

eLektor

www.elektor.com



Arduino, Thou Shalt Communicate!

Wi-Fi / Bluetooth / USB shield for Platinino and Arduino

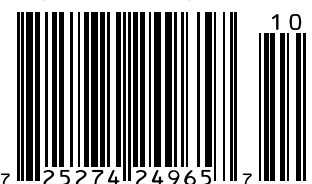
- + USB Isolator**
Avoid interference and earth loops
- + Vocal Annunciator**
How about making your next project speak?
- + Intelligent PC Fan Controller**
Control, monitor & configure a maximum of six PWM fans

Radiation Meter Reloaded

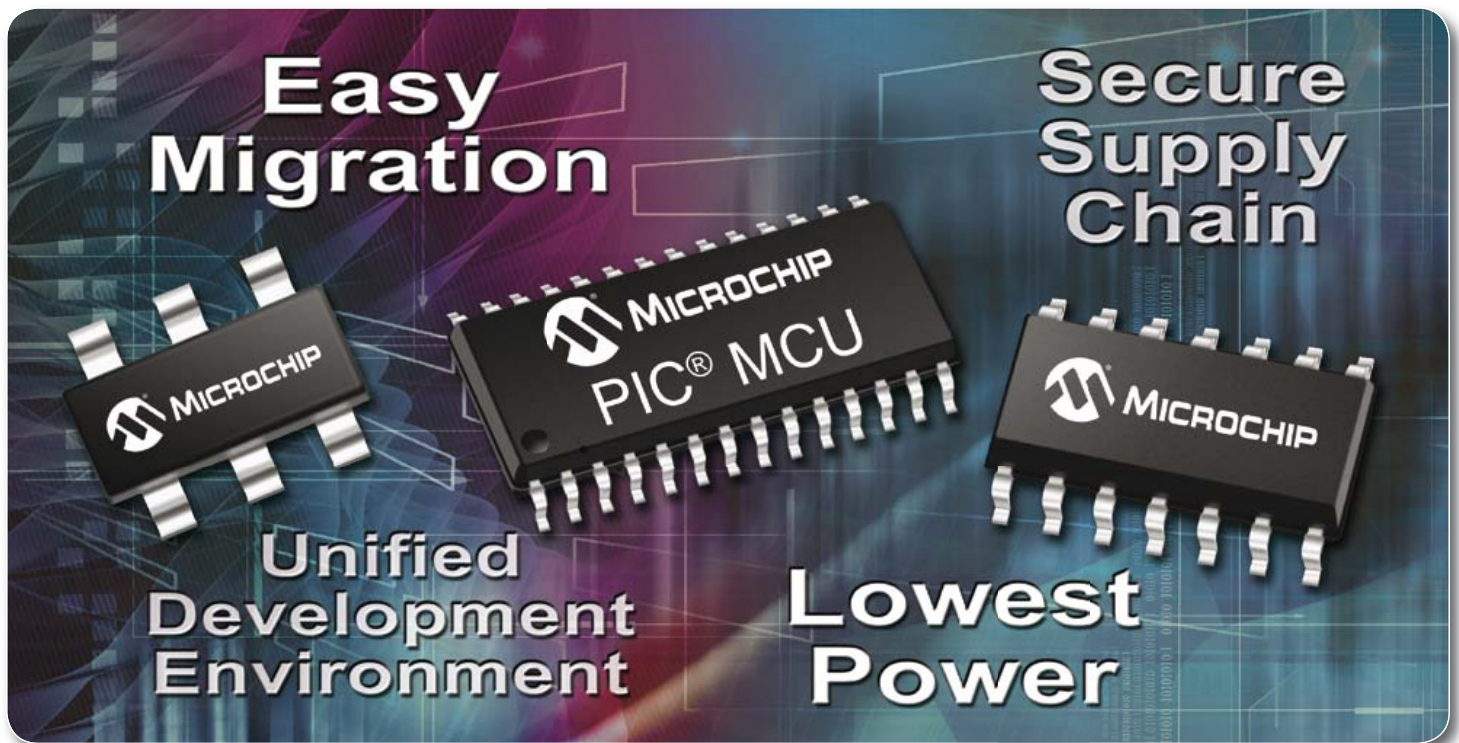
**More sensors,
more readout options**



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

one SOCKET for 30 click boards



Just place the Click™ board you need into the innovative new mikroBUS™ socket

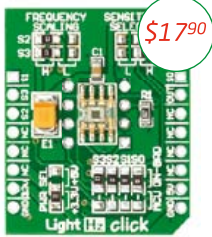
and it's ready to work right away. Adding **new functionality** to your device was never so easy.



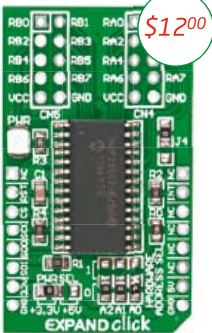
\$49⁰⁰



\$18⁰⁰



\$17⁹⁰



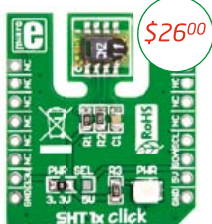
\$12⁰⁰



\$10⁰⁰



\$16⁹⁰



\$26⁰⁰



\$25⁹⁰



\$9⁰⁰



\$16⁰⁰



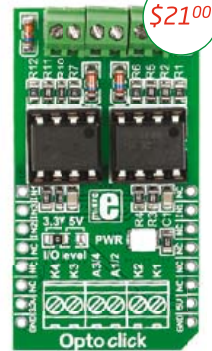
\$45⁰⁰



\$21⁰⁰



\$41⁹⁰



\$21⁰⁰



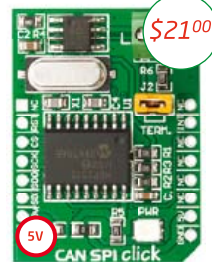
\$29⁰⁰



\$21⁹⁰



\$8⁰⁰



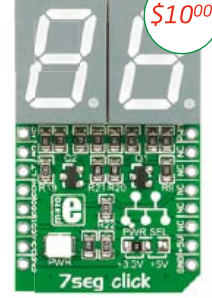
\$21⁰⁰



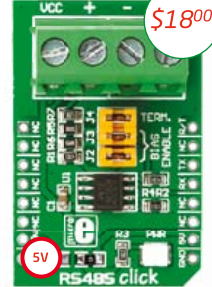
\$29⁰⁰



\$13⁹⁰



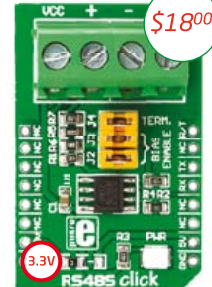
\$10⁰⁰



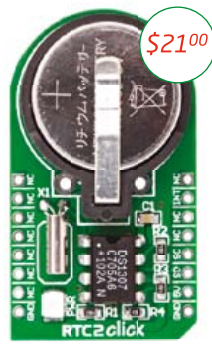
\$18⁰⁰



\$13⁹⁰



\$18⁰⁰



\$21⁰⁰



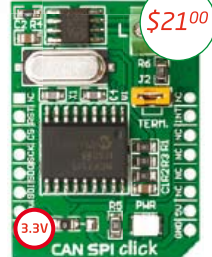
\$44⁰⁰



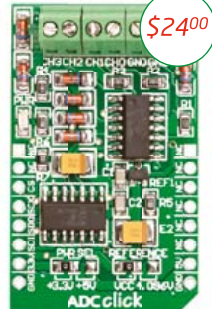
\$19⁰⁰



\$24⁰⁰



\$21⁰⁰



\$24⁰⁰



Looking forward to being inundated (again)

Here at Elektor we get the numbers all wrong occasionally. Although we're perfectly able to calculate, say, bit time offsets to less than a microsecond using C, or Johnson noise levels in an NE5532 opamp, we were horribly wrong with our guesstimate of the interest in our Embedded Linux article series. But happy to be wide off the mark! Let me explain.

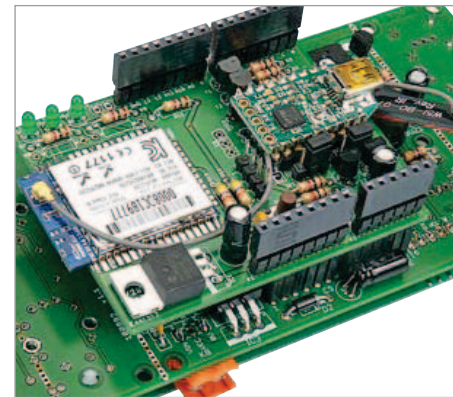
On Friday August 17, while working from my home on a new book called *LabworX 2: Mastering Surface Mount Technology*, I wanted to correct a small bit of text somewhere on the elektor.com website using my administrator rights. Now I have a pretty slow PC at home, but this time the backoffice login procedure seemed to take so long I decided to do a small inspection round of my garden. On returning to my desk, I was greeted by error message -392a on the screen basically saying something was overloaded. Then the telephone rang (it hardly ever does that) and our editorial secretary told me to sort-of-instantly write and email a short message telling customers that our webserver had been on 404 due to a flood of requests for an archive file that belongs with our recent Embedded Linux series.

The problem was fixed almost instantly by moving the archive file to another area with capacity to spare. Apologies to all of you Linux fans and all other Elektor readers seeing 404s and having to wait a while for the file to become available. We have adapted the kernel (read all about it in this edition!), and fitted a pressure release valve on our webserver.

Happy reading,
Jan Buiting, Managing Editor



- 6 Colophon**
Who's who at Elektor.
- 8 News & New Products**
A monthly roundup of all the latest in electronics land.
- 14 USB Isolator**
This little circuit provides electrical isolation of data lines and supply lines between the PC and the USB device.
- 18 Embedded Linux made Easy (4)**
This month we delve deep into the various commands and options offered by the kernel.
- 28 Thou Shalt Communicate!**
Here's a Platino and Arduino compatible shield that packs Wi-Fi, Bluetooth and USB on a single board.
- 34 AVR Software Defined Radio (6)**
In this installment we discuss the time encoding format applied on the BBC Droitwich transmitter on 198 kHz.
- 40 Picoammeter**
This little instrument was designed to cover those infinitesimally small currents between 0.1 pico-ampere to 1 micro-ampere.
- 43 E-Labs Inside:**
 - Shifting RS485 Grounds
 - AVR reset-out-of-the-blue issue
 - E-pollution!
 - Your all-time favorite component
- 46 Energy Harvesting with the LTC3108 & LTC3588-1**
Raymond's pick of the month in terms of components.



CONTENTS

Volume 4
October 2012
no. 46



14 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 here offers an optimal electrical isolation of the data lines as well as the supply lines between the PC and the USB device.



28 Thou Shalt Communicate!

You may have a Facebook account, but what about your oscilloscope? Does your multimeter tweet enough? Is your soldering iron linked in? You may be a non-communicative nerd but your bench power supply may be craving for social interaction. With the add-on board described in this article you can hook up anything to everything, with or without wires.



54 Radiation Meter Reloaded

Elektor's Simple Radiation Meter is currently being used by many readers, mainly in making long-term measurements and in investigating weakly radioactive samples. There are several ways in which we can modify and improve the design, and we look here at a few examples.



68 Intelligent PC Fan Controller

With the help of the circuit described here you can actively control up to six fans, while the temperature can be measured in various places inside the PC case using a number of sensors. A PC program is used to configure and monitor the fans, which communicates with the fan control board via a USB link.

48 Arduino on Course (2)

No matter where your sensors are on the globe, you will be able to read them using Arduino functionality extended with GSM / GPRS.

54 Radiation Meter Reloaded

Elektor's simple radiation meter keeps evolving, both at the front end (the sensors) and the user end (enter the PC!).

60 Electronics for Starters (8)

This series has met with great popularity with young & old. This month we tackle the basics of audio amplification.

64 Vocal Annunciator

Wouldn't it be great to have a measurement system read out values aloud? Here's how.

68 Intelligent PC Fan Controller

This circuit allows six (PC) fans to be controlled, configured and monitored with a good deal of intelligence.

76 Retronics: Marconi Instruments TF801D/1 AM RF Signal Generator

An RF signal generator built like rock and still working great after 50 years. Series Editor: Jan Buiting.

78 Gerard's Columns: Auto Engineering

The monthly contribution from our US columnist Gerard Fonte.

79 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. 46, OCTOBER 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

274932

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 41		Front Panel Express www.frontpanelexpress.com 27
	ARM TechCon www.armtechcon.com 42		Microchip www.microchip.com 2
	Beta Layout www.pcb-pool.com 27		MikroElektronika www.mikroe.com 3
	CES www.cesweb.org 13		Pico www.picotech.com/pco483 88
	Cleverscope www.cleverscope.com 59		Pololu www.pololu.com 41
	DLP Design www.dlpdesign.com 41		Saelig www.saelig.com 11
	ExpressPCB www.expresspcb.com 9		SoC Conference www.socconference.com 53

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

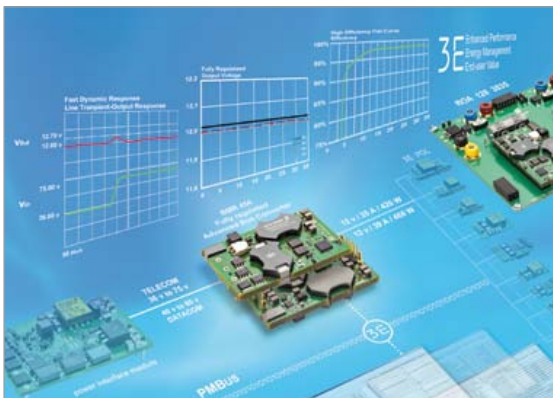
books and magazines (other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning or recording, in whole or in part without prior written permission from the Publisher. Such 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

New digital advanced bus converter lowers power consumption

Ericsson has unveiled the first model in its second generation of digital Advanced Bus Converter (ABC) products. Based upon the FRIDA II platform, which was announced earlier this year, the Ericsson BMR456 3E Advanced Bus Converter delivers unprecedented performance to system architects who are developing equipment for ICT (Information and Communication Technology) applications that require fast response time, tightly regulated intermediate bus voltages and high efficiency at any point of operation to reduce power consumption. This new generation of fully regulated digital DC/DC converters is based on a 32-bit ARM microcontroller that embeds the family's most advanced firmware to date. The FRIDA II firmware has been developed by Ericsson to guarantee the highest possible performance at any point of operation from low load to high load, and to handle line transients, such as those occurring in ATCA (Advanced Telecommunications Computing Architecture) applications when, for example, switching from Feed A to Feed B.



Designed to power telecom and datacom applications, the BRM456 is available in two input voltages ranges: 36 V to 75 V, delivering output power of 420 W; and 40 V to 60 V, delivering output power up to 468 W. Output voltage can be adjusted across a range from 4.0 V to 13.2 V via PMBus commands, making the BMR456 suitable for Dynamic Bus Voltage operation resulting in the reduction of energy consumption when communication data traffic is low.

In addition, taking full advantage of a highly optimized layout and firmware-optimized parameter switching to reduce power losses, the BMR456 delivers a typical efficiency of 96.5% and exhibits flat curve behavior from 14% to 100% load, positioning the product as the most efficient Advanced Bus Converter in its category. Designed for flexibility and high-power applications, when connected in parallel, the Droop Load Sharing (DLS) version of the BMR456 telecom (36-75 V) and datacom (40-60 V) versions can deliver 756 W and 842 W, respectively.

The BMR456 implements the FRIDA II firmware, called the 'Ericsson DC/DC Energy Optimizer', which combines advanced Ericsson Intellectual Property (IP) together with a series of industry-first functionalities to continually optimize switching parameters and reduce energy consumption to an absolute minimum. The firmware is not just limited to energy management, but includes an enviable number of features including the ability to handle input voltage transients with slew-rates of up to 0.5 V/ μ s, while keeping the output voltage within $\pm 10\%$ and ensuring that the output voltage does not trigger over-voltage protection. It also offers the highly efficient management of pre-bias start-up operation and a fully controlled shutdown process, avoiding voltage spikes that could cause an avalanche condition in the secondary-side synchronous rectification MOSFET, thereby contributing to improved reliability.

The BMR456 meets the insulation requirements of EN60950, sustains an I/O isolation voltage of 2250 VDC, and also offers an extensive set of capabilities and features such as: remote sensing; PMBus-configurable protection (over-temperature, over-current and over-voltage); configurable start/stop; precision delay ramp-up; voltage margining; configurable fault response; power good; and extensive power management programmability.

The BRM456 is fully backward compatible with Ericsson's previous generation of Advanced Bus Converters. The device's I/O connector has been designed to guarantee full alignment and co-planarity using different soldering processes for both through-hole and surface-mount assembly. In addition to the mechanical benefits, the Ericsson Advanced Bus Converter footprint offers the necessary flexibility for further board design evolution and the addition of new features currently under evaluation within the ICT industry.

www.ericsson.com/powermodules - (120487-IX)

JTAG development system using WiFi

The Debug Store is proud to announce the availability of the Macraigor Systems WiFiDemon in Europe. Priced at just £1475, the WiFiDemon is in stock and ready to ship. The WiFiDemon is Macraigor's fastest and most versatile OCD interface device for use in the design, debugging and programming of microprocessor and microcontroller based embedded systems.

The WiFiDemon can be connected to the host PC via a 10/100 BaseT Ethernet or WiFi connection allowing the developer to con-



trol the debugging operation some distance from his desk — in fact he can be located across the globe! The WiFiDemon supports configurable JTAG/BDM clock rates from 4.5 kHz to 24 MHz, allowing very fast code download and responsive debugging.

The WiFiDemon is fitted with a 40 GB hard drive, two USB ports (to connect a mouse and keyboard) and a VGA port to connect a monitor. The development tools pre-loaded onto the hard drive can be run on the system allowing development work to be carried out locally — this makes it ideal for use in the field.

www.thedebugstore.com - (120487-IX)

RFID SMT antenna with comprehensive protection for automotive applications

PREMO launches a new family of its TP0702 standard, universally adopted by the industry. This format provides up to 50 mV/ App/m (for 7.2 mH) sensitivity which gives it the best sensitivity in this transponder size. The new TP0702U and TP0702UCAP is an SMD antenna for low frequency 20 kHz-150 kHz receiver applications. TP0702UCAP provide an upper and lateral side protection with co-polyamide polyhexamethylene polymer walls, gamma radiated with high thermal stability (supports up to 290

° C) and mechanical resistance (exceeds 150 Mpa of mechanical strength).

This antenna features a NiZn ferrite core with high surface resistivity (>10 MΩ/mm) that provides a highly stable behavior (better than ±3%) over a wide temperature range (-40 °C to 125 °C).

