causes a small time delay in addition to its gain. At a very specific frequency, the combination of these three delays results in an additional 180-degree phase shift. The negative feedback therefore turns into positive feedback at this frequency, and the result is oscillation. If you want to use a circuit of this sort as an amplifier for very low input signal levels rather than an oscillator, you must do everything possible to prevent any form of positive feedback. With such high gain it is not especially easy to prevent parasitic oscillations.

It's easier to build a three-stage oscillator

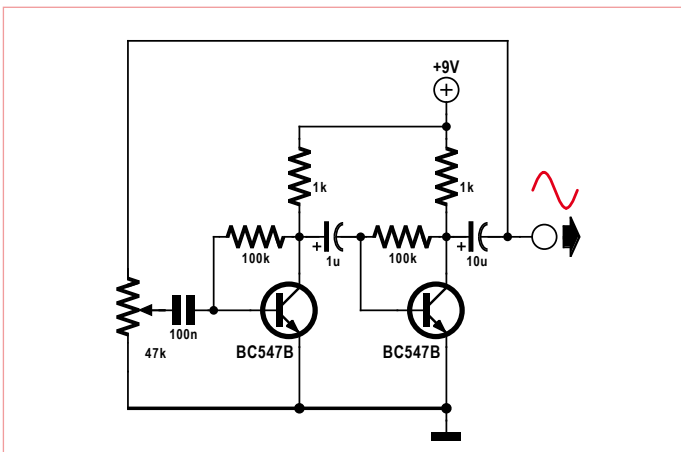


Figure 1. Oscillation caused by positive feedback.

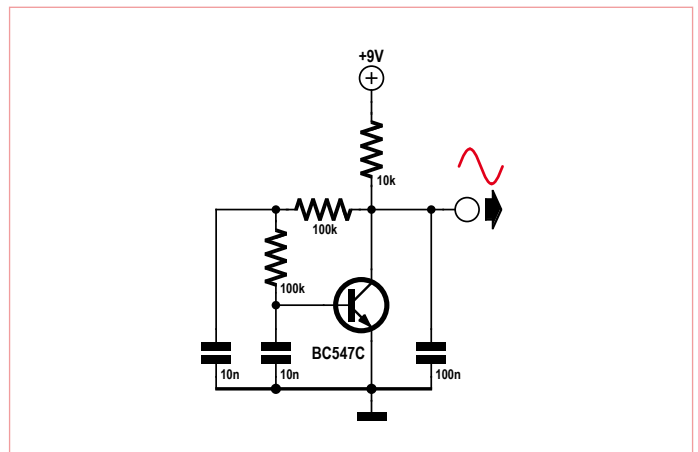


Figure 2. A phase-shift oscillator.

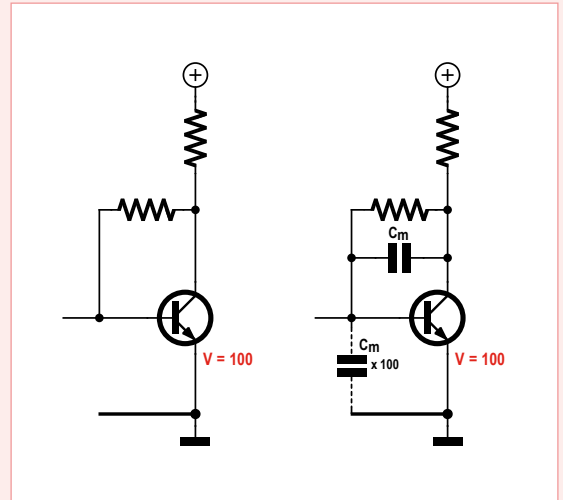
The Miller effect

The voltage gain of a common-emitter amplifier stage is typically around 100. This holds true up to fairly high frequencies, but sometimes not as high as you might wish. Although the unity gain frequency of the BC547 is approximately 300 MHz (the current gain drops to 1 at 300 MHz), the upper limit frequency of this amplifier circuit is much lower, especially if the circuit is designed with fairly high resistance values. The culprit here is the internal junction capacitances of the transistor.

The base–collector capacitance C_{bc} has an especially strong influence, even though it is only around 5 pF with a BC547. This is due to the Miller effect. The Miller capacitance C_m (i.e. C_{bc}) between the input and the output of the inverting amplifier is charged and discharged from two sides. For example, if the base voltage rises by 1 mV, the collector voltage simultaneously drops by 100 mV. This means that 100 times as much charge must be supplied. The net effect is that there appears to be a capacitor connected to the input with a value equal to the Miller capacitance multiplied by the voltage gain, which in this case would be around 500 pF. The combination of this capacitance and the internal resistance of the connected signal source forms a low-pass filter that drastically reduces the upper limit of the amplifier bandwidth.

For an amplifier this means that if wide bandwidth is important, you should keep the circuit resistances as low as possible. In addition, in some cases it can be worthwhile to work with lower voltage gain, for example by reducing the output impedance. Another good option is to use special HF transistors with much lower junction capacitance.

In the case of oscillators, the Miller capacitance allows us to build oscillators without using capacitors to determine the frequency, since the transistor itself provides the necessary capacitance.



than a three-stage amplifier. The lower the average collector current, the higher the impedance of the circuit – and the internal capacitances of the transistors have a stronger effect with increasing impedance. This is why the time delay is greater with a lower supply voltage, resulting in a lower oscillation frequency.

A circuit of this sort consists of a ring of individual amplifier stages, which is why it is called a ring oscillator. The same effect can also be achieved with five, seven or

nine stages. The only condition that has to be satisfied is that there is negative DC feedback. By contrast, with an even number of stages the result will always be a static flip-flop.

A three-stage ring oscillator can be operated with very high resistance values and therefore very low power consumption. With three 1-M Ω collector resistors, the oscillator operates with a supply voltage as low as 0.5 V and consumes less than 1 μ A. This means that a BPW34 photodiode in the sun, acting as a miniature solar cell, can

provide enough power to operate the oscillator (Figure 5). The frequency of the output signal is approximately 5 kHz. The frequency rises with increasing light level, so you might be able to put the circuit to good use as a light sensor.

You may be wondering how this circuit can oscillate at just 5 kHz, entirely without capacitors. This seems strange, considering that the internal capacitance of a transistor is only a few picofarads. The answer to this puzzle is what is called the Miller effect (see

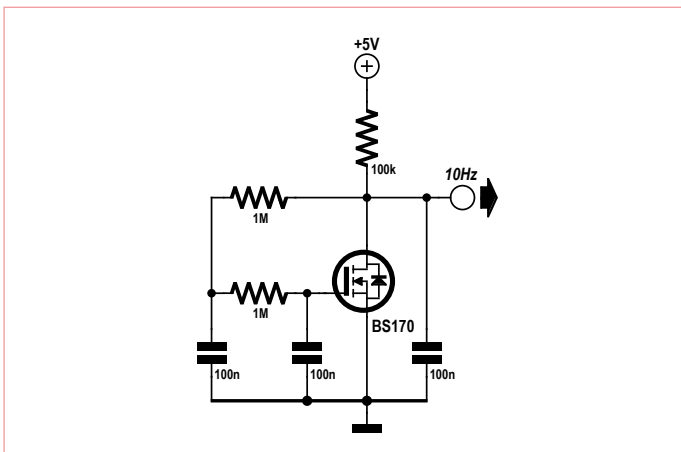


Figure 3. A phase-shift oscillator with a FET.

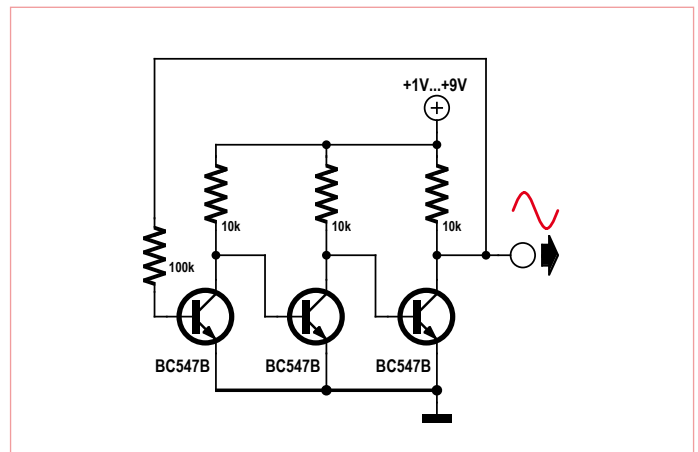
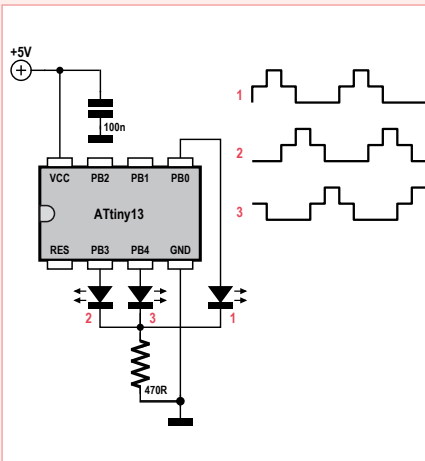


Figure 4. An oscillator with no capacitors.

Three-phase blinker



Three-phase signals can also be generated very easily with a microcontroller. This requires a total of six switching points with the same time spacing. The result with three LEDs is similar to that with a three-phase

ring oscillator, but distinctly more digital. A special feature of this circuit is that the three LEDs share a single series resistor. As a result, each LED operates at two brightness levels. There are three On states for each individual LED. In the first state it lights up together with the LED to its left and shares the operating current with the other LED. In the third state it does the same, but with the LED to its right. By contrast, in the middle state it lights up by itself and therefore receives the full current. Each LED accordingly passes through the following sequence of states: half on, full on, half on, off, off, off, and in each case with a phase difference of 120 degrees relative to its neighbour LED.