The new TP0702UCAP is particularly suitable for applications such as TPMS (Tire Pressure Monitoring Systems) which requires an excellent performance under extreme conditions, according to AEC-Q200 and additional requirements as EU regulations.

PREMO offers four standard values, 2.38 mH, 4.91 mH, 7.2 mH and 9 mH at 125 kHz.



Other inductance values and frequencies, from 340 μH to 18.5mH, are available upon request.

A surface mount (SMT) device, the new antenna allows easy use in the automated process of mounting circuit boards, thus eliminating any manual handling.

www.grupopremo.com - (120487-IV)

Rotary position sensor ensures accurate measurement at very high rotation speeds

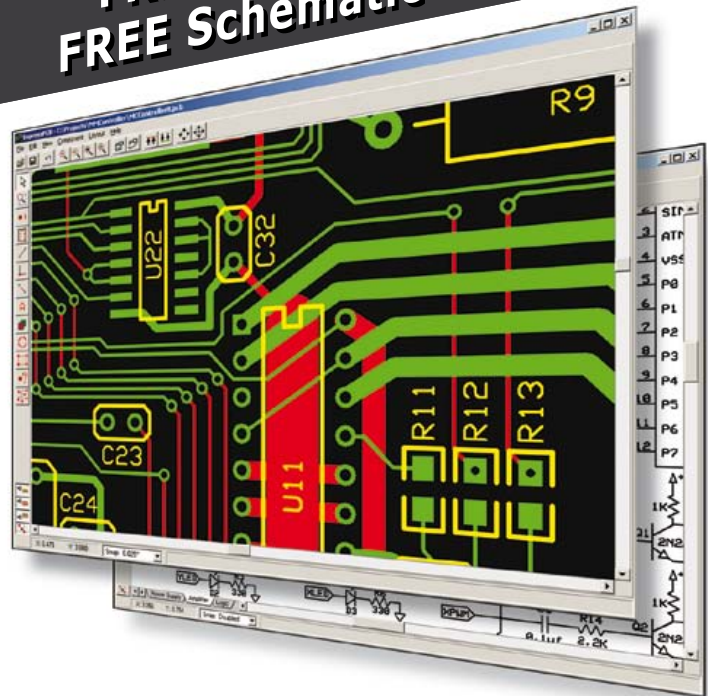
ams, a leading worldwide designer and manufacturer of high performance analog ICs for consumer & communications, industry & medical and auto-motive applications, today launched the AS5132, a magnetic rotary position sensor IC offering accurate angular measurement even at very high rotation speeds. ams is the new name of austriami-



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

croscystems, following the 2011 acquisition of optical sensor company TAOS Inc.

The AS5132 is a system-on-chip, combining integrated Hall elements, an analog front end and digital signal processing in a single device. It is particularly well suited to industrial and automotive brushless DC (BLDC) motors, where it offers designers a small, robust and easily assembled position sensing solution. It is highly resistant to interference from stray magnetic fields. The new device improves on previous gen-

erations of magnetic encoder by dynamically compensating for angle errors attributable to propagation delay, which are most noticeable at high speeds.

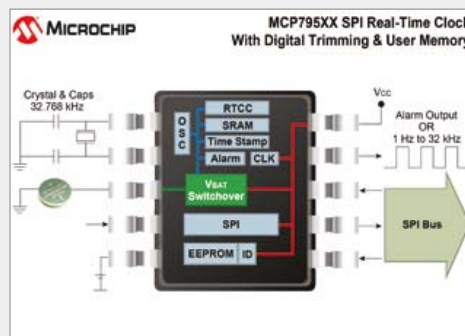
Error compensation is achieved through integrated pre-commutation functions. These are very easy to configure in the AS5132, and do not require external software routines to be executed, therefore unburdening host microcontrollers or electronic control units (ECUs). The configuration can be changed while the device is

SPI real-time clock/calendars in smaller, 10-pin packages

Microchip recently announced the expansion of its stand-alone Real-Time Clock/Calendar (RTCC) portfolio with the new 10-pin, SPI MCP795XX family. These new devices offer many of the same features as the larger 14-pin MCP795WXX family, including superior timekeeping performance.

Reducing overall component count in the system and eliminating the user programming costs for a serial ID make the MCP795XX RTCC family an ideal choice for the handheld, wireless and consumer markets. By including 64 bytes of SRAM, 2 Kbits of EEPROM and a 128-bit Unique ID, which can be ordered blank or preprogrammed with a MAC address, extra memory devices may not be needed. Utility power meters, manufacturing equipment, radios, GPS and hospital instrumentation applications that need accurate

time over a broad temperature range will also benefit from the very wide digital trimming range, which can compensate up to 22 seconds per day for crystal frequency drift. Digital trimming improves the timekeeping accuracy of RTCCs, and a wide digital trimming range provides customers with high accuracy over a large temperature range. Accurate timekeeping is also supported at a decreased power-consumption level, because digital trimming is maintained when the MCP795XX is operating from backup power on the VBAT input. Additionally, these devices join the industry's only portfolio to distinctively offer a battery-backed RTCC with both power-fail and power-restore timestamps plus three types of non-volatile memory: EEPROM, SRAM and a Unique ID. When the time and duration of power failures need to be logged, the MCP795XX family can accomplish this without adding any circuitry by using the on-chip power-fail timestamp feature, which is not offered by any other SPI RTCC on the market. With long battery life at a premium, applications such as wireless, portable communications, security and automotive can benefit from the 5 MHz SPI bus and millisecond alarm. This high-resolution alarm output provides a greater degree of control over the duty cycle needed to support longer MCU sleep and power-down modes.



<http://www.microchip.com/get/oCX3> - (120532-1)

operating, in response to changes in operating conditions such as variations in rotation speed.

In addition, the advanced signal-processing circuitry in the AS5132 gives it an excellent propagation delay figure of <math><22 \mu\text{s}</math>, which means that even before pre-commutation is implemented the angle error is small.

In combination, the small propagation delay and advanced pre-commutation deliver extremely accurate angle measurement even in motors rotating at speeds up to 80,000 rpm: angle measurements are accurate to within $\pm 3^\circ$ (maximum). This enables BLDC motor manufacturers to achieve high

and constant levels of torque even in high-speed applications over the whole dynamic range.

Other improvements introduced in the AS5132 include:

- increased sensitivity to allow operation in a magnetic field input as weak as 20 mT. This enables the sensor to be paired with a cheap ferrite magnet.
- simplification of the interface for easier integration.
- inclusion of an external clock mode, synchronizing sensor and controller systems.
- implementation of a 3 V interface mode

(in addition to a 5 V mode) eliminating the need for a level shifter.

When paired with a simple two-pole magnet, the AS5132 provides absolute angle measurement at a resolution of 8.5 bits, or 360 positions per revolution. This measurement is available as a serial output and as a pulse width modulated (PWM) signal.

An additional U,V,W output can be used for block commutation in a BLDC motor; alternatively an incremental signal (ABI) is available. In addition to the angle information, the strength of the magnetic field is represented as a 5-bit value.

The zero position can be set by software on the production line. This simplifies assembly, as the magnet does not need to be precisely aligned with a fixed starting position. The AS5132 is suitable for contactless rotary position sensing, rotary switches (human machine interface), AC/DC motor position control and BLDC motor position control in automotive applications such as pump BLDC motors (stop-start system) and steering column BLDC motors. The AS5132 is also particularly well suited to position sensing in double-clutch BLDC motors.

A demonstration board for the AS5132 rotary position sensor is available. Contact ams for the price. For further information on the AS5132 or to request samples, please visit www.ams.com/eng/Products/Magnetic-Encoders/Rotary-Encoders/AS5132

www.ams.com - (120487-X)

Lossless Extended Range Synchronized (LXRS™) Wireless Sensor Networks

MicroStrain's new "lossless" wireless communication protocol (patent pending) supports scalable networks of high speed sensor nodes without data loss due to temporary radio frequency (RF) fading, multipath, or interference.

MicroStrain's users have previously enjoyed a typical RF packet success rate of 99% — not bad. But now, with MicroStrain's new LXRS "lossless" protocol, wireless sensor networks can deliver 100% reliable data throughput under most operating conditions.

The new LXRS™ Wireless Sensing System (microstrain.com/what-is-LXRS?) includes:

Lossless wireless communication protocol

What is LXRS™?

MicroStrain's Lossless Extended Range Synchronized (LXRS™) wireless sensor networks are scalable, fast, and 100% reliable!

Scalable & Fast

of nodes: **15**

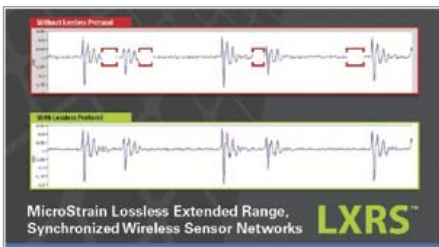
Sample rate: **256Hz**

synchronized

Lossless wireless communication protocol provides 100% packet success rate

Extended range radio link to 2 kilometers

Synchronized to ± 32 microseconds



providing 100% packet success rate under most operating conditions

Extended Range radio link to 2 kilometers
Scalable wireless sensor networks supporting continuous, burst, and hybrid sampling modes

Time Synchronized to ± 32 microseconds
The LXRS™ Wireless Sensing System works by leveraging an advanced bi-directional radio communication protocol. When data are received without errors by the WSDA base station, the WSDA sends an acknowledgement that these packets were received. Data that are not acknowledged remain within each LXRS sensor node's non-volatile memory for scheduled re-transmission without reducing the overall system bandwidth.

Each wireless node in the network has its own precision timekeeper and maintains time synchronization to within ± 32 microseconds. Data are time-stamped by each node at the time of analog-to-digital (A/D) conversion. Therefore, even when re-transmitted, all data are accurately time stamped and aggregated into a master file using time as a unifying variable.

This enables highly reliable data collection from scalable networks of wireless strain, acceleration, torque, force, temperature, etc. to be collected without loss of information. This is particularly useful when working in harsh operating environments, when the system may be subject to periodic RF interference, and when the experiment or test is very difficult to set-up or very expensive to replicate.

Synchronized data collection, combined

with robust and reliable LXRS data acquisition, allows users to select lower sampling rates and reduced RF power levels. This leads to longer battery lifetimes and also enables energy harvesters to be used in place of primary batteries.

www.microstrain.com - (120532-II)

Low-pressure-drop digital flow meter

Swiss sensor manufacturer Sensirion is launching the new digital SFM3000 mass flow meter for high-volume applications in medical devices, process automation and burner control. The flow channel is designed to achieve a very low pressure drop of less than 3mbar through the sensor element. The flow meter bi-directionally measures the flow of air and other non-aggressive gases at rates up to 200 slm with high accuracy and very high speed. It outputs a 14-bit digital signal at a 2 kHz update rate. The signal is internally linearized and temperature compensated. Furthermore, the new mass flow meter operates from a 5 Vdc supply voltage and features a digital 2-wire interface, making it easy to connect directly to a microcontroller.

The outstanding performance of the SFM3000 gas flow sensor is based on Sensirion's patented CMOSens® Technology, which combines the sensor element, signal processing and digital calibration on a tiny microchip. Gas flow is measured by a thermal sensor element to provide an extended dynamic range and enhanced long-term stability compared to other flow measuring technologies. Thanks to the proven CMOSens® technology, the flow meter is suited for high-quality mass production and is the ideal choice for demanding and cost-sensitive OEM applications in medical ventilation and anesthesia, as well as process automation and burner control.

www.sensirion.com/en/sfm3000 - (120532-V)



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

25MHZ SCOPE

Remarkable low cost 25MHz, 2-ch plus trigger USB benchscope with 8-inch full color TFT LCD display. Spectrum analysis and features autoscale functions.

PDS5022ST \$269



60MHZ SCOPE

60MHz 2-ch scope with 500MSa/s rate and huge 10MSa memory!
8" color TFT-LCD & FREE carry case!

SDS6062 \$349

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



The Electronics Maze Solved

In addition to the annual Jumbo edition of Hexadoku, Elektor's July & August 2012 edition also featured a rather different type of puzzle: the *Maze of the Lost Electronics Technician*. Our readers were faced with a maze consisting of passive components: diodes, zener diodes, resistors, coils, capacitors and the odd pushbutton. The challenge was to find the input to which 20 volts had to be applied to make a current flow at the output. Hundreds of readers entered the maze and managed to find the way out successfully. **The correct solution was: input 2.**

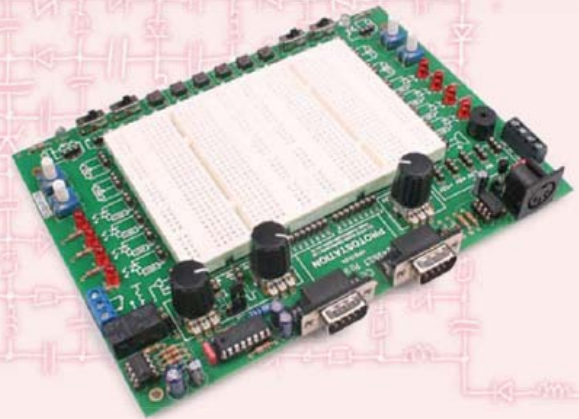
The winner of the
Electronic Workstation – Desktop,
approx. value \$900.00
and a
Protostation Advanced Breadboard,
approx. value \$200.00,

both kindly sponsored by Matrix Multimedia, is:

Evan Wasserman
of Lakewood, New Jersey, USA.

Evan's estimate of the total number of correct solutions received was closest to the actual figure. Congratulations, your prizes are on the way to you.

(120537)



Create complex electronic systems in minutes using Flowcode 5

FLOWCODE5

Design → Simulate → Download



Flowcode 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 experience to create complex electronic systems in minutes. Flowcode's graphical development interface allows users to construct a complete electronic system on-screen, develop a program based on standard flow charts, simulate the system and then produce hex code for PIC AVR, ARM and dsPIC/PIC24 microcontrollers.



NEW!
*Flowcode 5
for AVR*

**Convince yourself.
Demo version, further
information and ordering at
www.elektor.com/flowcode**



PRODUCED BY
CEA

THE GLOBAL STAGE FOR INNOVATION

*Genius attracts.
Feel the pull.*

TUESDAY, JANUARY 8—FRIDAY, JANUARY 11, 2013 * LAS VEGAS, NEVADA * REGISTER AT CESWEB.ORG



REGISTER NOW

USB Isolator

Avoid interference and earth loops

By Raymond Vermeulen (Elektor Labs)

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 here offers an optimal electrical isolation of both the data lines as well as the supply lines between the PC and the USB device.



The project itself is a spin-off from an earlier design for a portable oscilloscope for Android. During the development of that project we also encountered the problems of earth loop noise when a charger or PC was connected. But what was the best point where to introduce isolation, the digital USB connection or the analog section? After some serious analysis we came to the conclusion that it was better to isolate the USB bus rather than the analog part.

A popular chip from Analog Devices, the ADuM3160, was chosen to provide the isolation of the data lines. It supports both Low and Full speed USB connections. A flyback converter with a LT3575 as controller was used to isolate the power supply.

Special IC

The ADuM3160 [1] used in this project, and its bigger brother the ADuM4160, are, as far as we know, the only dedicated USB isolators currently available in the marketplace. These chips can also be found inside most commercially available USB isolators. The chips support Low-speed (1.5 Mbit/s) and Full-speed (12 Mbit/s) USB. According to

Specifications

- Electrically isolated data and power supply lines
- Configurable data rate of 1.5 Mbit/s or 12 Mbit/s
- Power for the device side can be taken from an extra USB host connection or an external AC power adapter (5 V/min. 0.5 A)
- Transformer isolation = 1500 V_{AC} for 1 min
- ADuM3160 isolation = 565 V_{peak} for 50 years
- Maximum output current I_{out} = 500 mA @ 5 V

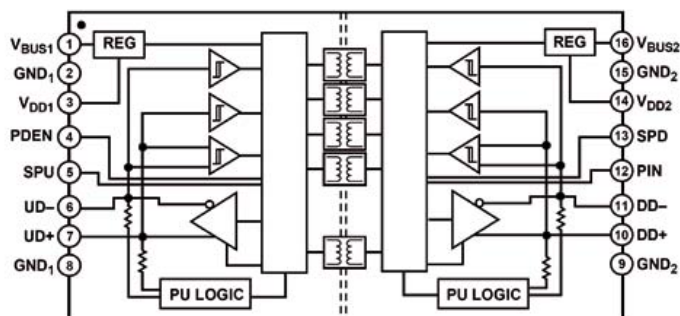


Figure 1. Internal block diagram of the ADuM3160. A number of monolithic transformers provide the electrical isolation between the two sides of the device.

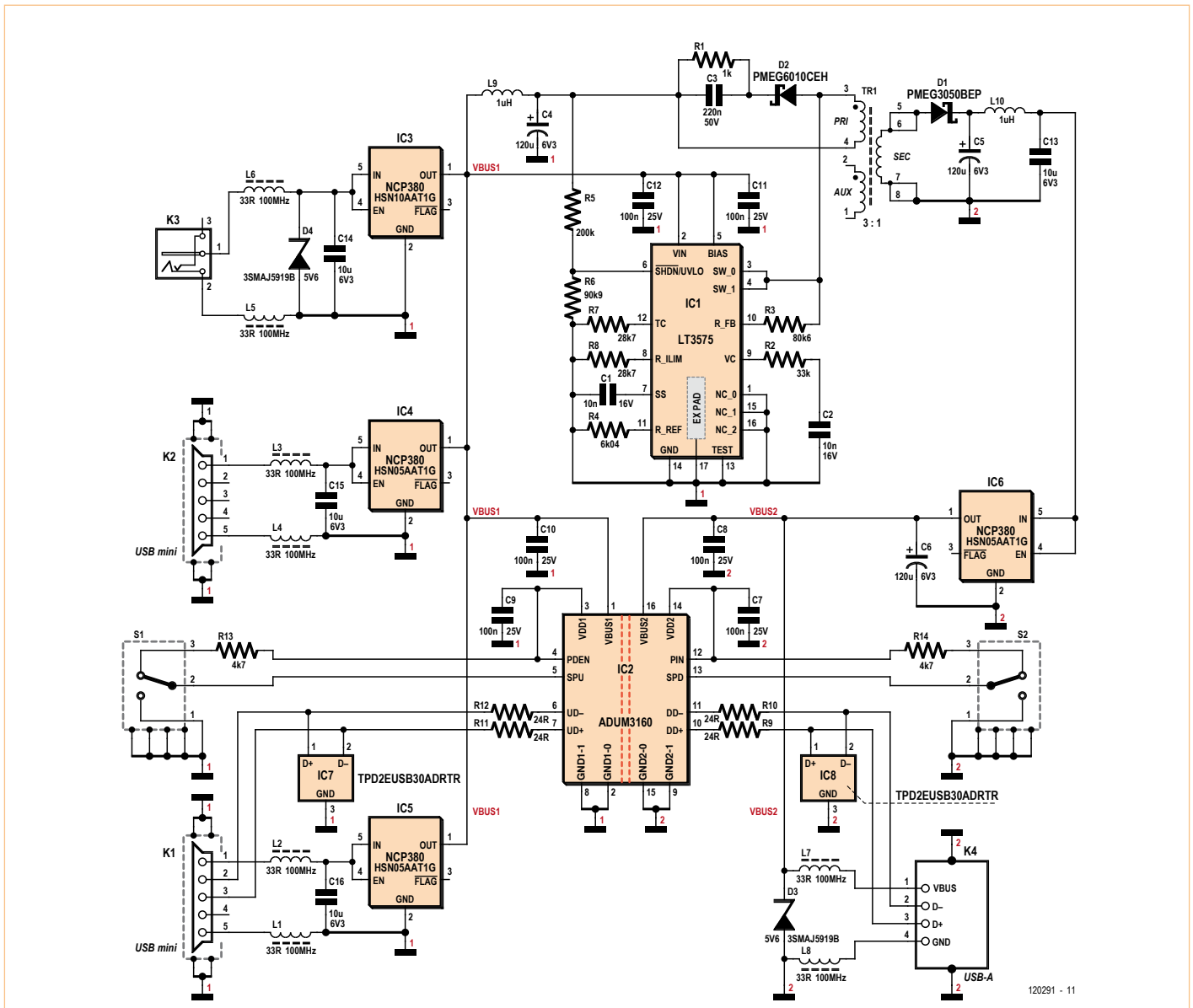


Figure 2. The circuit diagram for the USB isolator. At its heart is a special IC from Analog Devices (IC2), which provides electrical isolation of the USB signals.

Analog Devices the reason why High-speed (480 Mbit/s) USB is not supported is due to the fact that there are currently no digital isolators available that can operate reliably at these high data rates. There are currently no plans to introduce a new IC for this.

In **Figure 1** you can see the internal block diagram for the IC. The combination of

high-speed CMOS technology with air core transformers (the iCoupler technology from AD) results in an outstanding isolation between the primary and secondary sides. To detect the direction of the data flow over the USB lines (in order to turn on the appropriate output buffers) the IC determines the direction based on the contents of received data packets. The propagation

delay (the time taken for the received signal to appear at the output) is fairly small despite the stringent isolation requirements, and is comparable to that found in a standard USB hub.

Circuit diagram and power supply
You may wonder why the power supply is specifically mentioned in the heading for

COMPONENT LIST

Resistors

(1% 0.1W, SMD 0603, unless otherwise shown)
 R1 = 1kΩ 1% 0.5W (SMD 1206)
 R2 = 33kΩ
 R3 = 80.6kΩ
 R4 = 6.04kΩ
 R5 = 200kΩ
 R6 = 90.9kΩ
 R7,R8 = 28.7kΩ
 R9,R10,R11,R12 = 24Ω
 R13,R14 = 4.7kΩ

Capacitors

C1,C2 = 10nF 16V X7R (SMD 0603)
 C3 = 220nF 50V X5R (SMD 0603)
 C4,C5,C6 = 120μF 6.3V (Nichicon RFS-0J121MCN1G5, type C)
 C7,C8,C9,C10,C11,C12 = 100nF 25V X7R (SMD 0603)
 C13,C14,C15,C16 = 10μF 6.3V X5R (SMD 0805)

Inductors

L1,L2,L3,L4,L5,L6,L7,L8 = ferrite inductor 33Ω@100MHz (SMD 0603) e.g. Murata BLM18PG330SN1D
 L9,L10 = 1μH 1.2A (SMD 1007) e.g. Taiyo Yuden CB2518T1R0M

Semiconductors

D1 = PMEG3050BEP (NXP) (SOD-128)
 D2 = PMEG6010CEH (NXP) (SOD-123f)
 D3,D4 = 5.6V 3W zener diode
 IC1 = LT3575EFE#PBF (LT) (TTSOP-16)
 IC2 = ADUM3160BRWZ (AD) (SOICW-16)
 IC3 = NCP380HSN10AAT1G (On Semiconductor) (TSOP-5)
 IC4,IC5,IC6 = NCP380HSN05AAT1G (On Semiconductor) (TSOP-5)
 IC7,IC8 = TPD2EUSB30ADRTR (TI) (SOT-3)

Miscellaneous

S1,S2 = slide switch with changeover contact (e.g. C&K Components PCM12SMTR)

K1,K2 = USB mini-B-connector, PCB mount (e.g. On-Shore USB-M26FTR)
 K3 = DC adapter connector, 1.35 mm (e.g. CUI PJ-014DH-SMT)
 K4 = USB-A connector, PCB mount (e.g. FCI 87583-2010BLF)
 TR1 = transformer 3:1, 25 μH (e.g. Würth 750310471)
 Case, Hammond type 1593KBK
 2 self tapping screws, cross head, 4x6.4mm, DIN7981
 PCB # 120291-1 or assembled and tested board and case, # 120291-91

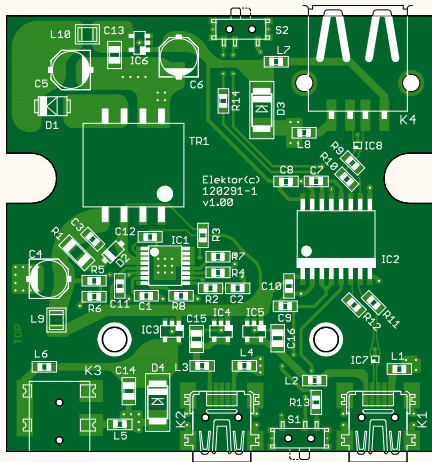


Figure 3. The dimensions of the board were chosen to fit in a type 1593KBK case from Hammond.



this section. It's because it plays such a significant role in this design. The ADuM3160 can only provide isolation for the data lines and not for the supply line. You could provide each side of the IC with its own, separate power supply.

In this case (see **Figure 2**) we decided to add an isolated switching power supply to the board. This not only provides power for the other half of the IC, but also provides the maximum permitted current for the

USB device connected to it (which is also completely isolated from the input!). To obtain the required 500 mA at 5 V from the output we have chosen a LT3575 fly-back converter from Linear Technology [2]. Remarkably this IC doesn't require an optocoupler for a feedback signal. The converter operates in a so-called boundary mode, where the information regarding the output voltage of the secondary inductor is fed back to the primary inductor. This works as follows:

When the power switch in the converter turns off, the voltage rises to a level that is equal to:

$$V_{FLBK} = (V_{OUT} + V_F + I_{SEC} \times ESR) \times N_{ps}$$

where

- V_F = voltage across diode D1;
- I_{SEC} = current through the secondary winding;
- ESR = total impedance of the secondary side;

- N_{PS} = turns ratio of the primary and secondary windings of the transformer.

In conjunction with the temperature compensation this provides enough information to regulate the output voltage. One consequence of this method is that the switch has to be turned off for a certain minimum amount of time during each cycle, to give the IC sufficient time to process the feedback information. An off the shelf transformer from Würth was chosen, which is a small potted type for SMD mounting. The data lines are protected from high input voltages by IC7 and IC8. These are two-channel ESD protection ICs (TPD2EUSB30A) from TI (this part would also be suitable for USB3 data lines). The dimensions of this IC are so small (1 x 1 mm!) that it can easily be put between the data lines on the PCB. All power supply connections are protected against overloading with the help of high-side power distribution switches (NCP380) from ON Semiconductor. These ICs include a detection and control circuit with a P-channel MOSFET that limits the output current when it exceeds a preset value. Two different types of the NCP380 have been used in this circuit: IC4, 5 and 6 limit the current to 0.5 A, IC3 to 1 A. These provide optimum protection to all inputs and outputs (both USB and supply) against (almost) all possible types of overload. In this way, the USB ports are limited to 500 mA and the 5 V_{DC} connection is limited to 1000 mA. A connected USB device always has 500 mA available to it, which doesn't quite conform to the official USB standards. Initially, the limit should really be 100 mA. It should be increased to 500 mA only after the device has announced itself to the host and requested the current increase. However, in practice it seems that you can normally get 500 mA from a USB port on your PC without the need for any requests. In order to keep ripple to a minimum at the primary and secondary side of the switching power supply, we've added LC filters to both sides (L9/C4 and L10/C13). The ripple currents through capacitors C4 and C5 are

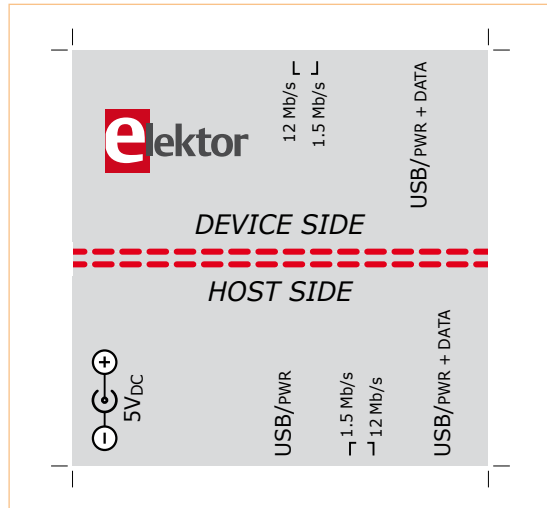


Figure 4. Print this out and stick it to the top of the case for the professional look.

very large, which you have to keep in mind when you select the types for these. It is best to use capacitors with the lowest possible ESR. To keep the turn-off spike at the primary side of the supply within manageable levels, an RCD snubber network has been added, consisting of R1, C3 and D2.

Construction

For a circuit that has so many special ICs (all SMD) it is a 'must' to have a thoughtfully designed board layout. In **Figure 3** you can see the board that was designed in the Elektor Labs. It is of course possible to obtain this board separately and populate it yourself. However, since several components are very tricky to solder due to their size, and some are also not easily obtainable, Elektor has made a fully populated board available, including the case. All you need to do is to add a few holes for the connectors and slide switches in the two side panels of the case. To be on the safe side, the two tags of K4 that come through the board should be soldered using an ordinary soldering iron. The board can then be fixed inside the case using the two screws provided.

The drawing in **Figure 4** can be copied and stuck to the top of the case. The original design for the drawing can also be down-

loaded from the Elektor website [3] accompanying this project.

Usage

It is very easy to use the USB isolator. First, set the required speed with switches S1 and S2.

Both of these should always be set to the same speed!

Normally a USB device can let the host know at which speed it operates via a pull-up resistor on one of its data lines. This isn't possible with this isolator and the speed has to be set manually via switches (this is also the case for most commercially available USB isolators).

Connect the 'host' (in most cases this is your PC) to K1 (HOST SIDE: USB/PWR+DATA) and the isolated USB device to K4 (DEVICE SIDE: USB/PWR+DATA). If the device needs more than 300 mA we suggest that you connect an extra USB cable to K2 (USB/PWR) or connect a regulated 5 V power adapter (min. 500 mA) to power supply connector K3 (5 V_{DC}).

We recommend that you don't use this project in situations where there may be voltage differences between the host and device side that are greater than the AC line voltage. You should also not subject the device to high voltage differences between the host and device side for extended periods of time, because the isolation can deteriorate over time if the components are continuously exposed to high voltage differentials.

(120291)

Internet Links

- [1] www.analog.com/static/imported-files/data_sheets/ADuM3160.pdf
- [2] <http://cds.linear.com/docs/Datasheet/3575f.pdf>
- [3] www.elektor.com/120291

Embedded Linux Made Easy (4)

A look at the kernel

By Benedikt Sauter [1]

GNU/Linux provides interfaces and applications for devices that we are used to seeing on desktop computers and servers. Much of this transfers to the embedded domain, where Linux relieves the programmer of a lot of work by handling networking, USB, Bluetooth and more without the need to write complex dedicated C programs. And as we shall see, using a modern operating system brings many other benefits too.

In the previous installments in the series we have looked at how the tool chain, the kernel and bootloader, and a standard file system can be got up and running. One aim of the series is to show developers what is happening 'under the bonnet', for example in how devices and processes are implemented. In this installment we will first look at what advantages there are in using an operating system when compared to driving hardware at the 'bare metal' level.

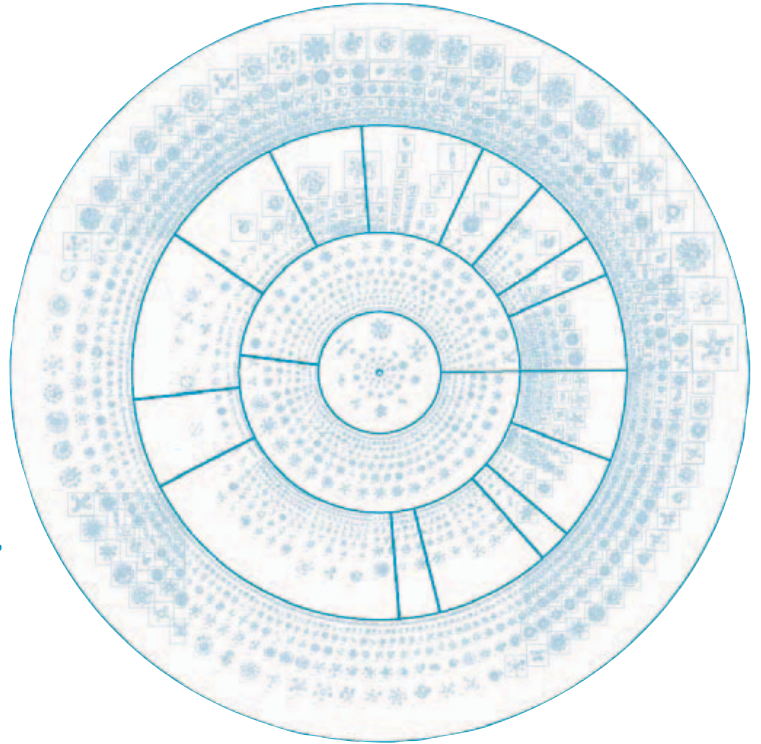
What is the essence of an operating system? At its heart it consists of a collection of all the software needed to run specially-written applications in parallel with others. The distinction between 'normal' firmware and operating-system-based firmware is, however, somewhat blurred. It is common enough for normal firmware to include a clean interface to the hardware layer to provide commonly-wanted functions, for example in memory management; and often there will be a small scheduler to divide the processor's attention between different programs. Together these already form the beginnings of what might be called an 'operating system'.

A typical operating system

There are of course many books on the subject of operating systems [2]. Here we shall only take a broad overview of the main components:

- Device management
- Process management
- Memory management
- File management
- User and permissions management

In the embedded world device management is usually a particularly



important aspect. The operating system must present peripherals and hardware to applications in such a way as to simplify their use as far as possible. This is done using device drivers. The operating system is also responsible for handling all the details behind allowing several programs to run simultaneously, all wanting access to SD cards, network interfaces and so on.

In the simplest programming model, an application running under an operating system is a single, independently executable process. A process has its own virtual address space, ensuring that it cannot disturb other processes. Processes are centrally managed via the process table, and each has its own 'process control block' [3] which typically includes space for temporary storage of processor register contents and other important variables:

- Program counter
- CPU registers
- Stack pointer
- Status of open files
- Process state
- Other status information

When we launch our 'hello world' program the kernel loads the application and creates a new process. A new process is always created on the basis of the current one, and inherits the user permissions associated with it. The command

```
ps -ejH
```

will display the current set of processes in the system in a tree structure. At the root of the tree is the first process launched by the ker-

nel, invariably called '/sbin/init'. Starting this process is one of the first things the kernel does when booted.

As developers we are particularly interested in the following features of an operating system:

- The interface for device drivers
- The file system for programs and data
- Launching and executing programs
- Shared use of libraries by different programs
- Access to input and output ports
- Transferring working memory to non-volatile memory
- Memory management for applications and processes
- Process management

Let us start with device management.

Configuring the kernel

Which interfaces and peripheral blocks of the embedded processor we need will depend on the application. The Linux kernel is designed in a modular fashion, which results in the developer having very fine control over its settings and in particular over what drivers and mechanisms will be included in the kernel. It is also subsequently possible to load other components dynamically. The best way to see how this is done is to look directly in the kernel source code. So, switch to the kernel source code tree:

```
cd ElektorLinuxBoardDownload_20120509
```

```
cd linux-2.6.33-lpc3131x
```

and invoke the kernel configuration menu:

```
make menuconfig
```

The main menu (**Figure 1**) should immediately appear. If it fails to appear, one possible cause is that the libraries for displaying console-style menus are missing. They can be installed on our Ubuntu system by typing:

```
sudo apt-get install libncurses5-dev
```

From the main kernel configuration menu it is possible to enable or disable any of its features and device drivers at will. For the beginner it will not always be immediately obvious where within the menu structure a particular option might be hiding, but with a bit of practice it soon becomes familiar.

An alternative to using the menu is to go straight to the configuration file '.config' in the kernel source directory (linux-2.6.33-lpc3131x) and edit it by hand. (The full stop at the start of '.config' indicates that it is normally a 'hidden file'.) However, it is usually more convenient to use the interactive menu. Before changing the configuration it is a good idea to make a backup copy of the current configuration file.

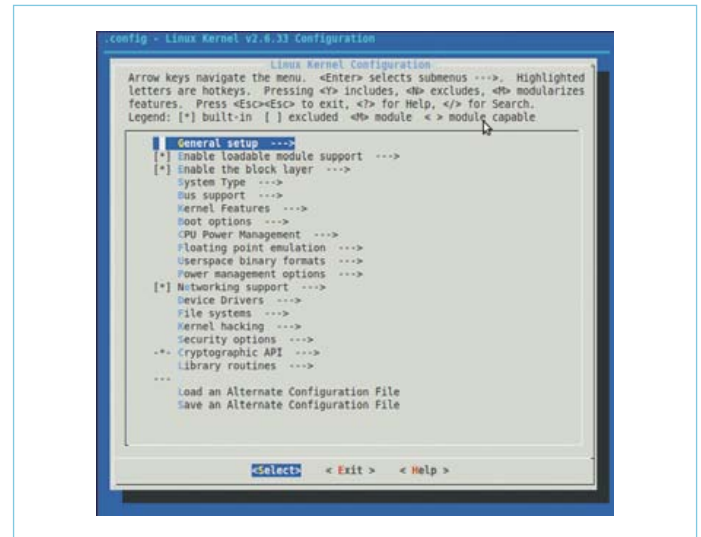


Figure 1. Main kernel configuration menu.

```
cp .config config_backup
```

Newcomers to Linux often ask where good documentation can be found. The best documentation for the kernel is in the kernel source itself. It is also possible to click on 'help' at any time in the menu to find information on a particular item, or you can browse through the 'Documentation' directory (at the top level of the kernel source tree) to look for information on a particular topic.