If you are new to AVR Basic, get a copy of *Elementary Course BASCOM-AVR*, see the online book store at www.elektor.com

```
'Three-phase blinker 1500 mA, 0.67 Hz
$regfile = "attiny13.dat"
$crystal = 1200000
Config Portb = Output

Do
  Portb.0 = 1
  Waitms 250
  Portb.3 = 0
  Waitms 250
  Portb.4 = 1
  Waitms 250
  Portb.0 = 0
  Waitms 250
  Portb.3 = 1
  Waitms 250
  Portb.4 = 0
  Waitms 250
Loop
End
```

inset), which causes the capacitance seen at the input to be the product of the collector–base capacitance and the voltage gain. Once you know this, you can easily connect additional capacitors between the collector and base leads to generate very low frequencies (Figure 6). With three 100-nF

capacitors, the output frequency is approximately 1 Hz.

Three-phase LED blinker

Attractive lighting effects can be generated with such low frequencies. The aim of the circuit shown in Figure 7 is to use three

LEDs to generate a pleasant flickering effect. This is a three-phase oscillator in which each of the three LEDs lights up in a different phase. The LED current is approximately sinusoidal, resulting in gentle transitions. Depending on whether you connect the circuit directly to the 9 V supply voltage or use

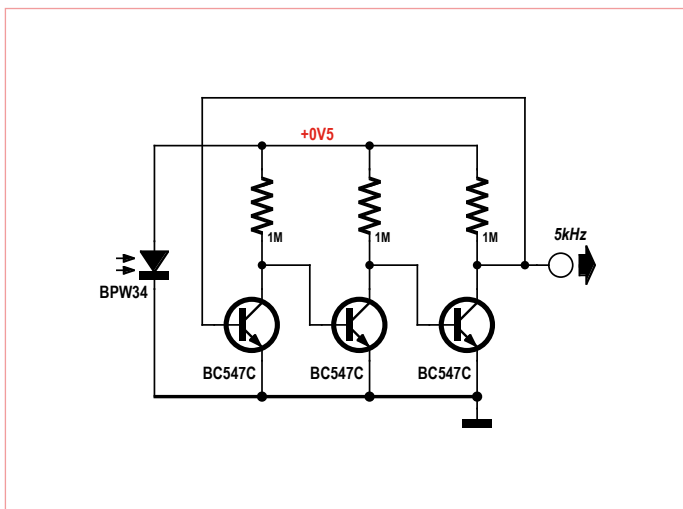


Figure 5. A ring oscillator powered by a solar cell.

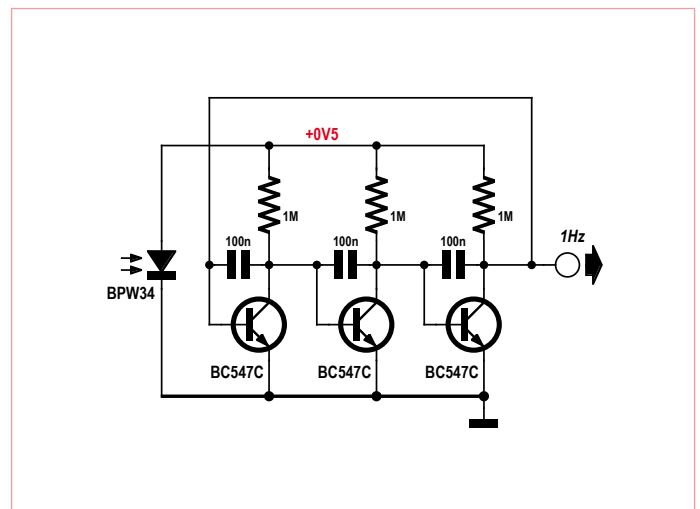
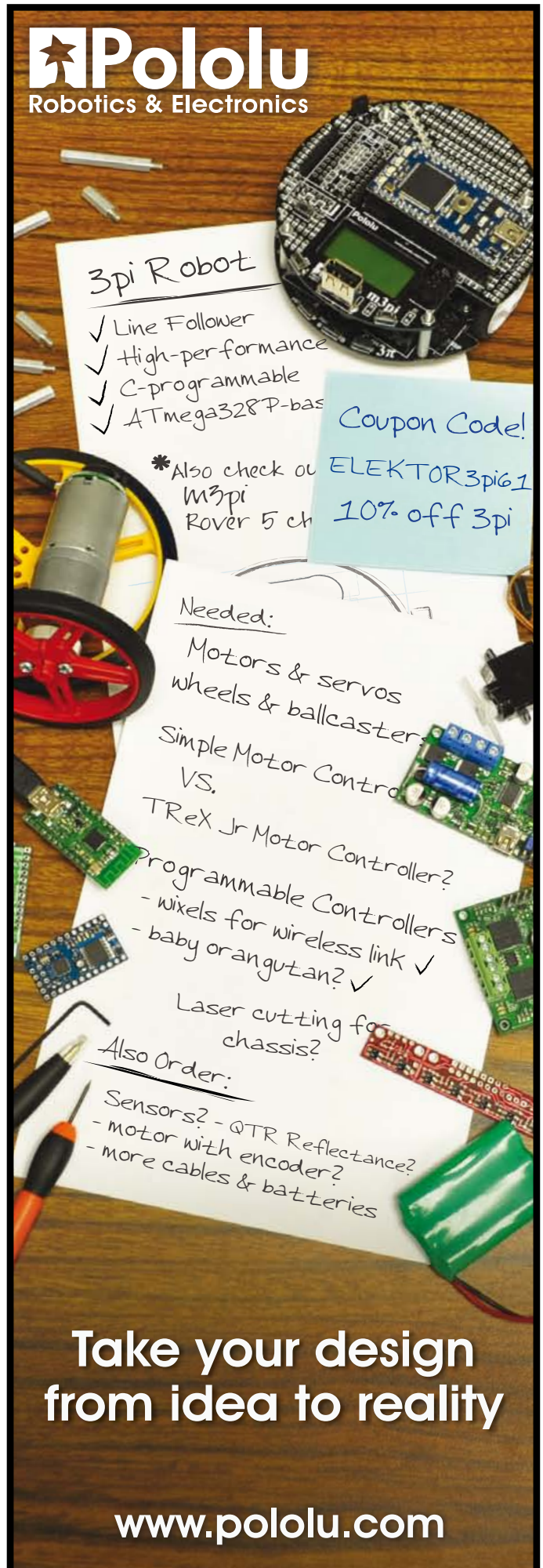


Figure 6. A lower-frequency ring oscillator with less power consumption.



Solution to the quiz in the September 2012 edition

The correct solution code is 'CFG'. Here are the explanations:

Answer 1:

The duration of a full period is $T = 2 \times 0.65 \text{ ms} = 1.3 \text{ ms}$, so $f = 1/T = 1/1.3 \text{ ms} = 0.769 \text{ kHz}$, or approximately 770 Hz (answer C is correct).

Answer 2:

The base voltage does not drop as much and returns to +0.6 V more quickly, so the frequency rises (answer F).

Answer 3:

The emitter current should be greater, so the emitter resistance needs to be reduced (answer G). This also causes the current through the loudspeaker to rise.

the potentiometer to reduce the operating current, the light is bright and flickers quickly or is less bright and flickers more slowly. Here again the frequency is highly dependent on the operating current.

(120009)

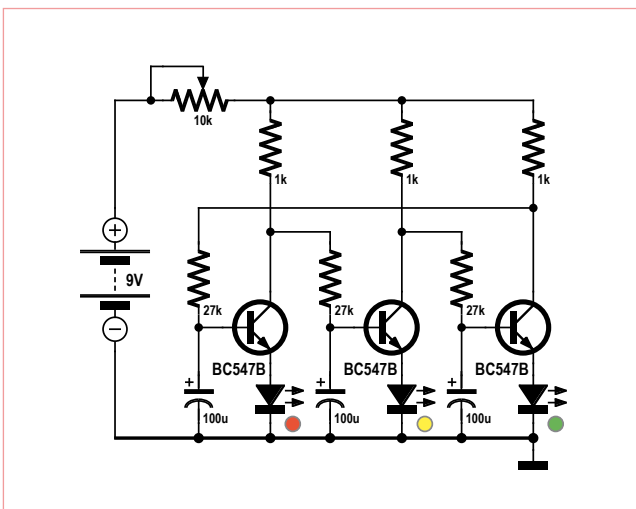


Figure 7. A three-phase lighting effect generator.