As Figure 1 shows, the menu items are sorted into groups. At the top are the general kernel settings, and then follow settings for the device drivers and file systems, and finally items relating to cryptography and security.

General setup

The menu item 'General setup' covers the basic characteristics of the kernel, including, for example, whether there is any swap space, how memory is managed and how internal communications are carried out. There are also settings relating to the kernel image itself, optimizations for small systems, statistics, tools and much more besides. The more you find out about the Linux kernel, the more handy features you discover: there are only a few people who can lay claim to understanding it completely!

Enable loadable module support

Another good feature of Linux is that the kernel can be extended, while it is running, by simply adding extra modules. The kernel configuration menu lets you decide whether a particular feature or driver should be permanently built into the kernel code or compiled as a module: an 'M' before a menu item means that the feature will be compiled as a module, whereas an asterisk before an item means that it will be built in permanently. If you want to be able

to support a wide range of different devices, building all the drivers into the kernel can make it unwieldy: it is better to compile the drivers as modules and load them manually or let the kernel load them automatically.

Enable the block layer

The kernel distinguishes between two types of device: character devices and block devices. A character device sends and receives messages and other data one character at a time; a block device always does so in complete blocks. Typical character devices include consoles and UART interfaces; block devices include hard disks and memory cards such as MMC and SD cards. If no block devices are needed (as is possible for some embedded systems) some memory space can be saved by disabling the 'block layer'.

System type

Some kernel features depend on the type of processor being used. For example, it may be desirable to enable or disable caches or memory management units. This menu item lets you specify (as far as is possible) the processor type: if your particular processor is not listed, you will need to find a patch or suitable 'board support package' (or write it yourself).

Bus support

Here it is possible to enable support for typical PC-type buses. In our case only the PCMCIA option is available.

Kernel features

This item is very kernel-specific: you can specify what binary interface convention is to be presented to programs and other miscellaneous features, such as displaying processor load via the blink frequency of an attached LED.

Boot options

How the kernel is started up depends in the case of the *Elektor* Linux board chiefly on the bootloader. The main interface between the bootloader and the kernel is the so-called 'kernel command line'. This command allows the bootloader to pass various parameters to the kernel that can subsequently be examined by it or by other programs. The menu item allows you to specify that a given different command line should be used instead. Alternatively, it is possible to compile the kernel so that it can be run directly from flash storage.

CPU Power Management

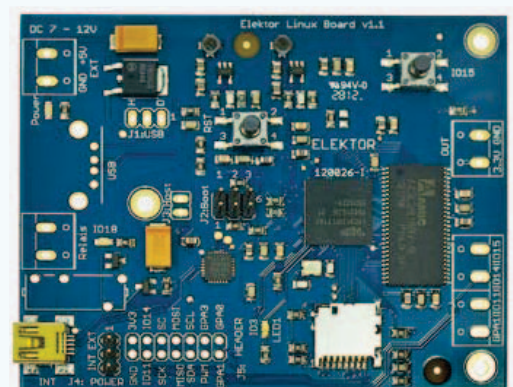
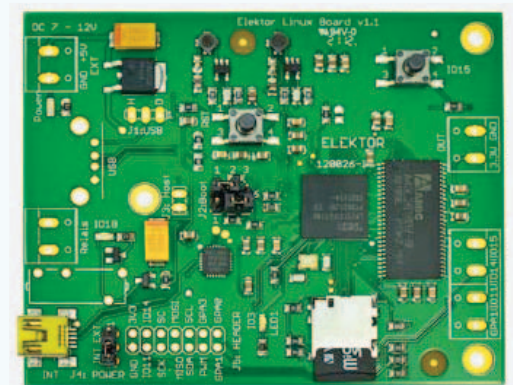
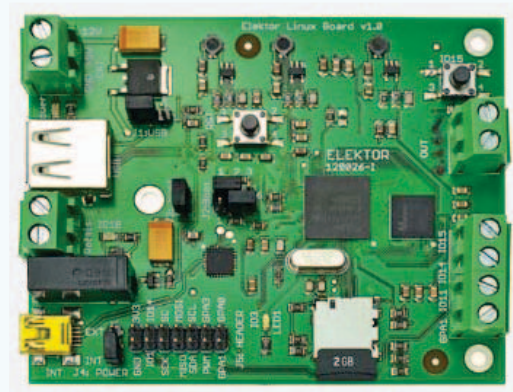
These days processors offer an ever wider range of features to support efficient power management. The kernel has to be properly configured to allow applications to take full advantage of these features.

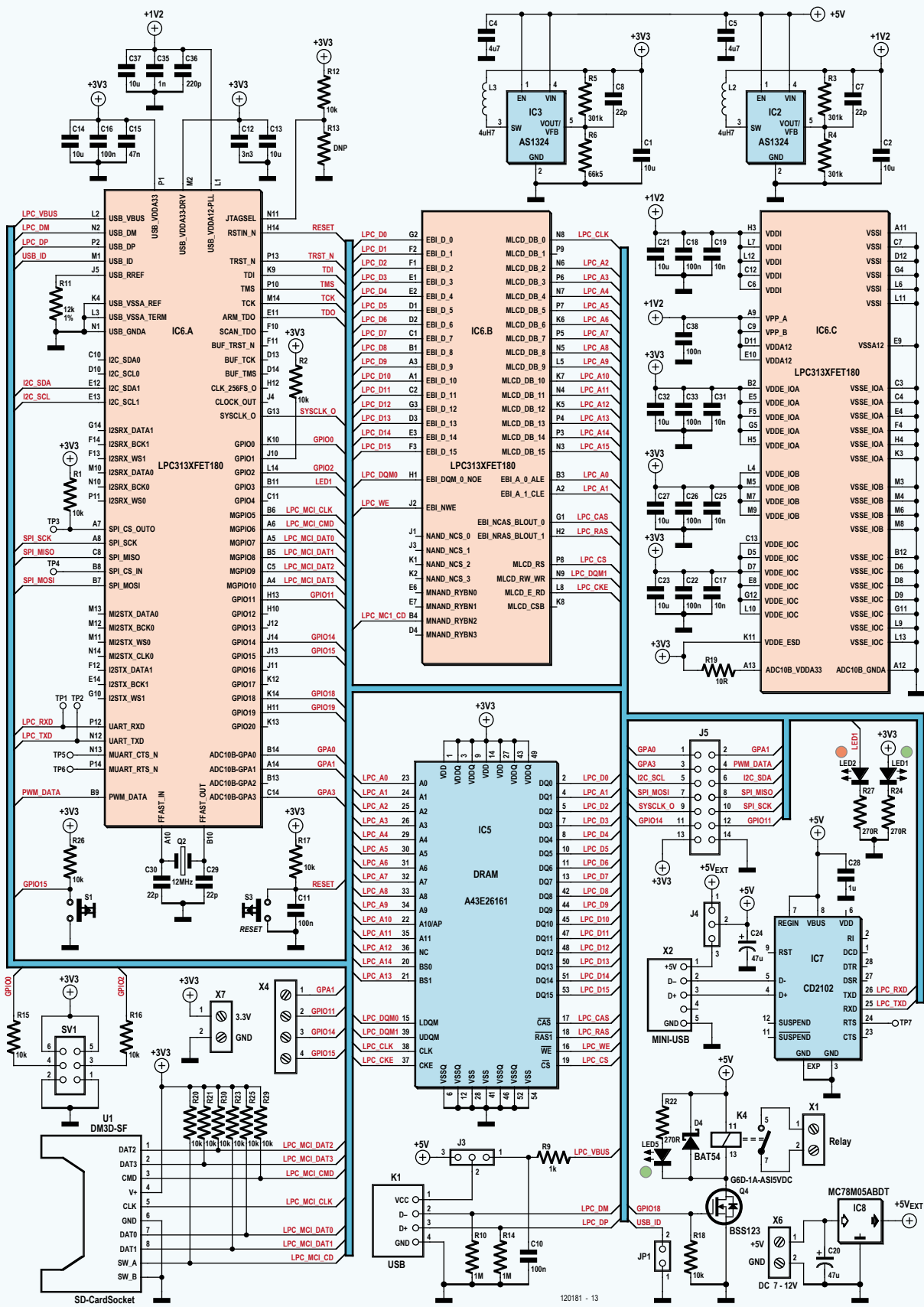
Floating point emulation

This kernel configuration menu option allows you to configure it so that it takes responsibility for floating-point calculations or so that hardware support is used. At least one of these alternatives must be selected to allow programs to be run.

New version of the *Elektor* Linux board

Demand for the *Elektor* Linux board has been truly overwhelming! Because of the large number of orders we have received the next production batch has had to be made rather sooner than expected. Unfortunately the DRAM chip in the BGA package has a lead time of some 20 weeks, and so we decided to change the layout to accept a TSOP54 device. The new memory has the same capacity as the old one but requires a supply voltage of 3.3 V instead of 1.8 V. The upper picture shows the original version of the board with the BGA memory chip, while the version in the middle picture has the TSOP54 device fitted. Since the beginning of July we have been delivering version '1.1 blue' of the board, in which the now redundant 1.8 V voltage regulator circuitry is removed. You can easily identify the most recent version of the *Elektor* Linux board by its blue colour. The corresponding circuit diagram is shown here.





120181 - 13

User space binary formats

Programs and applications are compiled using the tool chain (see the previous installment in this series). Normally, for a Linux system, the tool chain will produce either a .elf file ('executable and linkable format', corresponding roughly to a .exe under Windows) or an 'a.out file'. The kernel reads these formats and launches the compiled application appropriately.

A handy Linux command is `file` which lets you see what format a given file has. Switch to the appropriate directory using `cd` and then type `file` followed by the name of the file of interest. For example, in the Linux directory there is a file called 'vmlinux'. Typing

```
file vmlinux
```

produces the answer

```
vmlinux: ELF 32-bit LSB executable, ARM, version 1 (SYSV), statically linked, not stripped
```

The file is an executable file for ARM processors that is statically linked.

Power management options

It is not just the processor that offers power management features: many peripherals and other devices have the ability to go into a sleep mode. The operating system can take a copy of important data and send the peripheral to sleep. When the time comes to wake the peripheral up again, the data are copied back from memory and its state is restored. The exact behavior of the kernel in these situations can be configured here.

Networking support

Finally we come to one of the most interesting points for developers. Under 'networking support' we can enable, disable and configure all the software stacks (protocol libraries) for the various communication interfaces, including Ethernet, TCP/IP, CAN, infrared, amateur radio, Bluetooth and wireless.

Device Drivers

We are also especially interested in device drivers. This menu item includes everything you might wish for, from the simplest I/O port to the most up-to-the-minute high speed server bus. The kernel source code (see download at [4]) includes drivers for a wide range of USB, I²C and SPI devices which we now have at our disposal.

File systems

File systems are important to us for two reasons. First, the operating system itself must reside in a file system (on the SD card), so that the kernel can read the files it needs from it. And second, the operating system has to use a file system to allow application files and directories to be stored on a hard disk.

Kernel hacking

This menu item is chiefly aimed at kernel developers.

Security options

Linux includes a range of different security features. The most familiar of these is the system of user and group permissions and the existence of a 'root' user. Applications can be further protected using techniques such as trusted platform modules and internal security models.

Cryptographic API

This menu item covers cryptographic functions in the kernel.

Library routines

This menu item includes functions for calculating checksums alongside many other things.

If you want to save your new kernel configuration, use the last menu item 'Save an alternate configuration file'. After entering a name for the new kernel configuration it will be stored locally in the kernel source tree.

We described in the previous installment in this series [4] how to go about compiling the kernel (`make zImage`) and copying it to an SD card. If new modules are to be included in the kernel, they have to be separately compiled and installed. Compile the modules by typing:

```
make modules
```

and to copy them to the SD card, either give the path to the destination SD card directly in the following command, or install the modules into '/tmp' and then copy the new directory that is created there to the SD card, either as user 'root' or using the 'sudo' command. (It is not that root privileges are required for the copy operation itself, but rather that we must ensure that the files are copied to the file system on the SD card in such a way that the kernel can subsequently use them as the root user.)

```
make INSTALL_MOD_PATH=/tmp modules_install
```

Now we switch to the '/tmp' directory and copy the directory lib to the SD card:

```
cd /tmp
sudo cp -R lib /media/86b3be7-00f3-45e0-832e-1f48c2c3065e
```

with the appropriate change to the long hex string.

Interfaces

Linux has its roots in the PC world. There we are faced with interfaces such as IDE, SATA, PCI, AGP and the like. In the embedded world the focus is more on the traditional microcontroller interfaces and, of course, USB.

To summarize, our board has the following interfaces:

Table 1. Device files typically found in the /dev directory

Device type	File in /dev	Character or block device
IDE hard disk	/dev/hda, /dev/hdb, ...	Block
SCSI hard disk	/dev/sda, /dev/sdb, ...	Block
RS-232 interface	/dev/ttyS0, /dev/ttyS1, ...	Character
Printer	/dev/lp0, /dev/lp1	Character
Camera	/dev/video0, /dev/video1	Block
Mouse	/dev/input/mice	Character
RAM	/dev/ram	Block

- USB (as host and device)
- UART (in its current form as the system console, but can also be used as an interface to external devices)
- I²C (as master)
- SPI (as master)

Also, the processor has other hardware functions such as:

- GPIO
- A/D converter
- PWM output

As we proceed we will look more closely at these interfaces.

Device files

In order to allow applications to talk to devices, we need an interface to the device driver. The way that drivers are accessed varies greatly from one operating system to another. Linux is based on a UNIX file system, and under UNIX things are particularly simple: each device is treated as a file. The idea is that no special tools are required to drive a device; instead, we can just use small, simple programs (from the console), operating on devices exactly as if they were files. We showed an example of how this works in the case of GPIO pins in the second installment of this series [5], where we demonstrated how to flash an LED.

There are two important commands with which surprisingly much can be achieved: `cat` and `echo`. The command `cat` displays the contents of a file; `echo`, conversely, lets us write any desired text to a file. To put it more precisely, the output of `echo` can be redirected to any desired file.

So, for example, if we want to write some data directly to a hard disk (that is, not via a file system) or send data to a serial port, we can use `echo` in commands of the form

```
echo "Data for the hard disk" > /dev/sda
```

or

```
echo "Hello there, world" > /dev/ttyS0
```

But be careful: if you wish to experiment with writing data directly

to a hard disk or memory card, make sure you use one without any important information stored on it!

Some device files correspond to block devices and some to character devices. A hard disk, for example, is always treated as a block device, whereas an RS-232 interface is treated as a character device like a normal terminal, receiving and processing characters one at a time. **Table 1** gives a list of example device files that might be found in a typical UNIX or Linux system.

Readers may be wondering exactly what mechanism is hiding behind the commands `echo` and `cat`. How exactly does a driver send or receive data? Now, since everything is a file, we can access the device by simply reading from or writing to it as we would a file. This means that each driver simply has to provide three functions: `Read`, `Write` and `Ioctl` (which allows the device's settings to be changed). `Read` and `Write` are the standard functions for reading and writing files, the process in the case of a device being exactly analogous to that for a normal file. The first step is to open the device or file and obtain a 'handle' (a number which identifies the opened device or file). The handle is then passed to subsequent `Read` or `Write` operations on the device or file.

Kernel space and user space

Classical operating systems (in which category we include Linux) are generally constructed with a clear separation between the memory area that is used by the kernel and the memory area available for applications. An application is not permitted to access the kernel's memory. This is another important fact for developers working close to the system level to bear in mind. In Linux, the memory areas are called 'kernel space' and 'user space'. Usually in the case of a device driver, all the software will be in kernel space. So how does a user application program get its hands on the data received over an interface?

The answer lies in the kernel functions such as `copy_to_user` [6], which copies a block of data from kernel space into user space. So, when an application reads a device file the driver must use this function to make the information available to the application.

Integrating a device driver

We now know the basic principles of operation of a kernel device driver. The question remains, however, of how the kernel knows to associate a file such as `/dev/ttyS0` with the correct driver for the serial port.

```

root@gnublin:~# cat /proc/devices
Character devices:
1 mem
4 /dev/vc/0
4 tty
4 ttys
5 /dev/tty
5 /dev/console
5 /dev/ptmx
7 vcs
10 misc
13 input
21 sg
89 i2c
128 ptm
136 pts
153 spi
180 usb
189 usb_device
252 usbmon
253 lpc313x_pwm
254 lpc313x_adc

Block devices:
1 ramdisk
259 blkext
7 loop
8 sd
65 sd
66 sd
67 sd
68 sd
69 sd
70 sd
71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
179 mmc
    
```

Figure 2. List of available drivers.

To this end so-called ‘major’ and ‘minor’ device numbers are used. These are unique identifying numbers that are defined in the kernel documentation. The numbers are also present inside the drivers for each type of device. The following command can be used to display which drivers with which major numbers are currently available:

```

root@gnublin:~# cd /dev
root@gnublin:/dev# ls -l
total 1
crw----- 1 root root    5, 1 Sep 27 06:02 console
prw----- 1 root root    0 Sep 27 06:02 initctl
srw-rw-rw- 1 root root    0 Sep 27 06:02 log
crw-r--r-- 1 root root  254, 0 Sep 26 03:57 lpc313x_adc
crw-r--r-- 1 root root  253, 0 Sep 26 04:21 lpc313x_pwm
lrwxrwxrwx 1 root root   12 Jan 1 1970 mtab -> /proc/ mounts
crw-r--r-- 1 root root    1, 3 Dec 22 2011 null
crw-r--r-- 1 root root   89, 1 Sep 26 03:06 pca9555
crw-rw-rw- 1 root root    5, 2 Jan 1 1970 ptmx
drwxr-xr-x 2 root root    0 Jan 1 1970 pts
crw-r--r-- 1 root root    1, 9 Sep 26 04:28 random
brw-r--r-- 1 root root    8, 0 Sep 27 05:52 sda
brw-r--r-- 1 root root    8, 1 Sep 27 05:52 sda1
crw-rw-rw- 1 root root    5, 0 Jan 1 1970 tty
-rw-r--r-- 1 root root   24 Jan 1 1970 tty0
crw----- 1 root root    4, 64 Sep 27 06:03 ttyS0
crw-r--r-- 1 root root    1, 9 Sep 26 04:30 urandom
    
```

Figure 3. These files represent our devices.

cat /proc/devices

The result is a list of device drivers, sorted into character devices and block devices (Figure 2).