Take your design from idea to reality

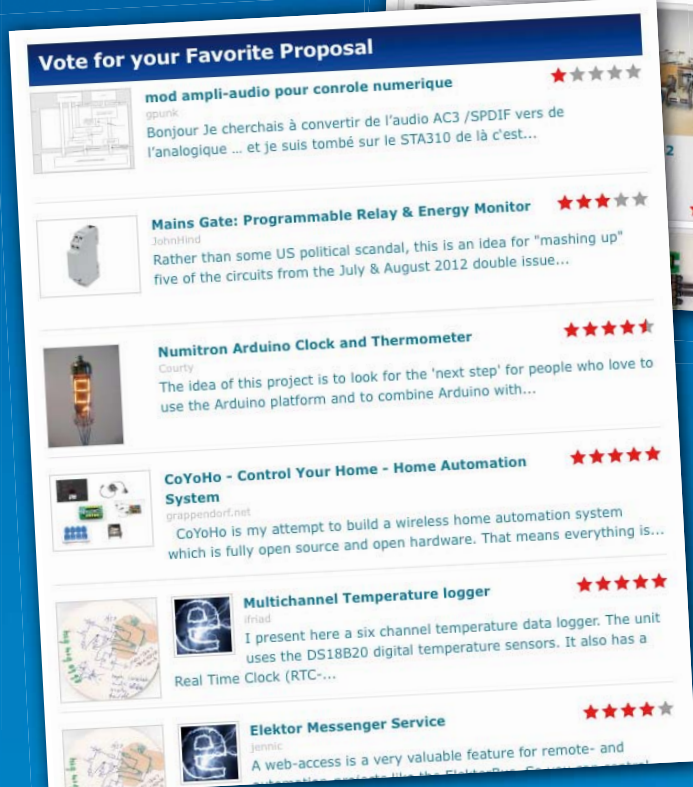
www.pololu.com

Get Elektorized

Sharing Electronics Projects

Elektor Projects is an online community for people passionate about electronics. Here you can share your projects and participate in those created by others. It's a place where you can discuss project development and electronics.

Elektor's team of editors and engineers assist you to bring your projects to a good end. They can help you write an article to be published in Elektor magazine or even develop a complete product that you can sell in the Elektor Shop!





See your project
in print!

Elektor magazine is
looking for
Technical Authors/
Design Engineers

If you have

- ✓ an innovative or original project you'd like to share with Elektor's 140 k+ readership and the electronics community
- ✓ above average skills in designing electronic circuits
- ✓ experience in writing electronics-related software
- ✓ basic skills in complementing your hardware or software with explanatory text
- ✓ a PC, email and Internet access for efficient communications with Elektor's centrally located team of editors and technicians

then don't hesitate to contact us for exciting opportunities to get your project or feature article published. Our Author Guidelines are at: www.elektor.com/authors.

Elektor USA
Jan Buiting MA, Editor
4 Park Street, Vernon, CT 06066, USA
Email: editor@elektor.com

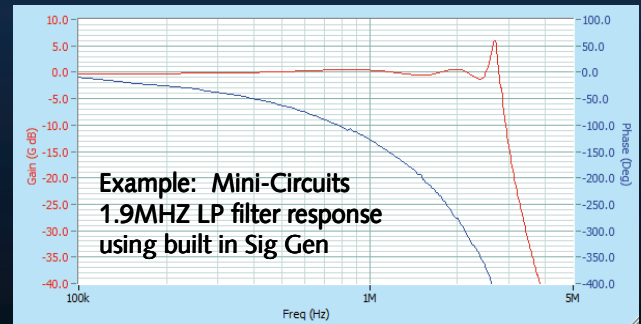
CS328A-XS



100MHz Mixed
Signal Oscilloscope
+ Signal Generator

Bode Plots / Filter Response

Bode plots are very useful to determine filter responses and check stability. Most other scopes don't do them. We do.



14 Bits 100 MSPS MSO



www.cleverscope.com



AP CIRCUITS

PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



HP-35: A Pocket Revolution (1972)

By Dipl.-Inf. Karl-Ludwig Butte (Germany)



It was 1968. With the HP-9100A Hewlett-Packard had broken new ground by broadening their product range to include electronic calculators. However, at a price of \$4900 (in today's money about \$33,000! [1]) the electronic calculator remained a dream for many engineers, for whom the slide rule was the most important tool. Everything changed four years later. With the help of low power integrated circuit technology developed by Mostek [2] Hewlett-Packard introduced the HP-35, the world's first pocket scientific calculator, on February 1, 1972. It had taken only eighteen months

Wild seventies — what were YOU doing?

The HP-35 was launched at an exciting time. The beginning of 1972 was marked by NASA's Apollo 16 and 17 manned space flights to the moon; Mariner 9 was beaming back pictures of Mars; and Pioneer 10 was just starting to snoop around Jupiter and the asteroid belt. In September the Öland road bridge in Sweden, one of the longest in Europe, was opened [3]. And of course technological advances in large scale integrated circuits at the beginning of the 1970s were a significant factor in the development of the electronic calculator [4].

for Tom Osborne and his design team to rise to Bill Hewlett's challenge and produce a scientific calculator that could be carried in a shirt pocket. Although initially only engineers in large companies could justify splashing out \$395 (about \$2600 in today's money) on an HP-35, the supremacy of the electronic calculator and the demise of the slide rule were now inevitable.

The hardware

Figure 1 shows the unit, which measures 147 mm by 81 mm by 33 mm (5.8 inches by 3.2 inches by 1.3 inches) and weighs 246.6 g (8.7 ounces). Below the single-line fifteen-digit display are the on-off switch and a keyboard of 35 keys. At Bill Hewlett's personal suggestion, the number of keys gave the calculator its name. The seven-segment display includes a tiny lens built into each digit, making the display easier to read while not increasing its physical size beyond the point where fifteen digits would no longer fit in the available space. Now LED displays are not exactly known for frugality of current consumption, and so the HP-35 came with a rechargeable battery pack made up from three standard AA-size cells in their own plastic housing (**Figure 2**). The positive and negative terminal contact areas are asymmetrically placed on the side of the housing to prevent reverse polarity connection. The calculator came with

Hewlett-Packard's HP-35 was the first scientific pocket calculator in the world. Suddenly tedious and inaccurate computations using log tables and slide rules became a thing of the past. The HP-35 offered square roots and trigonometric and logarithmic functions, all at the press of a button, and to an astonishing ten digits of precision.

The slide rule is dead: long live the HP-35!

'We should have one in a tenth the volume, ten times as fast, and at a tenth the price'. These words, or words much like them, were Bill Hewlett's reaction to an internal demonstration of the HP-9100A desktop calculator, a behemoth of a device weighing some 18 kg (40 lbs). The words were intended by way of congratulations to the development team!

Forty years ago this year Hewlett-Packard introduced the first scientific pocket calculator

a carrying case, a mains adaptor and a manual, all delivered in a rather imposing box. Sadly the last two items are missing from my collection (**Figure 3**).

Figure 4 shows the internals of the HP-35 (main circuit board on the right, keyboard and display board on the left), and **Figure 5** the block diagram. There is a total of five MOS LSI ICs on the main board, which measures 70 mm by 75 mm (2.8 inches by 2.9 inches), of which three are ROM chips that contain the device's software. The other two ICs contain the arithmetic and register unit, and the control and timing circuitry. The other board carries the keyboard and the display with its driver electronics.

The arithmetic and register unit is organized in five blocks: instruction storage and decoding, a timer circuit, seven 56-bit registers, and adder/subtractor, and the display decoder [5].

The control and timing circuit is responsible for coordinating the internal activities in the device. This includes scanning the keyboard, overall system synchronization, and generating instruction addresses [5].

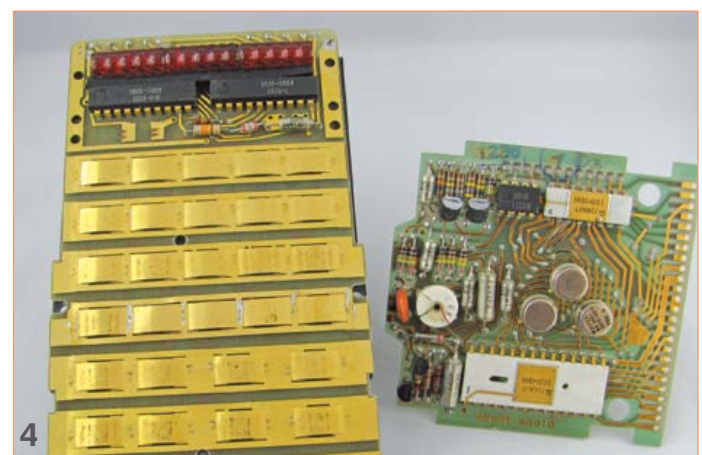
The three ROMs contain pre-programmed routines for calculating mathematical functions. Each ROM carries 256 instructions, each ten bits wide. There is also a decoder in each ROM. The total firmware storage available to implement all the calculator's functions is therefore just 7.5 kbits. By way of comparison, this brief article occupies about 38 kbytes, some forty times more. And that's not counting the illustrations! How did Tom Osborne and his team squeeze so much program logic into so small a space? We shall see below.

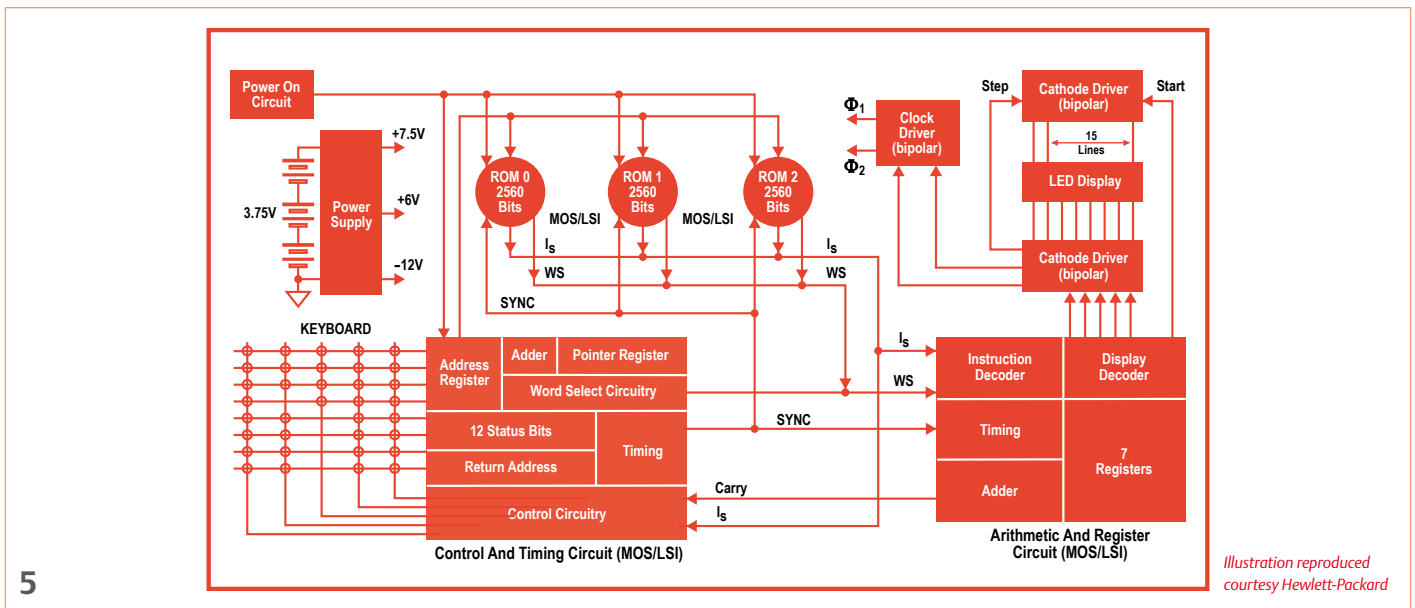
The software

There were two key elements to making the software fit in the tiny memory space available. The first was the use of 'reverse Polish notation' (RPN) for entering calculations and the second was the use of CORDIC algorithms.

Reverse Polish notation was a product of the work of Polish mathematician Jan Łukasiewicz, who developed a compact and bracket-free way of writing expressions in propositional logic called 'Polish notation' in the 1920s [6]. In Polish notation the operator is written first, followed by its operands. In the HP-35 it is the other way around: first the operands are pushed onto the calculator's four-level stack using the (conventionally double-size) 'Enter' key, and then the desired operator key is pressed. You will look in vain for the usual equals and brackets keys on the HP-35's keyboard: they are not needed in reverse Polish notation. In the 1970s, and in particular when the HP-65 was launched as the first programmable pocket calculator in the world, the debate over the relative merits or otherwise of RPN and Texas Instruments' competing 'algebraic operating system' (AOS) erupted into a quasi-religious war, much like subsequent scraps over the BASIC programming language versus Pascal and, later, C. There are many introductions to the elegant RPN entry method on the Internet, for example at [7].

Reverse Polish notation leads to economies in two ways: not only can the calculator dispense with the equals and brackets keys, but





5

Illustration reproduced courtesy Hewlett-Packard

also it simplifies the software required to implement expression evaluation. The user is also no longer constrained by a limit on the nesting of brackets, although it is necessary to keep in mind the finite depth of the stack.

The second key aspect of the HP-35's firmware design was the use of CORDIC algorithms. CORDIC is an abbreviation for 'COordinate Rotation Digital Computer' and it is a class of efficient iterative algorithms for calculating trigonometric and other mathematical functions developed by Jack E. Volder in 1958 [8]. Volder succeeded in designing a digital replacement for the unreliable navigation computer in the Convair B-58 bomber that could compute its position in real time. The 'real time' aspect can be seen as particularly significant when one considers that the B-58 was a supersonic aircraft. Tom Osborne had already made successful use of CORDIC algorithms in the HP9100A desktop calculator and so already knew how to implement multiplication and division, as well as trigonometric functions, using just addition and shift operations. The accuracy of the results bears comparison with that of any modern calculator, although there are sometimes small errors in the last digit. For example, the HP-35 gives $4^3 = 63.99999997$ rather than exactly 64.

Bugs

Despite the care that went into the development of the software a couple of bugs did remain, first discovered after more than 25,000 units had been sold [9]. For example, $e^{\ln 2.02}$ gave 2 rather than 2.02, and $\tan^{-1} 0.0002$ gave $5.729577893 \times 10^{-3}$ instead of 0.01145916 [10]. It was a difficult situation, and, these being the days before flash memory and the Internet, Hewlett-Packard did not have the option of publishing a software update for users to download as is now the norm for high-end graphical calculators. A crisis meeting was held, and Dave Packard got straight to the point: what were they going to do about all the units they had already sold? The now-famous suggestion of one colleague "Don't tell?" was of course not accepted [9]. Instead all customers were sent a notice informing them of the bug and offering a free repair (see [11] page 6, figures 5 and 6). It transpired that only about a quarter of the affected machines were sent back for repair: most customers kept their machines, often along with the notification letter describing the bug; many even ordered a new HP-35 with the corrected software.

The inexorable march of the pocket calculator

The HP-35 was one of Hewlett-Packard's most successful products ever, and on April 14, 2009 HP received the coveted 'IEEE Milestone in Electrical Engineering and Computing' award for the device [12]. One year after the HP-35 was launched its successor, the HP-45, appeared on the market and in 1974 HP introduced the HP-65, the first programmable pocket calculator in the world, which could store programs on tiny magnetic cards. In total HP has developed over a hundred different calculator models, and their machines have been used by many Nobel Prize winners. Although the hype surrounding electronic calculators has waned with the widespread use of personal computers, development continues with new, more sophisticated units offering graphics facilities and symbolic manipulation. Beyond the world of science, users in finance have also benefited from calculators specially designed for their needs: for example, in 1973 HP launched its HP-80 'business' calculator. The HP-12C financial calculator set another record for Hewlett-Packard, the device having been in continuous production since its introduction in 1981: over 30 years! (120454)

Sources and references

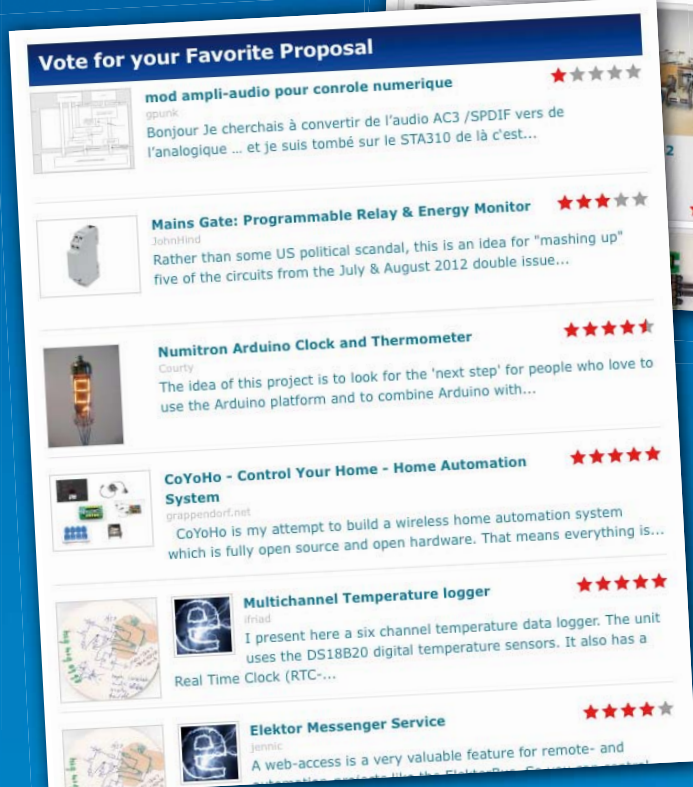
- [1] 'What is a dollar worth?' www.minneapolisfed.org/
- [2] Malone, Michael S.: 'Bill & Dave', Portfolio (Penguin Group) New York, 2007
- [3] <http://en.wikipedia.org/wiki/1972>
- [4] http://en.wikipedia.org/wiki/Integrated_circuit
- [5] HP Journal June 1971, page 2 onwards: www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1972-06.pdf
- [6] http://en.wikipedia.org/wiki/Polish_notation
- [7] <http://h41111.www4.hp.com/calculators/uk/en/articles/rpn.html>
- [8] <http://en.wikipedia.org/wiki/CORDIC>
- [9] <http://www.hpmuseum.org/hp35.htm>
- [10] <http://www.jacques-laporte.org/HP35%20ROM.htm>
- [11] <http://hhuc.us/2007/Remembering%20The%20HP35A.pdf>
- [12] http://www.ieee.org/about/news/2009/14april_1.html

Get Elektorized

Sharing Electronics Projects

Elektor Projects is an online community for people passionate about electronics. Here you can share your projects and participate in those created by others. It's a place where you can discuss project development and electronics.

Elektor's team of editors and engineers assist you to bring your projects to a good end. They can help you write an article to be published in Elektor magazine or even develop a complete product that you can sell in the Elektor Shop!



TANSTAAFL: NOT!

By Gerard Fonte (USA)

There Ain't No Such Thing As A Free Lunch (TANSTAAFL) — From 'The Moon is a Harsh Mistress' by Robert Heinlein. This is the idea that anything 'free' is not really free at all. The business integrates this overhead expense into its selling price of other products. So, if they didn't give away free things, the prices would be just a bit lower on everything else. However, this is not the case today. There are many free things. Some of which can be very valuable.

Make Believe

There is a completely free SPICE analog simulator software package. It's from Linear Technology (www.linear.com) and they call it LTspice IV. This is a professional simulator with lots of nice features. It includes a large 'spice-model' library of commonly used generic parts. And, of course, it includes an extensive library of most (if not all) Linear Technology parts. However, you can import third-party spice models, too. So if you want to simulate a Texas Instrument op-amp, you can use the Texas Instrument spice model in LTspice IV. They even give you instructions on how to do this (There are probably additional places on the web where you can get a Spice simulator for free).