Now we can look in the ‘/dev’ directory using `ls -l` to display its contents (Figure 3). There will be a large number of files in the directory representing our various devices. Let us look at the information provided alongside each device. At the beginning is the letter ‘c’ or ‘b’ which indicates whether this is a character device or a block device. Then there are the permissions as usual: we will look at these in more detail in a later article in this series. Then the user who is the ‘owner’ of the file. Shortly after that there are two numbers separated by a comma, and these are the major and minor device numbers. In this example the major numbers are 1, 4, 5, 8, 89, 253 and 254; the minor numbers are 0, 1, 2, 3, 9, 12, 24 and 64.

If we access one of these device files from user space, the operating system will be able to determine from these numbers and from the entries in ‘/proc/devices’ exactly which driver is needed.

It is easy to create device files using the tool ‘mknod’ [7]. We will see an example of this below.

Integrating a USB-to-serial converter

It’s now time to turn theory into practice. Let us drive a simple USB-to-serial converter from our Linux board (from the USB end!): see Figure 4. The basic steps in the process are as follows.

- Enable the driver in the kernel.
- Compile the kernel.
- Copy the new kernel to the SD card.
- Check the driver is available in the new kernel.
- Create a device file.
- Test the device.

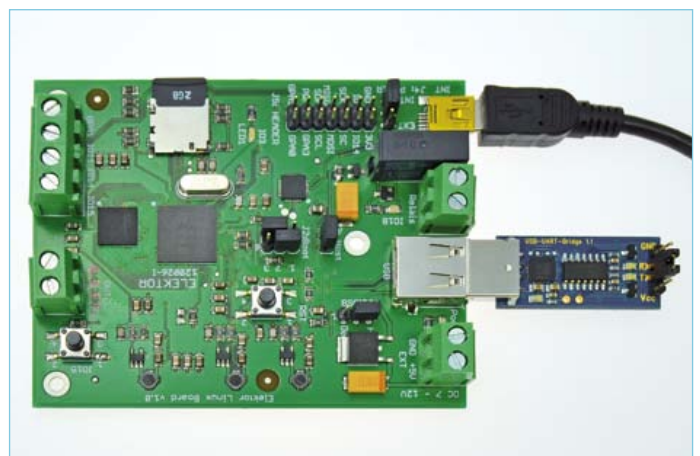


Figure 4. A CP2102-based USB-to-serial converter connected to the Elektor Linux board.

Enable the driver in the kernel

First switch to the Linux source tree:

```
cd ElektorLinuxBoardDownload_20120509/
```

```
cd linux-2.6.33-lpc313x/
```

```
./set.h
```

```
make menuconfig
```

In the first step (**Figure 5**) select the menu item 'Device drivers'. Then select 'USB support' (**Figure 6**). With the default configuration there will be an 'M' here instead of the '*' shown in the screenshot. Use the space bar to cycle through the three options 'M' (compile as module), '*' (compile into kernel), or neither. We select '*' to avoid the complexity of having to hunt down the module later. The next menu item is 'USB serial converter support' and here also we must turn the 'M' into a '*' (see **Figure 7** and **Figure 8**). Pressing the Enter key brings up a submenu to select the right driver (**Figure 9**). Most readers will probably want to use the FTDI devices or the Silicon Labs CP210x controllers, and so we set both of these drivers to '*'. To apply these changes, go repeatedly to 'Exit' and then, on the last page, confirm that you want to save a new configuration file.

Compiling the kernel

We can now set off the kernel compilation process.

```
make zImage
```

Typical resulting output is shown in **Figure 10**.

Copy the kernel to the SD card

Put the SD card from the board into an SD card reader and then connect the reader to the PC. Now copy the kernel to the SD card as follows

```
sudo cp arch/arm/boot/zImage /
media/386b3be7-00f3-45e0-832e-1f48c2c3065e/
```

with, as before, the appropriate change to the long hex string. When the copy operation is complete unmount the file system manually in order to ensure that all the blocks are safely written to the SD card [4]:

```
umount /media/386b3be7-00f3-45e0-832e-1f48c2c3065e
```

Check that the driver is available

Now we can boot up the *Elektor* Linux board as usual. It is important to check that JP1 is fitted, so that we force the USB interface to start up in host mode. We must also make sure that the device is provided with power over the USB connector from the Linux board. This is done by fitting J3 in position 3-2.

We can see the first differences during the boot process itself. The

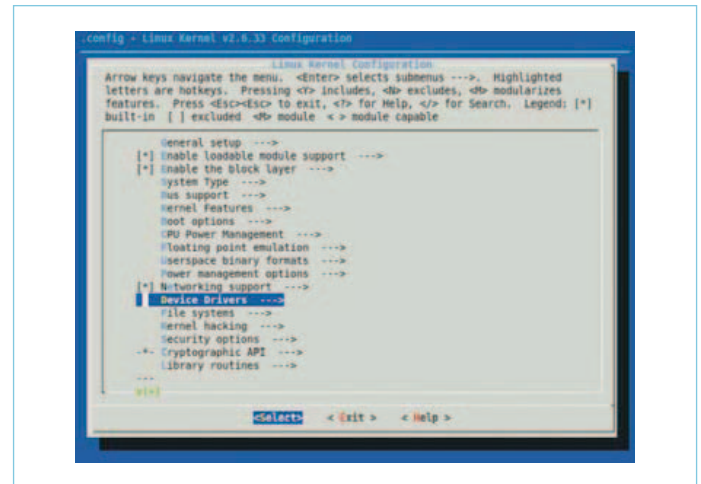


Figure 5. The 'Device drivers' menu item.

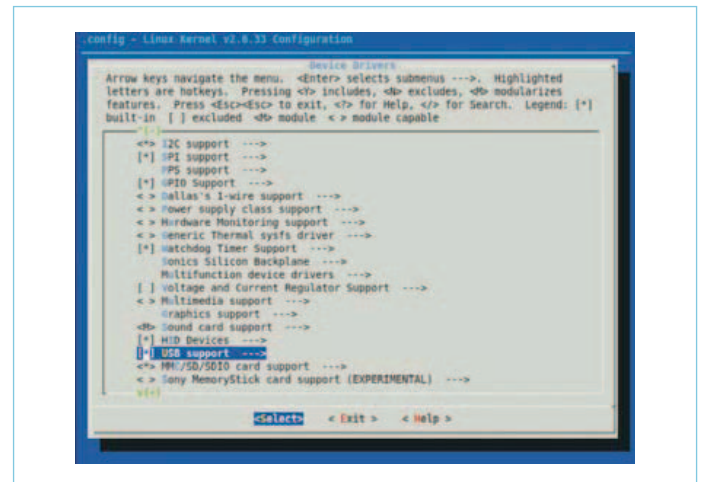


Figure 6. The 'USB support' menu item.

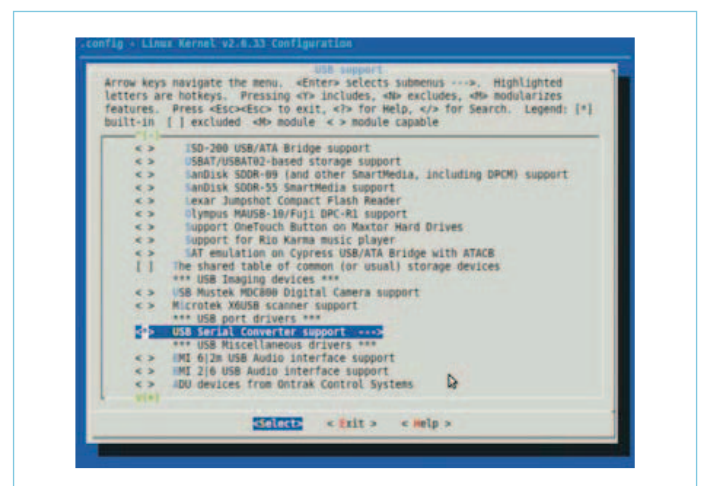


Figure 7. Selecting 'USB serial converter support' to be built in to the kernel.

MICROCONTROLLERS

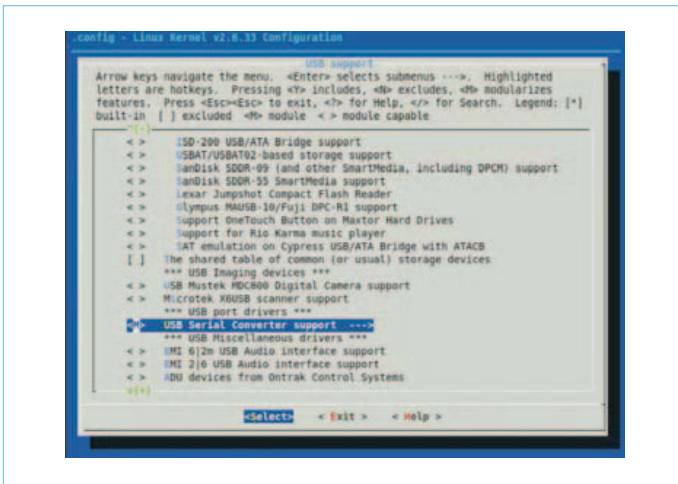


Figure 8. Selecting 'USB serial converter support' to be built as a dynamically-loadable module.



Figure 9. Here we select the USB-to-serial adapter(s) that will be supported.

two new drivers make their appearance as follows.

```
cp210x: v0.09:Silicon Labs CP210x RS232 serial
adaptor driver
```

```
USB Serial support registered for FTDI USB Serial
Device
```

Creating a device file

The command `cat /proc/devices` reveals a new entry '188 ttyUSB'. We create a device file with this major number as follows.

```
mknod /dev/ttyUSB0 c 188 0
```

Testing the device

If we now plug the USB-to-serial adaptor into the board we get the following output.

```
usb 1-1: New USB device found, idVendor=10c4,
idProduct=ea60
```

```
usb 1-1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
```

```
usb 1-1: Product: CP2102 USB to UART Bridge
Controller
```

```
usb 1-1: Manufacturer: Silicon Labs
```

```
usb 1-1: SerialNumber: 0001
```

```
cp210x 1-1:1.0: cp210x converter detected
```

```
usb 1-1: reset full speed USB device using lpc-ehci
and address 2
```

```
usb 1-1: cp210x converter now attached to ttyUSB0
```

The last line shows that the device file has been correctly created, and that the kernel has associated the newly-recognized device with the right device file.

Our Linux operating system includes a small terminal emulator program called 'microcom'. By adding a jumper to bridge the RXD and TXD lines of the serial port on the USB-to-serial adaptor we can use this to test the driver. Start up the terminal program using the following parameters.

```
microcom -s 9600 /dev/ttyUSB0
```

If you now type characters at the keyboard they will be sent to the adaptor and then returned to the board. If the driver is working properly these characters should then be echoed to the screen. Break the connection between RXD and TXD on the USB-to-serial adaptor and the characters should no longer be echoed, demonstrating that the device driver is working. Press control-X to close microcom.

Internet Links

[1] sauter@embedded-projects.net

[2] http://en.wikipedia.org/wiki/Operating_Systems:_Design_and_Implementation

[3] http://en.wikipedia.org/wiki/Process_control_block

[4] www.elektor.com/120180

[5] www.elektor.com/120146

[6] www.gnugeneration.com/mirrors/kernel-api/r4299.html

[7] <http://wiki.linuxquestions.org/wiki/Mknod>

```

CC      arch/arm/boot/compressed/misc.o
In file included from arch/arm/boot/compressed/../../../../lib/decompress_inflate.c:10:0,
         from arch/arm/boot/compressed/misc.c:248:
arch/arm/boot/compressed/../../../../lib/zlib_inflate/inflate.c: In function 'zlib fixedtables':
arch/arm/boot/compressed/../../../../lib/zlib_inflate/inflate.c:100:1: warning: the frame size of 2176 bytes is larger than 1024 bytes
SHIPPED arch/arm/boot/compressed/lib1funcs.S
AS      arch/arm/boot/compressed/lib1funcs.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready

```

Figure 10. Output when compiling the (newly extended) kernel.

In the same way we can modify the kernel to use USB adaptors for LAN and WLAN, USB sound cards and many other USB devices.

In the next installment in this series we will look at more devices and interfaces and test them out with a few simple experiments.

(120181)

Advertisement

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



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

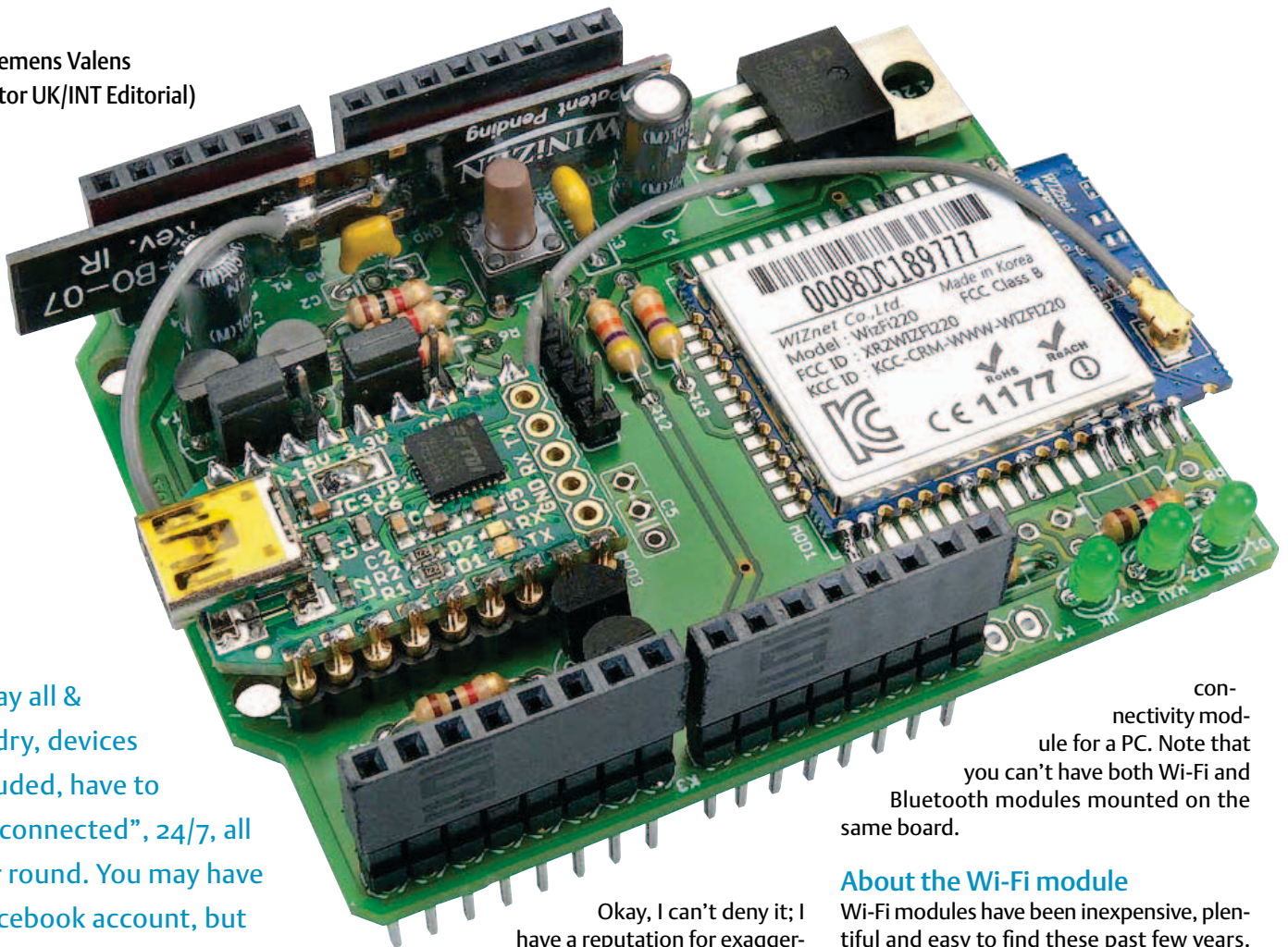
PCB-POOL® is a registered trademark of
www.pcb-pool.com

Beta LAYOUT

Thou Shalt Communicate!

Wi-Fi/Bluetooth/USB shield for Platino and Arduino

By Clemens Valens
(Elektor UK/INT Editorial)



Today all & sundry, devices included, have to be “connected”, 24/7, all year round. You may have a Facebook account, but what about your oscilloscope? Does your multimeter tweet enough? Is your soldering iron linked in? You may be a non-communicative nerd but your bench power supply may be craving for social interaction. With the add-on board described in this article you can hook up anything to everything, with or without wires.

Okay, I can’t deny it; I have a reputation for exaggerating. Connecting anything to, errm, everything is probably a bit over the top. On the other hand, the board I’m about to present in this article is quite versatile. It can be equipped with a Wi-Fi module allowing you to connect an electronic device to the Internet; or with a serial Bluetooth module it offers wireless connectivity to other serial Bluetooth capable devices, and a serial-to-USB bridge enables old-skool wired serial connectivity.

The board can be used as an add-on board for microcontroller systems with Arduino-style extension connectors (Figure 1), like Elektor’s own Platino [1] (Figure 2), but it can also be used stand-alone as a wireless

connectivity module for a PC. Note that you can’t have both Wi-Fi and Bluetooth modules mounted on the same board.

About the Wi-Fi module

Wi-Fi modules have been inexpensive, plentiful and easy to find these past few years. The problem with many of them however is their manufacturer who refuses to tell you how to use the module unless you agree to transfer a substantial amount of money to the manufacturer’s bank account in exchange for a few thousand modules or a customized software library for your host hardware. Luckily the Korean embedded Internet liberators from WIZnet have decided to be less secretive about their products, and sell Wi-Fi modules you can actually use. I managed to get my hands on two of their WizFi220 modules, and this article is the result.

The WizFi2x0 modules (there is also a WizFi210 module) operates with standard

802.11 b/g/n access points at speeds up to 11 Mbit/s (802.11b), and they support WEP, WPA, WPA2-PSK and Enterprise security standards (802.11i). The modules have some really cool possibilities. One of them is the Wi-Fi serial port that's remarkably easy to use. Another nice feature is its limited access point (AP) capability that allows direct connection to smartphones and tablets like Android-based devices and Apple gadgets.