It's my understanding that this didn't start out this way. It started as a simulator for their line of 'Simple Switcher' switching power regulators. It allowed you to simulate their (and only their) power regulator chips. True, it was a 'free' Spice-based simulator. But they didn't include non-power products in their library. And it wasn't as easy to import third-party models. As such, this was an example of TANSTAAFL. The simulator only simulated their parts (easily) so you paid a bit more when you purchased their Simple Switcher or other components.

But that's not the case now. It's a completely free download and is not at all limited to their parts. It wasn't that long ago that this type of graphical analog simulation software cost hundreds or even thousands of dollars. There is absolutely no reason why every reader shouldn't download and learn to use it. This is an extremely valuable and useful tool (It can even be used as a simple schematic drawing package).

To the Drawing Board

Recently I wanted to change my schematic capture and Printed Circuit Board (PCB) layout software. And since I'm such a cheap person, I decided to see what I could get for free. There are a number of PCB manufacturers that provide 'free' schematic capture and PCB layout tools. But, the PCB files are special and can only be read by the PCB manufacturer. So if you want to actually fabricate the PCBs you have to go to that manufacturer. This is a classic case of TANSTAAFL (However, their schematic capture programs are still very useful and can be used independently).

I came across TinyCad (www.tinycad.sourceforge.net). This is a remarkably complete and professional schematic capture package. And it's absolutely free. It includes a library of commonly used parts. But since it's impossible to keep up with all the new parts being offered by manufacturers, it's important for any schematic

capture software to allow the user to easily create or edit their own parts. TinyCad does this. TinyCad also allows hierarchical designs. It exports the schematic in half a dozen standard formats, including: PADS-PCB, Protel, Eagle and XML. It supports SPICE (but has no libraries). Most impressive is the 74 page PDF user manual. Naturally, there are user groups and other support features. But this is only half of what I wanted. I also needed a good PCB layout program. I found one at www.freepcb.com. FreePCB is a very powerful and flexible layout tool with a nice user manual. It incorporates three important features.

The first is that it will link to a schematic file (including TinyCAD!). This allows FreePCB to display a 'rat's nest' which shows all the pin connections (as thin, straight lines). This makes the layout easy by showing you which pin goes where. This greatly reduces the chance of mis-wiring a trace. Additionally you can see how the placement of parts will affect the traces. Fewer and shorter traces is better. The second feature is a 'Design Rule Check'. This verifies that all traces, pads and copper areas are separated from each other by at least a minimum distance (that you specify). This can be used to improve manufacturability and reliability. But, perhaps more importantly, it will identify any mis-wiring that may have occurred. Obviously, it's much better to catch an error before you make the PCB, than after.

Lastly, FreePCB creates a standard Gerber file output (with associated files). Virtually every PCB manufacturing house accepts this format. So you are not limited to any particular PCB fabricator. There is one weak point. FreePCB cannot print your design on a printer. However, there are free Gerber 'Viewers' that will display the Gerber file and they do allow you to print out your PCB image. Alternatively, you can take a screen shot (shift+Print Screen) and paste it into a word-processing program and then print that. There is an auto-router companion program for FreePCB that I haven't tried. I prefer to manually route my boards. Simple point-to-point routers can create hard-to-find problems due to crosstalk and other subtle layout complications.

Free for All

A legitimate question is why these (and other) valuable tools are completely free. I don't know for sure. But I suspect that these people are proud of what they've created (I would be). Like an artist displaying their paintings, these authors use the web to publicize their works. But unlike the artist, the brainchildren of these engineers are functional creations.

As little as ten to fifteen years ago, this suite of three CAD tools would have cost at least a thousand dollars. If someone had told me that they would be free in the near future, I would have just laughed and replied: "The next thing you'll say is that money grows on trees."



(120613)

ELEKTOR SHOWCASE

To book your showcase space contact Strategic Media Marketing Inc.

Tel. 1-978-281-7708

Fax 1-978-281-7706

Email ElektorUSA@smmarketing.us

SCOPES and more

HAMEG[®]
Instruments

A Rohde & Schwarz Company

www.hameg.com

**Great Value in
TEST & MEASUREMENT**

Pololu
Robotics & Electronics

Save 10% on motor controllers with coupon code **ELEKTORMC94**

Robot chassis, wheels, casters, motors & servos

Jrk USB motor controller with feedback \$49.95

RP5 chassis \$49.95

Controllers, sensors, cables, batteries & more!

Plastic SMT stencils & custom laser cutting

www.pololu.com

**LISTEN
TO YOUR MACHINES**
Ethernet PLCs for OEMs



From \$119

- Built-in Ethernet
- MODBUS TCP/IP
- Digital and Analog I/Os
- PWM/PID/Stepper Control

Tel : 1 877 TRI-PLCS
web : www.tri-plc.com/ek.htm

TRI TRIANGLE RESEARCH INTERNATIONAL

MaxSonar
Ultrasonic Ranging is EZ



LV-ProxSonar-EZ

- People sensing made easy
- Multi-sensor operation
- Low power requirements
- Fast measurement cycle
- MSRP \$29.95

I2CXL-MaxSonar-EZ

- Incredible noise immunity
- I2C interface
- Automatic calibration
- Great for UAV's or Robotics
- Starting at \$39.95



www.maxbotix.com

NEW BUD Board

BUD is a compact Atmel ATmega640 microcontroller board with a big extra: when a BOB-4 module is installed, BUD easily generates complex text and graphics on TV monitors.

INTRODUCTORY PRICING THROUGH DECEMBER 2012!

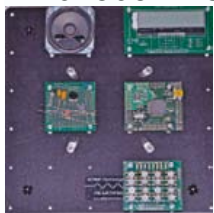


www.decadenet.com



DECADE ENGINEERING
503-743-3194 Turner, OR, USA

MICROCONTROLLER KITS



EX-MORSE-1
Pictured

**LEARN
CREATE
EXPLORE**



ATRIA Technologies Inc
www.AtriaTechnologies.com

BPS BusBoard Prototype Systems www.BusBoard.us

Many PCB styles available

SB400 Solderable PC BreadBoard

BB400T Transparent BreadBoard

Solder and Keep

Build and Test

Available from:

JAMECO ELECTRONICS

amazon.com

MOUSER ELECTRONICS
a TI company

**TO BOOK YOUR SHOWCASE SPACE
CONTACT STRATEGIC MEDIA MARKETING INC.**

Tel. 1-978-281-7708

Fax 1-978-281-7706

Email ElektorUSA@smmarketing.us

Hexadoku

Puzzle with an electronics touch

Here's another Hexadokunut to crack for those having had enough of all them schematics and software listings but still wanting a light mental challenge.

Enter the right numbers or letters A-F in the open boxes, find the solution in the gray boxes, send it to us and you automatically enter the prize draw for one of four Elektor Shop vouchers.

The instructions for this puzzle are straightforward. Fully geared to electronics fans and programmers, the Hexadoku puzzle employs 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 draw for a main prize and three lesser prizes. 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 one Elektor Shop voucher worth \$ 125.00* and three Elektor Shop Vouchers worth \$ 60.00* each, which should encourage all Elektor readers to participate.

*Subject to exchange rate.

Participate!

Before December 1, 2012,

send your solution
(the numbers in the gray boxes) by email to

hexadoku@elektor.com

Prize winners

The solution of the September 2012 Hexadoku is: 3F126.
The Elektor \$125.00 voucher has been awarded to William Neumann (Canada).
The Elektor \$60.00 vouchers have been awarded to Yves Printems (France),
Jean-Louis Vidaud (France) and Eugene Stemple (USA).
Congratulations everyone!

	E		A	6		9	D		4		2	7	8	C
1		2	9		B	7	C			8		3		
	B				2			3			A			
5	8		C		3			0	A	9				1
							7		4					5
B				4	5						7			8
	5	7	1		C				D	0		9		
2	6					B				C	E		D	F
6	7			8				2		E	F	C	3	
		4	D				8			3			2	7
3					4							6		B
	C		5	0		6	2	4	7					E
A		0	F		D		6	3						
9	1				C		A		2			F	B	
8						E	5	C				2		D
C			7	2	5		3		E	6	F			1

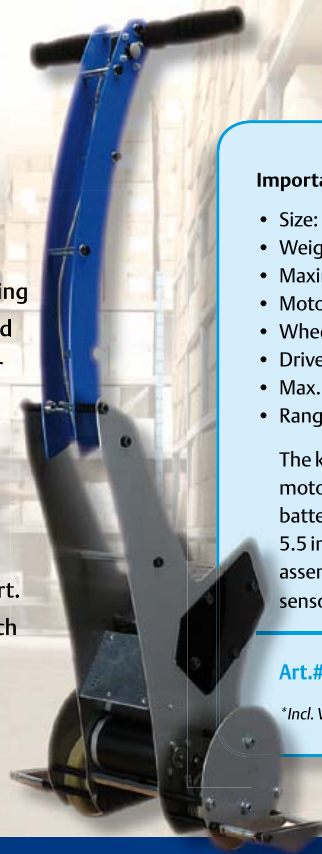
1	4	C	8	9	3	B	A	2	5	7	E	0	6	D	F
7	9	2	B	C	0	D	E	6	4	A	F	3	8	1	5
A	6	D	3	1	F	4	5	8	0	9	B	2	E	7	C
E	F	0	5	2	6	8	7	1	3	C	D	4	9	A	B
0	7	3	C	B	2	9	F	4	8	E	1	D	A	5	6
9	1	8	F	3	4	0	D	5	6	B	A	E	7	C	2
2	5	E	D	6	7	A	1	C	9	0	3	8	F	B	4
4	A	B	6	E	C	5	8	D	7	F	2	1	0	9	3
C	8	A	4	D	9	E	3	F	1	2	6	5	B	0	7
F	2	7	9	0	A	C	4	3	B	8	5	6	1	E	D
6	D	1	E	5	8	2	B	7	A	4	0	F	C	3	9
3	B	5	0	F	1	7	6	9	E	D	C	A	2	4	8
5	C	9	7	8	D	6	0	A	F	1	4	B	3	2	E
B	0	F	A	4	5	3	C	E	2	6	9	7	D	8	1
8	3	6	2	A	E	1	9	B	D	5	7	C	4	F	0
D	E	4	1	7	B	F	2	0	C	3	8	9	5	6	A

The competition is not open to employees of Elektor International Media, its business partners and/or associated publishing houses.