The WizFi220 module measures 32 x 23.5 x 3 mm, has an on-board chip antenna and a Hirose U.FL connector for an external antenna. The module's compatible little brother, the WizFi210, consumes less power, but cannot provide as much output power and hence has "less range". 49 connection pads provide enough soldering points to fix the module on a PCB in such a way that it will not easily come off. Most of these pads are probably not useful for unassuming users like us, being pitched at high-volume customers. For the others a simple two-wire serial connection is available to transmit and receive data and to configure the module.

As is often the case with such wireless modules, the WizFi2x0 can be controlled over a serial link using AT modem commands (or Hayes commands). Such commands consist of human-readable strings a few characters long and starting with 'AT'. The module powers on in Command mode so you can configure it. Once the setup is complete the module can be switched into Data mode with the O(nline) command ("ATO"). To get it out of Data mode back to Command mode three consecutive '+' characters have to be send ("+++").

Design considerations

Some of the configuration commands, like setting the baud rate of its serial port, are more or less one-time only, and the module remembers these settings. Since the AT commands are human-readable it may therefore be practical for the initial setup, but also for experimenting, to be able to communicate with the module using a PC and a serial terminal program. That's why I decided to throw in a serial-to-USB con-

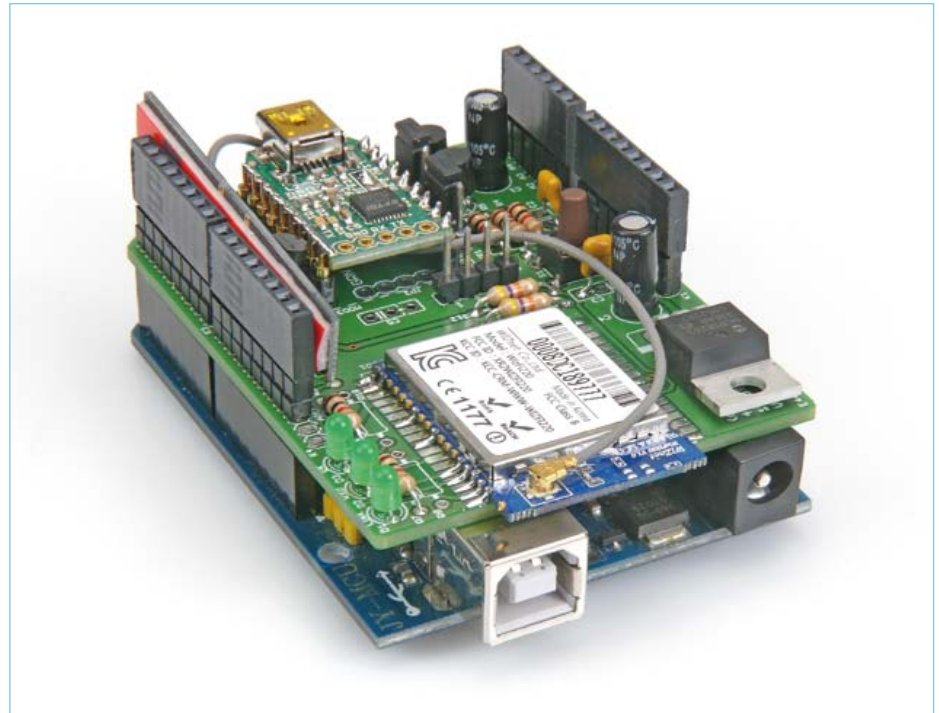


Figure 1. Fully built up Wi-Fi shield on top of an Arduino Uno. Note the absence of R8 and R10.

verter. Since many Elektor readers may already have a so-called FTDI serial USB cable [2], I wanted to be able to use it with this module. On the other hand, having such a converter directly on the board would also be useful, especially if the host system does not have a USB serial port, like Platino. In the September 2011 edition

we presented BOB [3], a very convenient USB/serial bridge module, and I thought it was a good idea to add it to my design too. Now if I could add some clever wiring it should be possible to connect the USB converter to either the Wi-Fi module, the host system or even disconnect it. Flexibility started creeping into the design.



Figure 2. Fully built-up Wi-Fi shield on top of Platino, also equipped with a 20x4 alphanumerical LCD.

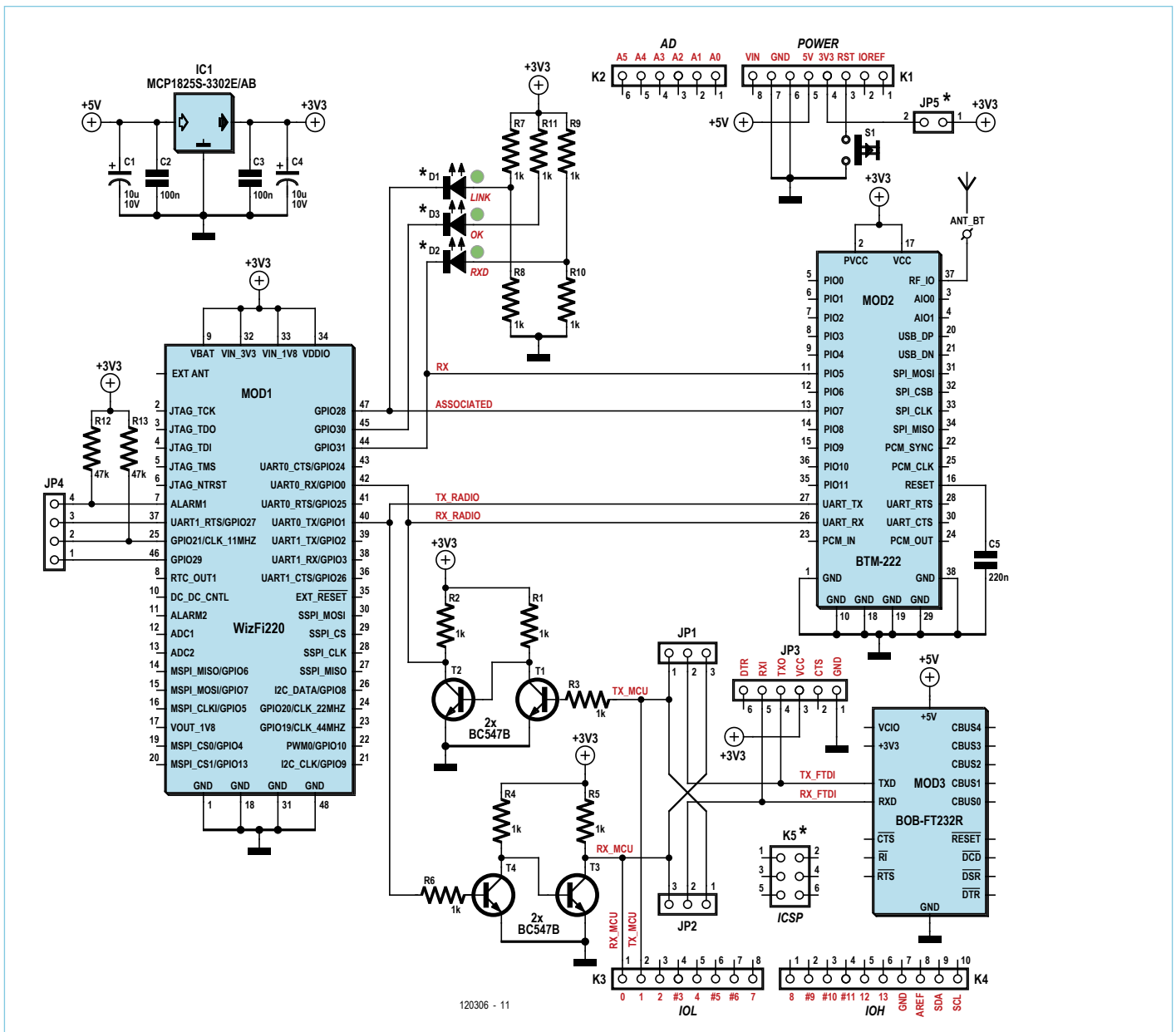


Figure 3. Complete circuit diagram of the Wi-Fi/Bluetooth/USB shield. Refer to the text to find out which parts you need for each configuration.

The Wi-Fi module has to be powered from a 3.3 V rail and it is not 5 V tolerant. The main objective is to connect the Wi-Fi module's serial port to a microcontroller. Such a system may run from 3.3 V, but if it is Arduino or Platino, then the microcontroller runs from 5 V. Level converters on the serial port would therefore be necessary. The FTDI serial USB cable comes in two flavors, 5 V and 3.3 V, and with level converters both types would be usable. The BOB was not an issue because it can handle both levels. Although the host system may have a 3.3 V rail, like Arduino, I thought it wise to add a voltage regulator to the system because

the WizFi220 can consume up to 250 mA in active mode. As an example, the LP2985-33 voltage regulator on an Arduino Uno board is specified for 150 mA, which is clearly not enough. Platino does not have a 3.3 V rail so this shield could provide it too. A jumper should allow connection of the 3.3 V as produced by the shield to the host system. In this section you can replace all instances of "Wi-Fi" by "Bluetooth" without having to change anything else and still have a coherent text. Indeed, many Bluetooth modules are very similar to the WizFi module: they can be controlled with AT commands over a serial link; they run from 3.3 V; they have

similar dimensions and similar requirements. It is therefore not very difficult to add Bluetooth capabilities to the system, all you have to do is wire a usable Bluetooth module in parallel with the Wi-Fi module and that is what I did. I chose the Rayson BTM220 module because I am at ease with it [4] and it is very cheap. We have used it several times in Elektor too. [5][6]

This completes the reasoning that determined my design. Let's now have a closer look at the circuit diagram (Figure 3). If you have grasped my prose up to here, then the schematic will have only a few surprises.

COMPONENT LIST

Resistors

R1–R11 = 1k Ω 5% 0.25W
 R12, R13 = 47k Ω 5% 0.25W

Capacitors

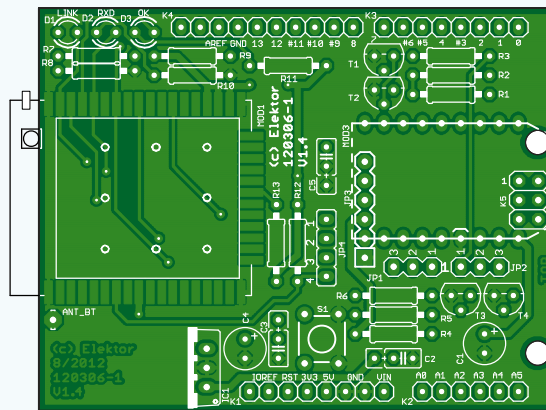
C2, C3 = 100nF 50V, 5mm pitch
 C5 = 220nF 50V, 5mm pitch
 C1, C4 = 10 μ F 35V, 2.5mm pitch

Semiconductors

D1, D2, D3 = LED, green, 3mm diam.
 T1–T4 = BC547C, TO-92 case
 IC1 = MCP1825S-3302E/AB, 3.3 V voltage regulator, TO-220 case

Miscellaneous

JP1 = 6-pin pinheader, 0.1" pitch, straight
 JP3, JP4 = 3-pin pinheader, 0.1" pitch, straight, w. 2 jumpers



K2 = 6-pin stackable header, 0.1" pitch, vertical
 K5 = 6-pin (2x3) stackable header, 0.1" pitch, vertical
 S1 = pushbutton, SPNO, through-hole, 6x6mm
 Mod3 = BOB-FT232R USB-to-Serial Bridge, Elektor Shop # 110553-91, or USB-to-Serial cable, Elektor Shop # 080213-71 (5 V) or # 080213-71 (3.3 V)
 Mod2 = Rayson BTM22x Bluetooth module
 Mod1 = WIZnet WIZFi2x0 Wi-Fi module
 PCB # 120306-1 [1]

JP2 = 4-pin pinheader, 0.1" pitch, straight
 K1, K3, K4 = 8-pin stackable header, 0.1" pitch, vertical

Putting it all together

On the left side of figure 3 we have the Wi-Fi module; on the right side we see the Bluetooth module. As you can see they share the serial RX and TX lines, meaning that you should not mount both. It is either Wi-Fi or Bluetooth, never both. The modules also share two LEDs (D1 and D2); D3 is only used by the Wi-Fi module. A surprise here may be R8 and R10. Maybe I could have drawn a better schematic, but I didn't, so I have to explain this in writing. It is actually very simple. The Wi-Fi module sinks the current for the LEDs whereas the Bluetooth sources the current. As a result the LEDs have to be mounted the other way around as drawn if you use the Bluetooth module. In that case you should mount R8 and R10 and **not mount** R7 and R9 (mounting R11 and D3 is useless unless you connect the LED to a free I/O pin on one of the connectors). For the Wi-Fi module the position of the LEDs is drawn correctly and you should **not mount** R8 and R10 (although nothing

breaks or blows if you do), only R7 and R9 are needed now.

The WIZFi module has some pins with special functions that you may want to use. Since I didn't use them I wired them to a separate connector JP4. To enable you all to stack many shields on top of each other I decided not to wire these signals to the Arduino extension connectors K1 to K4. If you need them you can wire them yourself. Pin 25 (GPIO 21) allows you to restore the module's factory defaults. Pulsing it Low twice will restore the module to Limited Access Point mode, three pulses will restore it to Ad-hoc mode. Pin 46 (GPIO 29) provides a hardware way of switching between Command (High) and Data mode (Low). Pin 37 (GPIO 27) is needed when you want to upgrade the firmware of the module (probably never). To do so you have to pull it High. Finally, a low level on pin 7 (Alarm1) will wake up the module from deep sleep mode.

The level converters (R1-3, T1-2 and R4-6, T3-4) are straightforward and have been used before in Elektor. Transistors T1 and T4 do the level conversion, but invert the signal at the same time. T2 and T3 correct this. The resistor values are not critical. I chose to use as many 1 k Ω resistors as possible. The RX and TX lines connect to JP1 and JP2. These 3-pin jumpers may be a bit difficult to understand at first sight, but when you follow the lines you should be able to figure them out. With a jumper on JP1's pins 1 and 2 the USB serial converter TX pin is connected to the Wi-Fi (or Bluetooth) module's RX input. A jumper on JP2's pins 2 and 3 connects the USB serial converter RX pin to the Wi-Fi (or Bluetooth) module's TX output. This is the Configure-Module-With-PC mode. In these positions the USB serial converter cannot talk to the host MCU system and it may be better to disconnect the board from the host system. The board will be powered from the USB port. If you put a jumper on JP1's pins 2 and 3

Elektor Projects & Products

- Platino, a versatile board for AVR microcontroller circuits (October 2011); PCB # 100892-1 from Elektor Shop
- BOB-FT232R USB-to-Serial Bridge (September 2011); module, Elektor Shop # 110553-91
- USB-to-TTL Serial Cable (June 2008); cable, Elektor Shop # 080213-71 (5 V) or # 080213-71 (3.3 V)

Listing 1.

Example of setting up the WizFi2x0 module as a serial server using AT commands. The commands are in **boldface**, the module's responses are in *italic*. Comments appear in brackets (). See the WizFi2x0 user manual for more commands.

```

AT (wake up)
[OK]
AT+NSTAT=? (what is your status?)
MAC=00:08:dc:18:97:76
WSTATE=NOT CONNECTED MODE=NONE
BSSID=00:00:00:00:00:00 SSID="" CHANNEL=NONE SECURITY=NONE RSSI=0
IP addr=0.0.0.0 SubNet=0.0.0.0 Gateway=0.0.0.0 DNS1=0.0.0.0 DNS2=0.0.0.0
RxCount=0 TxCount=0
[OK]
AT+WPAPSK=germaine,"Philippe Noirette" (set SSID & pass phrase)
Computing PSK from SSID and PassPhrase...

[OK]
AT+NDHCP=1 (request IP number from DHCP server)
[OK]
AT+WAUTO=0,germaine (automatically connect to germaine)
[OK]
AT+NAUTO=1,1,,8011 (setup for auto connect: server, TCP, port 8011)
[OK]
ATA (start auto connect)
   IP SubNet Gateway
   192.168.2.7: 255.255.255.0: 192.168.2.1
[OK]

(now you can connect (telnet) to the module over Wi-Fi)

```

and on JP2's pins 1 and 2 the USB serial converter can be used to talk to the host system. In this case you should not mount the wireless module (or the level converters) as this kind of communication may disturb it. This is the USB-Shield mode. Not installing any jumpers will put the board in Wireless mode and the USB serial converter has no use. However, you could leave it on and wire it to other pins on the extension connectors. This is for instance useful in an Arduino-with-software-UART configuration or in a host system with more than one UART.

Finally some remarks on details. The Reset button is mainly intended for Platino which doesn't have one. It is also useful when the Arduino reset button becomes inaccessible because of the shield mounted on top of it. This depends on the Arduino board and its revision.

K5 is not connected. I only put it on the board to provide access to its signals if the host is Arduino.

Turning practical

To turn a circuit diagram into a real shield or extension board a printed circuit board (PCB) is needed. I designed one and you can download the Eagle CAD files from the web page that accompanies this article [7]. The PCB is the size of an Arduino Uno except that it is a rectangle without the funny-shaped short side of an Arduino Uno. All the components are through-hole types; only the Wi-Fi and Bluetooth modules have surface mount footprints. JP5 and the Bluetooth module are located on the solder side of the board; the other components should go on the component side. Note that if you use non-stackable connectors for K1 to K4 you should mount them on the solder side if you want to stick the board on an Arduino or Platino.

JP1 and JP2 are positioned in such a way that the jumpers should always be in the same position, i.e. both to the left, both to the right or both absent. Actually, you could leave a jumper on JP2 in case you wanted to listen in on the MCU wireless module communications. This can be useful for debugging purposes.

JP3 has a few contacts in common with the BOB serial-to-USB bridge, the idea being that you use either a BOB or an FTDI cable, not both.

For full 3.3 V systems the level converters and voltage regulator can be left off. In this case you have to place two bypasses (wire bridges) from R3-JP1 to R2-T2 and from R6-MOD1 to R5-T3. Resistors R3 & R2 and R5 & R6 are positioned in such a way that the wire bridges are very easy to install and span only minimum distance. You will also have to short JP5 on the solder side of the board.

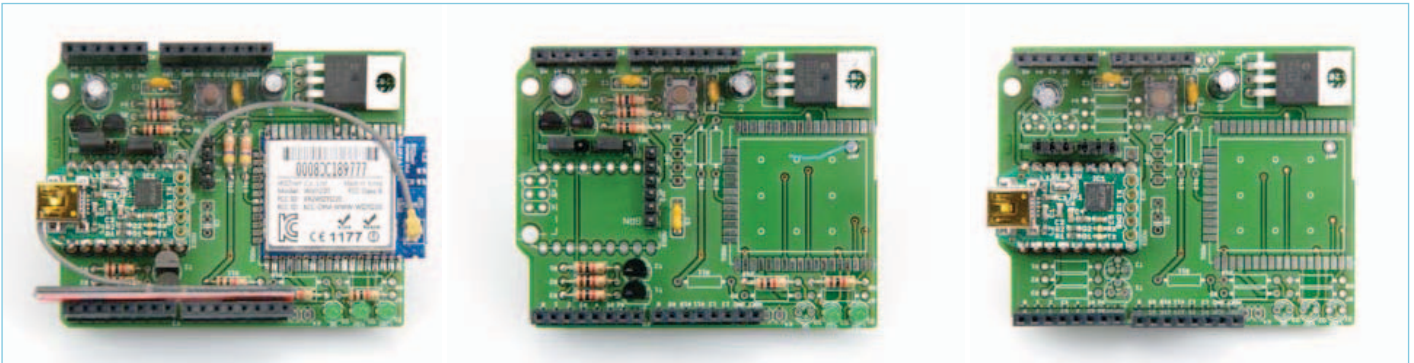


Figure 4. The three main configurations next to each other. The Bluetooth module is not visible (middle board) because it is mounted on the other side of the PCB. Note that this is an early revision of the PCB that has the same shape as an Arduino Uno board. The final PCB is rectangular.

As mentioned before, the orientation of the LEDs depends on the wireless module. The component print on the PCB corresponds to the Wi-Fi module. If you mount a Bluetooth module you should mount the LEDs D1 and D2 ‘the wrong way around’. Also, only mount R7 or R8 and R9 or R10. The 3.3 V voltage regulator can be installed lying on his back or standing up.

Three main configurations are possible (Figure 4), but you may have a need for other variants:

- **Wi-Fi shield** — no need for the BOB if you own an FTDI cable, although you can always install one. JP3 is available for connecting the FTDI cable. Mount all LEDs as indicated on the PCB and do not mount R8 nor R10. Level converters will be needed and the 3.3 V voltage regulator probably too, depending on your host system.
- **Bluetooth shield** — similar as the Wi-Fi shield except for the LEDs. D3 has no function and you should not mount R7 and R9 but mount R8 and R10 instead. As an antenna you can use a piece of (insulated) wire of 31 mm long.
- **Serial-to-USB bridge shield** — just the BOB without level converters. Do not forget to put a drop of solder on JP1 of the BOB to configure it for 5 V or 3.3 V operation. Add 3.3 V voltage regula-

tor and reset push button to taste. The LEDs can be useful too, but you will have to wire them to a connector pin. You should short pins 2 and 3 of JP1 and pins 1 and 2 of JP2 with a jumper or a wire bridge.

Testing

In case of the Wi-Fi or Bluetooth configurations connect the board to a USB port on the PC using an FTDI serial converter cable or the BOB serial-to-USB bridge. Figure out which COM port was created by the operating system (OS) of your PC (make sure you have the appropriate drivers for your OS installed. If you haven’t you can get them, including the necessary documentation, at www.ftdichip.com) and start a serial terminal program. The WizFi220 module has a default baud rate of 115 200 bits/s, for the BTM220 module this value is 19 200. Both modules use 8 bits, no parity and no hardware flow control. In the terminal type ‘AT’ followed by the Enter key. If you set up everything the right way the module should respond with ‘[OK]’ or ‘OK’. If this test succeeds you’re in business. Refer to Listing 1 for a working Wi-Fi example (don’t forget to adapt the SSID and pass phrase to your network).

For a BOB serial-to-USB bridge configuration the steps are more or less the same except that there is no point in punching in AT commands in the terminal unless your host system responds to these. It is up to

you to provide a host that can communicate over a serial link.

Thanks go out to Joachim Wulbeck of WIZnet Europe GmbH (www.wiznet.eu) for providing the WizFi220 modules and the Wi-Fi antennas.
(120306)

Internet Links and References

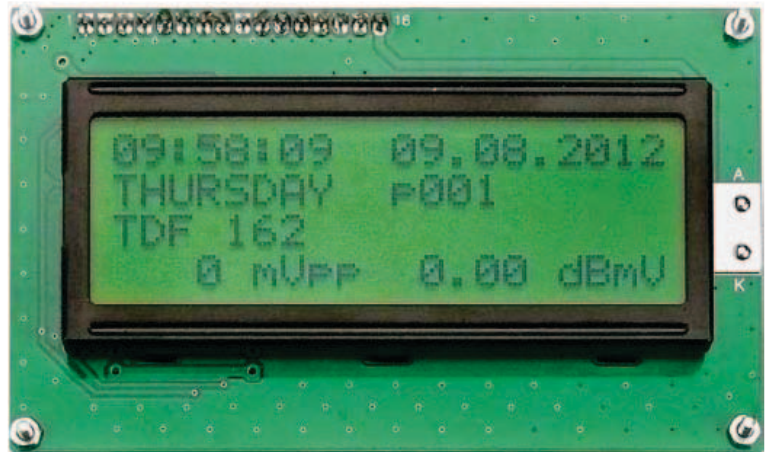
- [1] **Platino:** www.elektor.com/100892
- [2] **USB-to-TTL serial cable:** www.elektor.com/080213
- [3] **BOB serial-to-USB bridge:** www.elektor.com/110553
- [4] **Experiments with Rayson Bluetooth modules:** <http://elektorembded.blogspot.com/2010/08/rayson-btm222-btm112-bluetooth-modules.html>
- [5] **Bluetooth with the ATM18:** www.elektor.com/080948
- [6] **Bluetooth for OBD-2:** www.elektor.com/090918
- [7] **Wi-Fi shield:** www.elektor.com/120306
- [8] **Wi-Fi shield on Elektor Projects:** www.elektor-projects.com/project/wi-fi-bluetooth-usb-shield-for-arduino-platino.12252.html

AVR Software Defined Radio (6)

Decoding options for BBC 198 kHz data signals

By Martin Ossmann (Germany)

This series has demonstrated that the ever-popular AVR controllers are also useful for signal processing. In this final installment we take a closer look at a few more decoding methods. As usual we like to back up the theory with some practice: This installment shows how to extract time-of-day information from BBC AM broadcasts.



In the last installment of this series [5] we demonstrated the decoding of time-signal transmissions. Using the described digital IIR and matched CIC filter it is possible to receive and decode time signals sent by the German DCF77, the UK MSF and French TDF162 stations. In this installment we go on to decode the digital information transmitted by the BBC on long wave at 198 KHz (Droitwich) and also the recently discontinued medium wave 648 KHz service. To this aim we look at alternative decoding strategies and demonstrate how they can be implemented.

Bit-clock synchronization

In the world of communications technology there are many methods by which synchronous data can be sent through the ether. Here a digital data stream runs continuously through the communication system at a defined clock rate. At the receiver the signal often first undergoes demodulation. The resultant signal is then a sequence

of data bits which contain the transmitted information.

For this experiment to achieve bit-clock synchronization with the receive data it will be necessary to use a simple radio front-end (or the receiver board available from the Elektor shop), the receiver software 'EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c', a signal generator with a 125 kHz series resonant circuit and the transmitter software 'EXP-TX-FM-125kHz-RDSlike-V01.c'. This last program generates frequency modulated digital data at 125 kHz with a data rate of 31.25 bit/s. The data stream and bit clock are output from the D/A converter outputs.

At the receiver site the signal is sampled at a frequency of

$$31.25 \text{ Hz} \times 16 = 500 \text{ Hz}$$

to recover the clock. Each bit is therefore sampled 16 times. You could try to reconstruct the 31.25 KHz bit clock from the received

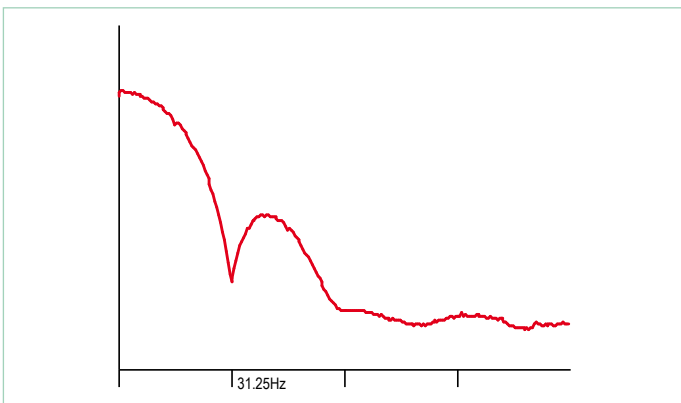


Figure 1. Frequency spectrum of the 31.25 Hz data signal.

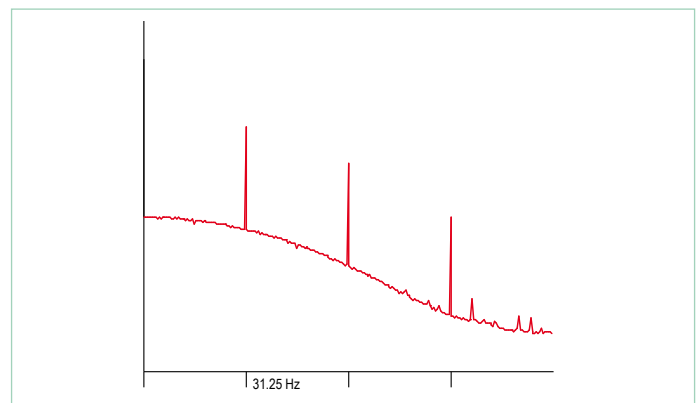


Figure 2. The spectrum of the rectified data signal.

data signal by using a PLL (phase locked loop). Unfortunately it is not quite so simple. The reason becomes clear when you look at the received signal spectrum given in **Figure 1**. There is in fact no signal component at 31.25 Hz for a PLL to lock onto.

Once the data signal has been rectified however, its spectrum (**Figure 2**) now shows a component at 31.25 Hz. A PLL will now be able to lock onto the signal. The principle of the data recovery PLL is outlined in **Figure 3**.

This PLL is in fact similar to the design used in an earlier installment of this series to recover the carrier frequency (see Figure 7 in the fourth installment of this series [4]). With reference to the 'EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c' program the bit-clock frequency is controlled in the following manner: The doSignalSample() routine in **Listing 1** is called at 16 times the bit clock rate. In this routine the variable Clk16 is incremented or reset to zero when it reaches 16. The phase of the saw-tooth signal is regulated so that the center of the ramp (Clk16 = 8) is coincident with the center of the transmitted bit. The bit value is evaluated and processed in the routine doBitSample(). Phase adjustment is performed in the following manner; in the normal case Clk16 is incremented to 16 and then reset by the digital signal clock. Now depending on the sign of the output from the phase comparator this value can be changed to either 15 or 17 to provide some phase adjustment.

In order to correctly control the clock PLL it is necessary to output the demodulated signal at the output of DAC1. Clk16 with Peak at Clk16 = 8 (i.e. at the sampling instant) is output from DAC2. **Figure 4** shows the reception of BBC Radio 4 at 198 KHz LW (see **Figure 5**).

Early-Late Synchronizer

In addition to the conventional PLL a so-called 'Early-Late' synchronizer can be used to recover the clock signal. The eye-diagram shown in **Figure 6** helps explain how this technique works.

Again it is necessary for the Clk16 signal to synchronize with the bit clock so that the data signal is sampled in the center of the bit. In order for this to be achieved the signal is sampled at the early time E and the late time L. This can be performed, for example on the third (E) and twelfth (L) clock of the Clk16 clock. When both of these measured values are the same amplitude then we can be confident that E and L are centered about the midpoint. When on average one set of the values is greater than the other it indicates that the eye has shifted from its optimal point and phase of Clk16 requires some correction.

The program code corresponding to this process is given in **Listing 2**. Ten samples of the E (early) and L (late) values are summed to produce the value 'earlyLateDelta'. This value is then used to correct the Clk16 period. The code for this synchronizer can be tested with the 'EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c' program.

Decoding the phase modulated BBC198 signal

Now we come to a practical application of bit synchronization when we program the receiver to recover the phase-modulated data sent out by the BBC as part of its Radio 4 LW broadcast on 198 kHz. Details of the transmitted signal are given in the BBC publication

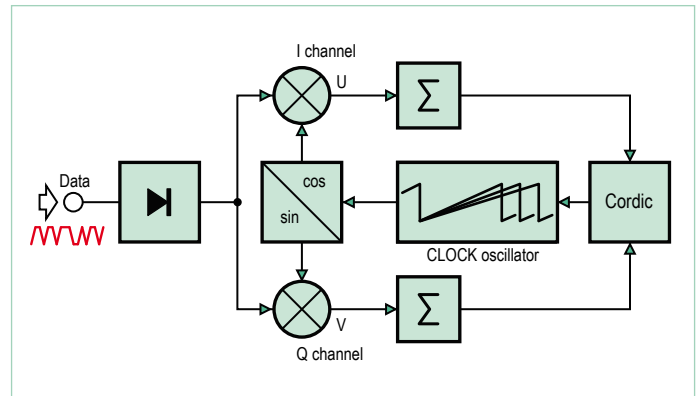


Figure 3. The clock-PLL principle.

Listing 1: Phase adjustment

```
void advanceCLK16() {
    Clk16++;
    if ( Clk16 >= ClockPeriod ) { Clk16=0 ; }
}

void doSignalSample(){
    int16_t Signal ;
    ATOMIC_BLOCK(ATOMIC_FORCEON) {
        Signal=FRQcicout ;
    }
    ClockPLL16(Signal) ;
    doBitSample(Signal) ;
    advanceCLK16() ;
}
```

[7]. Data is sent by modulating the 198 KHz carrier phase by $\pm 22.5^\circ$. The data rate achieved is 25 bit/s using Manchester coding (also known as bi-phase coding) with filtering. Using this type of modulation ensures that the average phase deviation always sums to zero

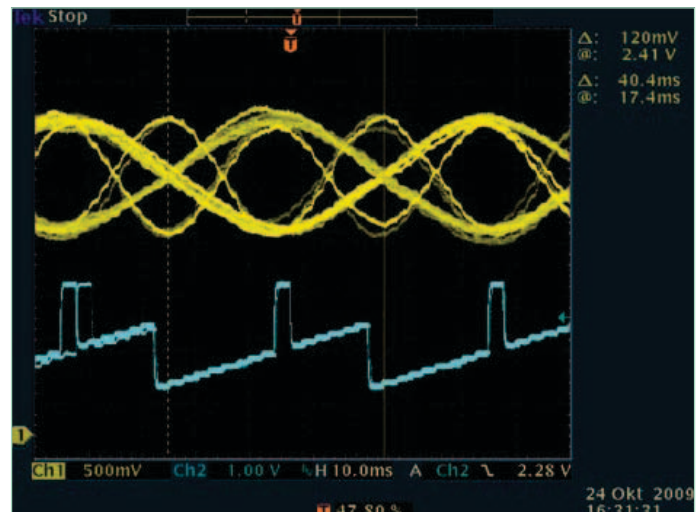


Figure 4. Eye diagram of the BBC198 signal.

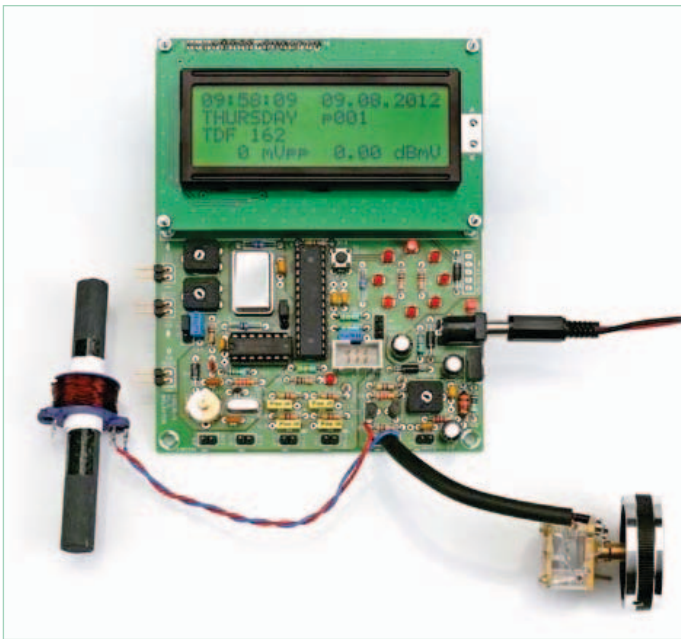


Figure 5. The complete receiver for reception of BBC198.

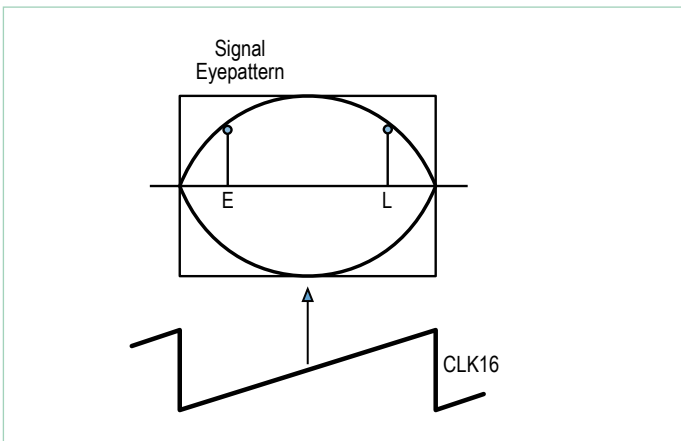


Figure 6. Eye diagram of the early-late synchronizer.

```

Listing 2: Early-Late Synchronizer
void earlyLateSynchronizer(int16_t Signal){
  if ( Clk16==3 ) { earlyValue=abs(Signal) ; }
  if ( Clk16==12 ) { lateValue=abs(Signal) ; }
  if ( Clk16==8 ) {
    earlyLateDelta += lateValue-earlyValue;
    ClkCnt++;
    ClockPeriod=16 ;
    if ( ClkCnt==10){
      ClockPeriod += earlyLateDelta/16 ;
      earlyLateDelta=0 ;
      ClkCnt=0 ;
    }
  }
}

```

so (as before the modulation was introduced) the carrier can still be used as a highly stable frequency reference. The A/D converter uses a clock equal to

$$20 \text{ MHz} / 2,500 = 8 \text{ kHz.}$$

At this sampling rate the 198 kHz carrier is mixed down to

$$198 \text{ kHz} - (25 \times 8 \text{ kHz}) = -2 \text{ kHz.}$$

The ‘intermediate frequency’ is therefore –2 kHz and is sampled at four samples per period in the same way as in the IQ sampling mixer earlier. The negative sign indicates that the mirrored sideband is used. After separation into the *I* and *Q* components both are first passed through second-order CIC filters and downsampled with an M_d factor = 16. Next in line is another low-pass CIC but this time without downsampling. Instantaneous values of amplitude and phase values are now produced from these filtered *I* and *Q* signals processed using the CORDIC process.