Elektor OSPV

Open Source Personal Vehicle

Last year we launched the Elektor Wheelie, a self-balancing personal transport device. Our new Elektor OSPV is based on the same concept, but with the difference that it's for indoors, it's easy to steer, it's light and foldable and... it's open source. You can configure or modify it to suit your wishes! The OSPV is primarily intended for moving people, but it doesn't have to be limited to that. A variety of other uses are conceivable, ranging from an electric wheelbarrow to a handy motorised shopping cart. This is where the advantages of the open source approach come to the fore!



Price Slashed!

Important specifications:

- Size: 147.2x18.5x18.5 inch (HxWxD)
- Weight: 25lbs
- Maximum load: 200 lbs
- Motors: DC 2 x 200 W
- Wheels: Polyurethane, 5.5 inch dia.
- Drive train: HDT toothed belt
- Max. speed: 9.3 mph
- Range: 5 miles

The kit comprises two 200-watt DC drive motors, two 12-V lead-acid AGM batteries, battery charger, two wheels Polyurethane 5.5 inch, casing, control lever and fully assembled and tested control board with sensor board.

Art.# 110320-91 • ~~\$1570.00~~ \$1280.00*

*Incl. VAT, excl. shipping costs

Further information, demo video and ordering at www.elektor.com/ospv

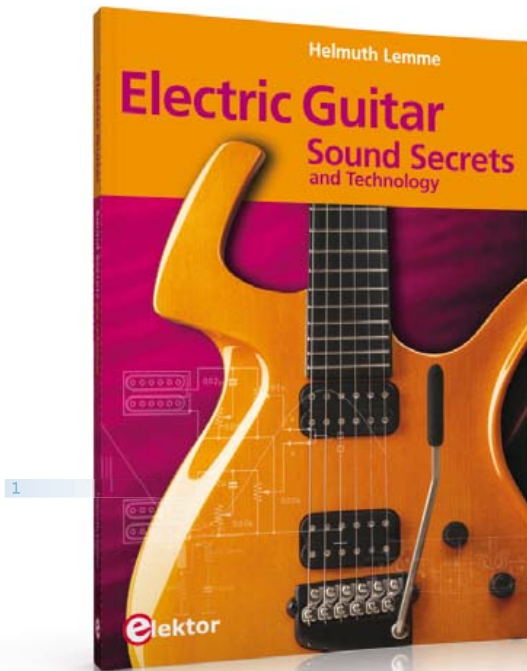
JOIN THE WORLD'S LEADING PLATFORM FOR ELECTRONICS ENGINEERS

Take out a FREE membership to Elektor Weekly

- The latest on electronics and information technology
- Hints, tips and interesting offers
- In your own mailbox each Friday



Register today at www.elektor.com/newsletter



1



Limited Time Offer for Elektor Magazine Members!
15% DISCOUNT
www.elektor.com/guitar

Sound Secrets and technology

1 **Electric Guitar**

What would today's rock and pop music be without electric lead and bass guitars? These instruments have been setting the tone for more than forty years. Their underlying sound is determined largely by their electrical components. But, how do they actually work? This book answers many questions simply, in an easily-understandable manner. For the interested musician (and others), this book unveils, in a simple and well-grounded way, what have, until now, been regarded as manufacturer secrets. The examination explores deep within the guitar, including pickups and electrical environment, so that guitar electronics are no longer considered highly secret. With a few deft interventions, many instruments can be rendered more versatile and made to sound a lot better – in the most cost-effective manner.

Approx. 280 pages • ISBN 978-1-907920-13-4
 US \$47.60

More than 75,000 components

2 **CD Elektor's Components Database 7**

This CD-ROM gives you easy access to design data for over 11,100 ICs, 37,000 transistors, FETs, thyristors and triacs, 25,100 diodes and 2,000 optocouplers. The program package consists of eight

databanks covering ICs, transistors, diodes and optocouplers. A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data.

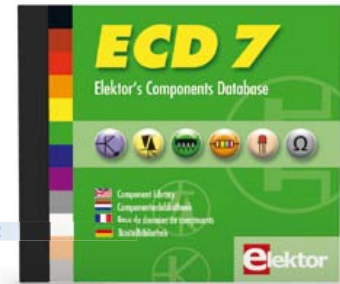
ISBN 978-90-5381-298-3 • US \$40.20

LabWorX 2

3 **Mastering Surface Mount Technology**

This book takes you on a crash course in techniques, tips and know-how to successfully introduce surface mount technology in your workflow. Even if you are on a budget you too can jumpstart your designs with advanced fine pitch parts. Besides explaining methodology and equipment, attention is given to SMT parts technologies and soldering methods. Many practical tips and tricks are disclosed that bring surface mount technology into everyone's reach without breaking the bank. A comprehensive kit of parts comprising all SMT components, circuit boards and solder stencils is available for readers wishing to replicate three projects described in this book.

282 pages • ISBN 978-1-907920-12-7
 US \$47.60



2



3

Elektor Linux Board

4 **Embedded Linux Made Easy**

Today Linux can be found running on all sorts of devices, even coffee machines. Many electronics enthusiasts will be keen to use Linux as the basis of a new microcontroller project, but the apparent complexity of the operating system and the high price of development boards has been a hurdle. Here Elektor solves both these problems, with a beginners' course accompanied by a compact and inexpensive populated and tested circuit board. This board includes everything necessary for a modern embedded project: a USB interface, an SD card connection and various other expansion options. It is also easy to hook the board up to an Ethernet network.

Populated and tested Elektor Linux Board
 Art.# 120026-91 • US \$93.30

A comprehensive and practical how-to guide

5 **Design your own PC Visual Processing and Recognition System in C#**

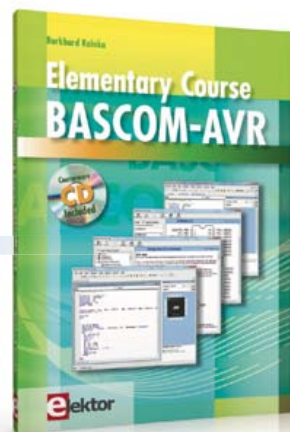
This book is aimed at Engineers, Scientists and enthusiasts with developed programming skills or with a strong interest in image processing technology on



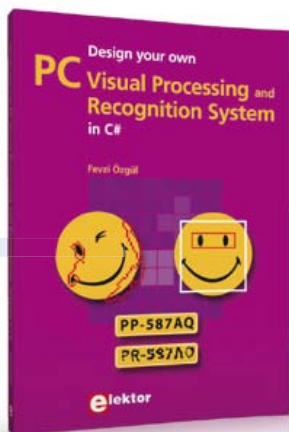
4



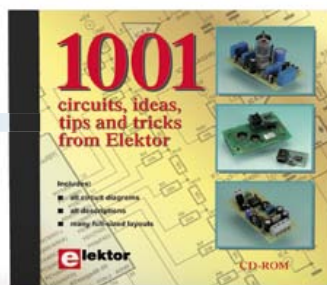
6



7



5



8



9

a PC. Written using Microsoft C# and utilizing object-oriented practices, this book is a comprehensive and practical how-to guide. The key focus is on modern image processing techniques with useful and practical application examples to produce high-quality image processing software. Analysis starts with a detailed review of the fundamentals of image processing. It progresses to explain and explore the practical uses of two highly sophisticated and freely downloadable, open source image processing libraries; AForge.NET and Emgu.CV, utilizing dotnet technology within the Microsoft Visual Studio environment. All code examples used are available – free of charge – from the Elektor website; you can easily create and develop your own results to demonstrate the concepts and techniques explained.

307 pages • ISBN 978-1-907920-09-7
US \$57.30

**Ultrasensitive wideband E-smog detector
TAPIR Sniffs it Out!**

6

Attention boy scouts, professionals and grandfathers! This ultrasensitive wideband E-smog detector offers you two extra senses to track down noise that's normally inaudible. TAPIR — short for Totally Archaic but Practical Interceptor of Radiation — also makes a nice project to build: the kit comprises everything you need. Even the enclosure, ingeniously consisting of the PCB proper! Using the TAPIR is dead easy.

Connect the headphones and an antenna and switch it on. Move it around any electrical device and you'll hear different noises with each device, depending on the type and frequency of the emitted field.

Kit of parts, incl. PCB
Art.# 120354-71 • US \$21.50

**Free Software CD-ROM included
7 Elementary Course
BASCOM-AVR**

7

The Atmel AVR family of microcontrollers are extremely versatile and widely used. In Elektor magazine we have already published many interesting applications employing an ATmega or ATtiny microcontroller. The majority of these projects perform a particular function. In this book we focus more on the software aspects. Using lots of practical examples we show how, using BASCOM, you can quickly get your own design ideas up and running in silicon.

224 pages • ISBN 978-1-907920-11-0
US \$56.40

**Circuits, ideas, tips and tricks
8 CD 1001 Circuits**

8

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Circuits issues 2001-2010 of Elektor, supplemented with various

other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & modelling, home & garden, high frequency, power supply, robotics, test & measurement and of course a section miscellaneous for everything that didn't fit in one of the other sections.

ISBN 978-1-907920-06-6 • US \$55.70

**Package Deal: 12% off
9 AVR Software
Defined Radio**

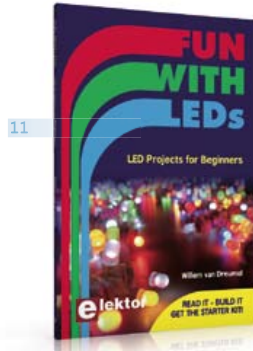
9

This package consists of the three boards associated with the AVR Software Defined Radio articles series in Elektor, which is built around practical experiments. The first board, which includes an ATtiny2313, a 20 MHz oscillator and an R-2R DAC, will be used to make a signal generator. The second board will fish signals out of the ether. It contains all the hardware needed to make a digital software-defined radio (SDR), with an RS-232 interface, an LCD panel, and a 20 MHz VCXO (voltage-controlled crystal oscillator), which can be locked to a reference signal. The third board provides an active ferrite antenna.

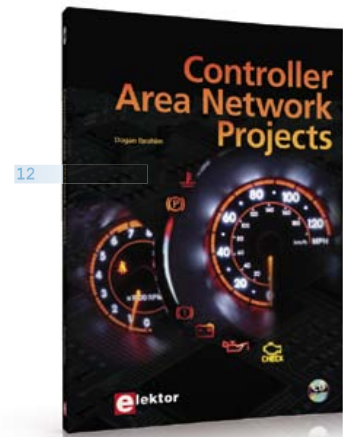
Signal Generator + Universal Receiver + Active Antenna: PCBs and all components + USB-FT232R breakout-board
Art.# 100182-72 • US \$133.00



10



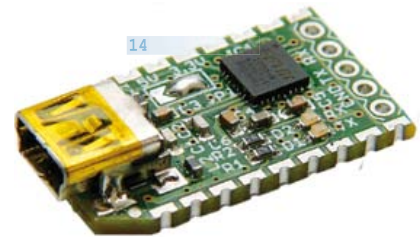
11



12



13



14

Avoid interference and earth loops

10 USB Isolator

If your USB device ever suffers from noise caused by an earth loop or if you want to protect your PC against external voltages then you need a USB isolator. The circuit described in Elektor's October 2012 edition offers an optimal electrical isolation of both the data lines as well as the supply lines between the PC and the USB device.

[Populated and tested Board](#)
[Art.# 120291-91 • US \\$101.40](#)

Associated 60-piece Starter Kit available

11 Fun with LEDs

This booklet presents more than twenty exciting projects covering LEDs, aimed at young & old. From an Air Writer, a Party Light, Running Lights, a LED Fader right up to a Christmas Tree. Use this book to replicate various projects and then put them into practice. To give you a head start each project is supported by a brief explanation, schematics and photos. In addition, the free support page on the Elektor website has a few inspiring video links available that elaborate on the projects. A couple of projects employ the popular Arduino microcontroller board that's graced by a galaxy of open source applications. The optional 60-piece Starter Kit available with this book is a great way to get circuits

built up and tested on a breadboard, i.e. without soldering.

[96 pages • ISBN 978-1-907920-05-9](#)
[US \\$38.00](#)

Free mikroC compiler CD-ROM included

12 Controller Area Network Projects

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters.

[260 pages • ISBN 978-1-907920-04-2](#)
[US \\$47.60](#)

Bridging Android and your electronics projects

13 AndroPod

With their high-resolution touchscreens, ample computing power, WLAN support and telephone functions, Android smartphones and tablets are ideal for use as control centres in your own projects. However, up to now it has been rather difficult to connect them to external circuitry. Our AndroPod interface board,

which adds a serial TTL port and an RS485 port to the picture, changes this situation.

[Andropod module with RS485 Extension](#)
[Art.# 110405-91 • US \\$74.70](#)

Meet BOB

14 FT232R USB/Serial Bridge/BOB

You'll be surprised first and foremost by the size of this USB/serial converter – no larger than the moulded plug on a USB cable! And you're also bound to appreciate that fact that it's practical, quick to implement, reusable, and multi-platform – and yet for all that, not too expensive! Maybe you don't think much of the various commercially-available FT232R-based modules. Too expensive, too bulky, badly designed, That's why this project got designed in the form of a breakout board (BOB).

[PCB, assembled and tested](#)
[Art.# 110553-91 • US \\$20.90](#)

110 issues, more than 2,100 articles

15 DVD Elektor 1990 through 1999

This DVD-ROM contains the full range of 1990-1999 volumes (all 110 issues) of Elektor Electronics magazine (PDF). The more than 2,100 separate articles have been classified chronologically by their dates of publication (month/year), but are also listed alpha-



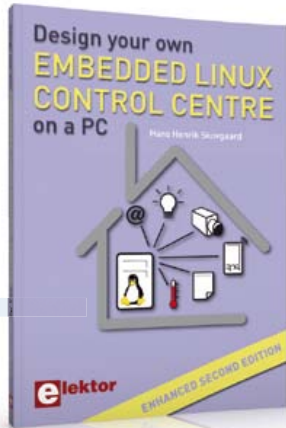
15



17



19



16



18

betically by topic. A comprehensive index enables you to search the entire DVD. What's more, this DVD also contains the entire 'The Elektor Datasheet Collection 1...5' CD-ROM series.

ISBN 978-0-905705-76-7 • US \$111.30

Enhanced second edition

16 Design your own Embedded Linux Control Centre on a PC

The main system described in this book reuses an old PC, a wireless mains outlet with three switches and one controller, and a USB webcam. All this is linked together by Linux. This book will serve up the basics of setting up a Linux environment – including a software development environment – so it can be used as a control centre. The book will also guide you through the necessary setup and configuration of a webserver. New edition enhancements include details of extending the capabilities of your control center with ports for a mobile phone (for SMS messaging) and the Elektor "thermo snake" for lowcost networked real-time thermal monitoring of your house and outbuildings. Now you can additionally also send all kinds of useful temperature and sensor warnings to a mobile phone. All software needed will be available at the Elektor website.

416 pages • ISBN 978-1-907920-02-8

US \$55.70

Processor design in the real world

17 Microprocessor Design using Verilog HDL

If you have the right tools, designing a microprocessor shouldn't be complicated. The Verilog hardware description language (HDL) is one such tool. This book is a practical guide to processor design in the real world. It presents the Verilog HDL in an easily digestible fashion. You're led through the microprocessor design process from the start to finish, and essential topics ranging from writing in Verilog to debugging and testing are laid bare.

340 pages • ISBN 978-0-9630133-5-4

US \$45.00

18 Counter for alpha, beta and gamma radiation Improved Radiation Meter

This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radio-active samples. The photodiode has a smaller sensitive area than a Geiger-Müller tube and so has a lower background count rate, which in turn means that the radiation from a small sample is easier to detect against the background. A further advantage of a semiconductor sensor is that it offers the possibility of measuring the energy of each particle.

Kit of parts incl. display and programmed controller

Art.# 110538-71 • US \$57.30

Dual-layer DVD: 165 mins. video

19 DVD Modern Valve Electronics

This filmed seminar (presented by Menno van der Veen) starts with a short discussion of the classic approach using valve load line graphs, followed by current sources and current foldback techniques. Next, the negative effect of cathode electrolytics is covered as well as reducing supply voltage interference. With the help of state of the art measurement techniques the (in)correctness of feedback is proven, while also clarifying what's happening deep within the core of the output transformer.

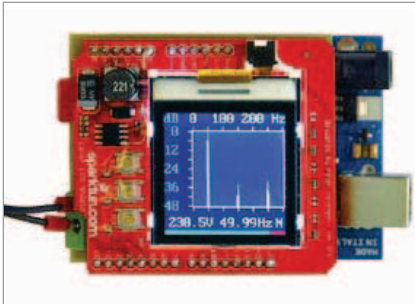
ISBN 978-1-907920-10-3 • US \$40.20

More information on the Elektor Website:

www.elektor.com/store

Elektor USA
4 Park Street
Vernon, CT 06066 – USA
Phone: 860-875-2199
Fax: 860-871-0411
Email: order@elektor.com





Arduino AC Powerline Analyzer

A simple Arduino-board should not be underestimated in terms of performance. With this project we demonstrate the power of a small 8-bit microcontroller, not forgetting of course to pinpoint its limitations. The spectrum analyzer discussed in this article uses FFT to analyze and quantify the harmonics of the mains frequency and shows the results on a display. We also explain how the components of the microcontroller can be used in an efficient way and what free development tools are necessary to set up this type of project.



USB I/O Interface Cable

Nowadays most computers only have USB connections for communication with the outside world. Surely it's useful to employ that interface for your own measurements and control purposes. We developed an interface cable based on a USB-TTL cable from FTDI, to which a small PCB with an R8C microcontroller is added for fitting into a DB25-plug housing. The controller handles the communication with the PC, and at the output side offers 24 pins for the user to configure and do a variety of measuring and switching tasks.



The 7-Up Alarm Clock

How many people are really satisfied about the digital alarm clock on the nightstand? A few, maybe. It's high time for someone to design a comprehensive alarm clock that really has all the features you'd want it to have. These requirements have resulted in this design based on an Atmel AT89C5131 USB microcontroller that distinguishes itself clearly in terms of functionality from the usual digital alarm clocks in the shops. The device is even equipped with a USB socket for communication with a PC.

Article titles and magazine contents subject to change; please check the Magazine tab on www.elektor.com

Elektor UK/European December 2102 edition on sale November 29, 2012.

Elektor USA December 2012 edition on publication November 26, 2012.

www.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com

Elektor on the web

All magazine articles back to volume 2000 are available individually in pdf format against e-credits. Article summaries and component lists (if applicable) can be instantly viewed to help you positively identify an article. Article related items and resources are also shown, including software downloads, hyperlinks, circuit boards, programmed ICs and corrections and updates if applicable.

In the Elektor Shop you'll find all other products sold by the publishers, like CD-ROMs, DVDs, kits, modules, equipment, tools and books. A powerful search function allows you to search for items and references across the entire website.

Also on the Elektor website:

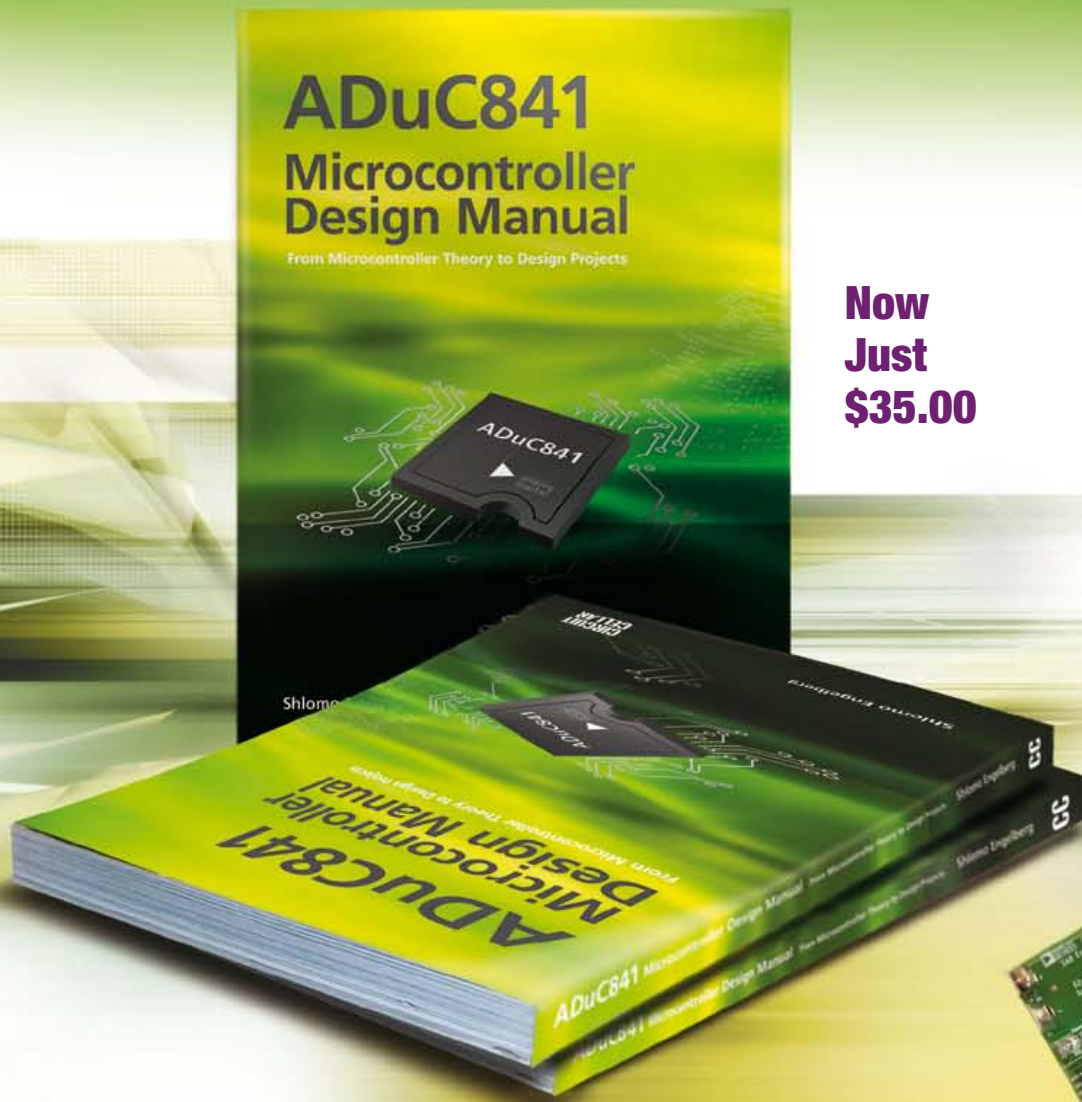
- Electronics news and Elektor announcements
- Readers Forum
- PCB, software and e-magazine downloads
- Time limited offers
- FAQ, Author Guidelines and Contact



CIRCUIT CELLAR

ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects

If you've ever wanted to design and program with the ADuC841 microcontroller, or other microcontrollers in the 8051 family, this is the book for you. With introductory and advanced labs, you'll soon master the many ways to use a microcontroller. Perfect for academics!



**Now
Just
\$35.00**



Buy it today!
www.cc-webshop.com

ORDERING INFORMATION

To order contact customer service:

Phone: 860-875-2199

Fax: 860-871-0411

Mail: Elektor US
4 Park Street
Vernon, CT 06066
USA

E-mail: order@elektor.com

On-line at www.elektor.com/store

Customer service hours: 8:30 AM-4:30 PM EST Monday-Friday. Voice mail available at other times. When leaving a message please be sure to leave a daytime telephone number where we can return your call.

PLEASE NOTE: While we strive to provide the best possible information in this issue, pricing and availability are subject to change without notice. To find out about current pricing and stock, please call or email customer service.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in the magazine. If difficulties in obtaining components are suspected, a source will normally be identified in the article. Please note, however, that the source(s) given is (are) not exclusive.

PAYMENT

Orders must be prepaid. We accept checks or money orders (in US \$ drawn on a US bank only), VISA, Mastercard, Discover, and American Express credit cards. We do not accept C.O.D. orders. We also accept wire transfers. Add \$20 to cover fees charged for these transfers.

TERMS OF BUSINESS

Shipping Note: All orders will be shipped from Europe. Please allow 3-4 weeks for delivery. Shipping and handling via airmail: \$20.00 per order.

Returns

Damaged or miss-shipped goods may be returned for replacement or refund. All returns must have an RA #. Call or email customer service to receive an RA# before returning the merchandise and be sure to put the RA# on the outside of the package. Please save shipping materials for possible carrier inspection. Requests for RA# must be received 30 days from invoice.

Patents

Patent protection may exist with respect to circuits, devices, components, and items described in our books and magazines. Elektor accepts no responsibility or liability for failing to identify such patent or other protection.

Copyright

All drawing, photographs, articles, printed circuit boards, programmed integrated circuits, diskettes, and software carriers published in our books and magazines (other than in third-party advertisements) are copyrighted and may not be reproduced (or stored in any sort of retrieval system) without written permission from Elektor. Notwithstanding, printed circuit boards may be produced for private and personal use without prior permission.

Limitation of liability

Elektor shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connection with, the supply of goods or services by Elektor other than to supply goods as described or, at the option of Elektor, to refund the purchaser any money paid with respect to the goods.

MEMBERSHIPS (US & CANADA ONLY)

Order memberships on-line at www.elektor.com/members

All memberships begin with the current issue. Expect 3-4 weeks for receipt of the first issue. Membership renewals and change of address should be sent to:

Elektor USA
P.O. Box 462228
Escondido, CA 92046

E-mail: elektor@pcspublink.com

Memberships may be paid for by check or money order (in US \$ drawn on a US bank only). We accept Mastercard, VISA, Discover and American Express credit cards.

For gift memberships, please include gift recipient's name and address as well as your own, with remittance. A gift card will be sent on request. Memberships may be cancelled at any time for a refund of all unmailed issues.

Does your membership expire soon?

Renew it on-line at www.elektor.com/members



SPECIAL: SAVE 50% SPECIAL: SAVE 50% SPECIAL

Celebrate Circuit Cellar's **25th Anniversary**



\$25 Print or Digital :: **\$50** Combo

Celebrate *Circuit Cellar's 25th year* of bringing readers insightful analysis of embedded electronics technology.

Visit www.circuitcellar.com/el912
to take advantage of these great deals.

BONUS OFFER! BONUS OFFER! BONUS OFFER! BONUS OFF

Sign up today and you'll also receive the **Special 25th Anniversary Edition** with your subscription!



WIRELESS POWER: CHARGING INNOVATION



Wireless power integration made easy with TI's Qi Compliant Wireless Power bqTESLA evaluation modules and solution portfolio from TI & Würth enable design engineers to easily accelerate the integration of wireless power technology in consumer electronics, such as smart phones, digital cameras, MP3 players, along with infrastructure applications such as furniture and cars. We'll help you cross the design finish line in record time

with a wide range of evaluation modules both on the transmitter and receiver side to help reduce the design cycle of wireless power solutions. Whether implementing wireless charging within an existing design, or adding it to a new one, we've got the tools, support and expertise to help you – cut the cord! Make your own kit by selecting a TI transmitter and receiver module with corresponding Charging-Coils provided by Würth Elektronik.



element14

element14.com/wireless-power-solution