One possibility for signal demodulation is to use a PLL to recover the carrier and then decode the NRZ (Non Return to Zero) signal. Here it refers to the original data signal which has been used to phase modulate a carrier and which will be recovered in the receiver. For decoding however the use of a PLL control loop which interprets a phase modulation as a frequency modulation is not necessary. The conversion is shown in **Figure 7**.

The theoretical NRZ-phase response is represented as a signal response $p(t)$. To reduce the signal bandwidth phase changes occur only slowly. The actual phase response is more like the signal $q(t)$. The sampling rate for the demodulated signal is 400 Hz so each bit is sampled 16 times. Now according to communication theory frequency is the rate of change of phase. The frequency response $f(t)$ is given by the derivative of $q(t)$ with time. When the phase increases linearly with time there is a constant positive frequency. When the phase decreases linearly with time there will be a constant negative frequency.

Using additional low pass filtering the signal $g(t)$ is recovered from $f(t)$ and can now be readily evaluated. The data stream is sampled mid-bit to determine its value. When the value is positive it is interpreted as a ‘1’ and when negative as a ‘0’. Application of the CORDIC algorithm produces phase samples to give the instantaneous frequency. These are again low-pass filtered with a CIC filter to produce the signal $g(t)$. This signal in turn supplies the bit-clock recovery. This results in the waveform shown in Figure 8 in the second installment of this series [2].

With more processing the blocks of data are identified and decoded. How this functions will be explained later on. For now **Listing 3** shows a data sample received from the BBC 198 kHz signal.

Block Synchronization

Once the data bit stream has been recovered from the receive signal the final task is to identify the beginning and end of the data packet. This process is known as block synchronization. The majority of radio traffic systems use blocks of data of fixed

length. When the receiver is first switched on it is necessary to determine where the beginning of each block occurs in the data stream. There are several commonly used techniques to identify data start; one method is to use a unique sequence of bits to form a preamble. This sequence is never allowed to occur in the data field. RFID tags of the type EM4102 use this method. This preamble occupies space in the data stream and therefore, to some extent reduces the available channel data bandwidth. Many communication systems use the information contained in the data for error correction field as the block boundary marker.

Error detection

Many data communication systems use a checksum added to the end of each block of data which allows the receiver to detect data corruption and also to correct the error. It can be shown that this additional information can also be used to perform block synchronization. First we will describe how the checksum is calculated using the 125 KHz test transmitter together the 'EXP-TX-FM-125kHz-RDSLlike-V01.c' program. The calculation software is given in **Listing 4** and the CRC hardware generator block diagram is shown in **Figure 8**.

Ten CRC check bits are calculated using a shift register with XOR gated feedback taps. The 16 data bits are clocked through a shift register and control the new bits which end up in the syndrome shift register. The CRC checksum is produced after 16 clock periods corresponding to the 16 data bits. A hex value of 198H is added to the checksum using the XOR function which simplifies the synchronization process at the receiver site. The 10 check bits are sent after the 16 data bits. This technique is very similar to the standard RDS system described in [8].

When the receiver knows where a new block starts it computes the syndrome (in exactly the same way as the transmitter) of the following 16 data bits and compares it with the 10 CRC bits following the data. In order to find the start of the block it is necessary to check each time a new bit is received whether the previous 26 bits produced an error-free combination of 16 data bits and 10 checksum bits. This test produces an 'OK' at a spacing of 26 bits while receiving a continuous data stream. Occasionally an 'OK' will occur in an incorrect position. The block synchronization algorithm looks for a pattern with three 'OK' signals spaced by 26 bits. When this is detected it assumes that the block start has been correctly identified.

Figure 9 shows schematically how the check bits are continuously calculated. The syndrome calculator can be looked on as a type of filtering. The feedback shift register is configured in the same way as the one used in the transmitter path. The continuous stream of bits are processed and as each bit emerges from the 10+16 bit shift register it is taken into account in the syndrome shift register. The software need only check if the syndrome shift register contains the value 198H which was the value added during encoding at the transmitter end. When this value is found the software responds with an 'OK'. A counter called 'localTime' is incremented from zero to 25. A value of zero indicates that a complete block together with the check bits have been processed.

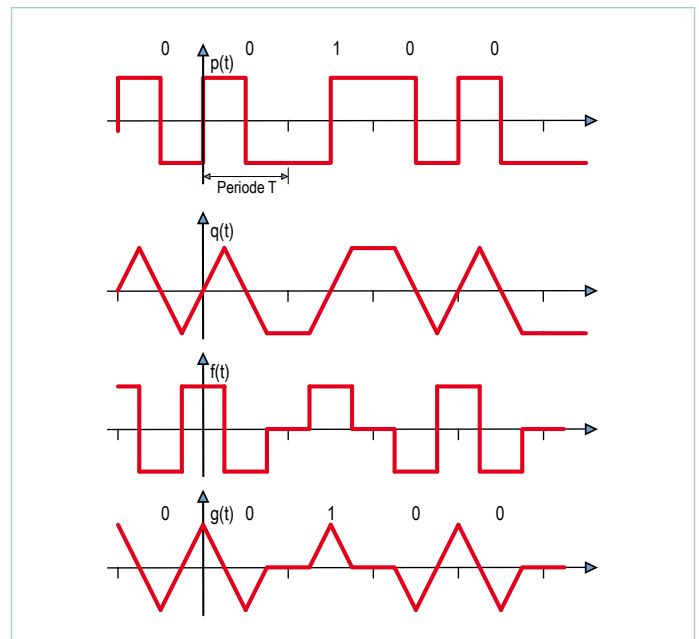


Figure 7. Phase modulation as frequency modulation.

Listing 3: BBC 198 kHz signal: receive Data

```
+E EB11 0391
+0 AAAA AAAA
+E E8F1 0391
+E EB09 0391
+E EB49 0391
+0 AAAA AAAA
+E EB31 0391
+E EB41 0391
+0 5475 2980 time= 18:38 week=07 day=02 tuesday
+E EB29 0391
```

Listing 4: Transmitter CRC calculation

```
SendBits(w,16) ;
Syndrom=0 ;
for (j=0 ; j<16 ; j++) {
    Syndrom=Syndrom<<1 ; // outgoing bit is at mask
    0x0400
    if (w & 0x8000) {
        Syndrom ^= 0x0400 ; // xor into outgoing bit
    }
    w=w<<1 ;
    if ( Syndrom & 0x0400 ) {
        Syndrom ^= 0x1B9 ;
    }
} ;
Syndrom=(Syndrom^ofs)<<6 ;
SendBits(Syndrom,10) ;
```

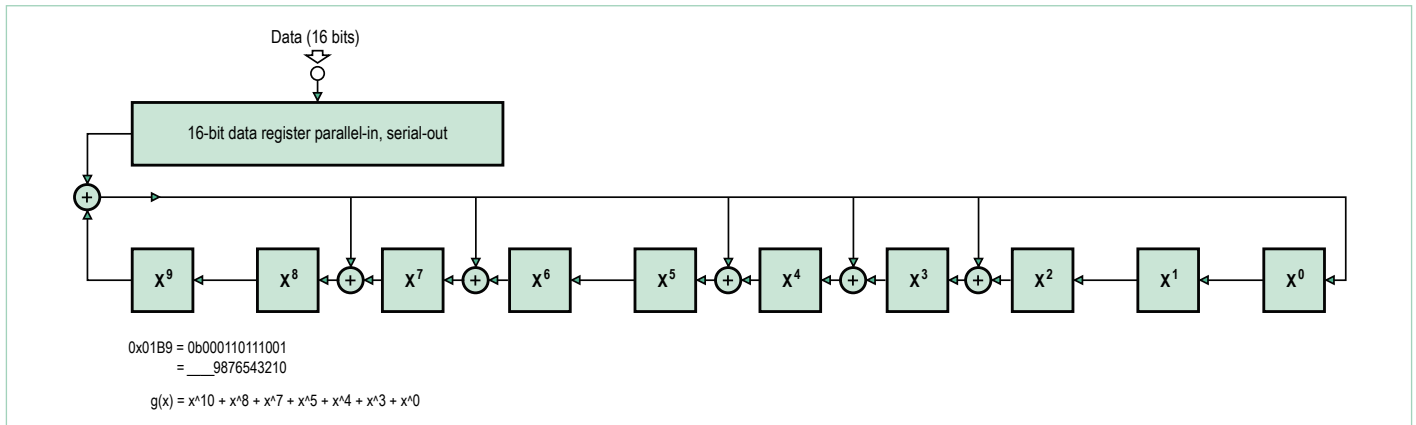


Figure 8. CRC1.

To demonstrate this behavior, a variant of the receiver was programmed so that the actual time (localTime) is output as a sawtooth waveform from the PWM-DAC output. The second PWM DAC output will be high when the synchronization circuit detects an 'OK'. The repetitive waveform can be seen in **Figure 10**. Also visible is a position where a false 'OK' is detected. It is may be necessary to wait a little before this random event happens. After synchronization when the 'OK' occurs at the time interval zero it indicates that error-free block has been received and the data can be output. With this the data reception process is complete. This description of synchronization applies for the BBC long wave transmission on 198 kHz. The data sent by the BBC does not however contain RDS parameters but uses blocks of data containing 47 data bits with a 13-bit checksum. The data is described in detail in the documentation [7] and is relatively easy to extract. In particular, for simplicity the time-of-day information is simply coded in one complete block.

BBC648 AMSS reception and decoding

This last exercise was intended to demonstrate data reception from the European World Service signal which the BBC transmits on 648 KHz. In the mean time due to budget cuts transmission on this frequency has sadly been discontinued. It is however instructional to demonstrate how the AMSS data is decoded in this case. Up until recently the BBC sent low frequency AMSS data as part of

its 648 kHz AM transmission (see the associated literature [10]). The front end of our receiver samples at 20 MHz / 1,875 = 10,666... kHz. This results in an intermediate frequency of:

$$648 \text{ kHz} - 61 \times 10,666... \text{ kHz} = -2,666... \text{ kHz} = 0.25 \times 10,666... \text{ kHz}.$$

Bi-phase signalling is used with a phase modulation of $\pm 20^\circ$. With regard to the modulation and bit-coding method used by the data traffic it is (almost) identical to 198 KHz system described earlier. The difference here is that each block consists of 36 message bits and 11 CRC check bits. Two blocks form a group and a defined number of groups form a DRM-SDC block. A complete description of all the features is available in the DRM specification. Our simple receiver only extracts and displays the name of the transmitter station but is, never the less, a fully functioning data receiver.

Listing 5 shows the receive data. The output '2/7+' indicates that the second group from seven has been received with no detected errors. The data is decoded when all seven groups have been received.

And finally...

This now brings to a close the six-part series on signal processing and 'Software Defined Radio' using AVR microcontrollers. It has powerfully demonstrated just what can be achieved with even relatively modest microcontrollers. Commercial SDR receivers employ

Listing 5: BBC 648 kHz signal: AMSS SDC data

```
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
```

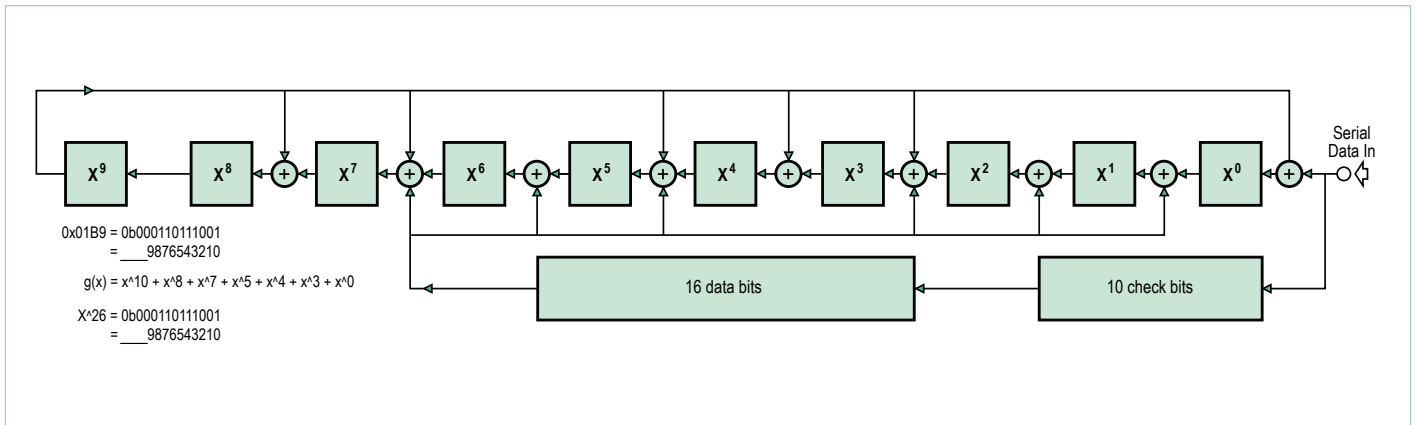


Figure 9. CRC2.

exactly the same techniques as detailed here; the only difference is that they use more powerful DSPs and radio front-ends which of course also makes them correspondingly more expensive!

(120392)

Internet Links

- [1] www.elektor.com/100180
- [2] www.elektor.com/100181
- [3] www.elektor.com/100182
- [4] www.elektor.com/120088
- [5] www.elektor.com/120089
- [6] www.elektor.com/120392
- [7] <http://downloads.bbc.co.uk/rd/pubs/reports/1984-19.pdf>
- [8] <ftp://ftp.rds.org.uk/pub/acrobat/rbds1998.pdf>
- [9] www.drm.org/
- [10] www.ebu.ch/fr/technical/trev/trev_305-murphy.pdf
www.broadcastpapers.com/whitepapers/ABUBBCamss2006.pdf?CFID=16508900&CFTOKEN=dac28b1a87e54d77-47F9A337-9A30-F5E3-667FAB2A9EA27223

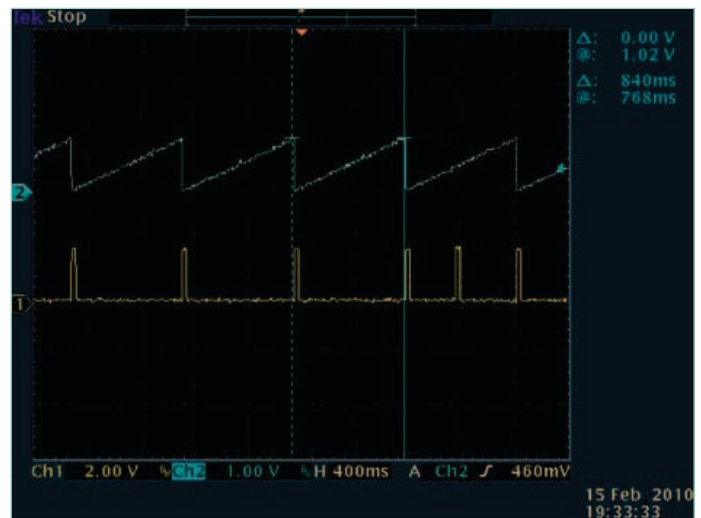


Figure 10. The OK signal.

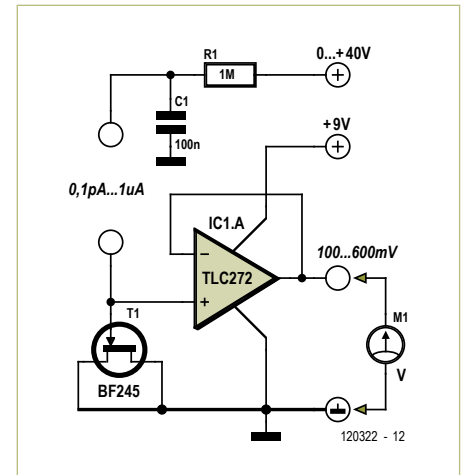
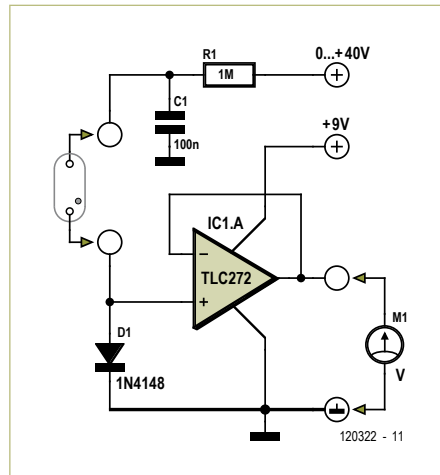
Elektor products and support

- Signal generator (Kit with PCB and all components, # 100180-71)
 - Universal receiver (Kit with PCB and all components, # 100181-71)
 - Active ferrite antenna (Kit with PCB and all components, # 100182-71)
 - Combi-kit with all three components plus USB/TTL converter BOB FT232: # 100182-72
 - USB/TTL converter BOB FT232, fully populated and tested, # 110553-91
 - Free software downloads (hex files and source code)
- All of these products and downloads are available through the web page: www.elektor.com/120392

Picoammeter

By Burkhard Kainka (Germany)

It is often necessary to deal with tiny currents in the picoamp to microamp range when making measurements in ionization chambers and other radiation sensors, as well as when testing insulation.



The instrument described here can be used to measure currents from around 0.1 pA to 1 μA, without the need to change range. One approach is to exploit the logarithmic characteristic curve of a silicon diode, with the diode voltage buffered by a type TLC272 CMOS opamp. For the first experiment we used a 1N4148 silicon diode (see I/h circuit). However, the lowest current measurable using this arrangement was over 10 pA, as below that value the diode's characteristics deviate from the logarithmic curve.

the curve can be extended to even lower currents. Experiments show that it is even possible to get useful results with currents of less than 1 pA. In this case, however, it is essential to carefully screen the whole circuit, including the item under test, against the effects of external fields. For the prototype a metal tin was used with feed-throughs.

- ionization chamber with a sample of pitchblende: about 1 pA;
- BPW34 photodiode used as a radiation detector (in complete darkness): about 10 pA;
- burnt-out filament lamp: about 100 pA;
- burnt-out halogen lamp: about 0.1 pA.

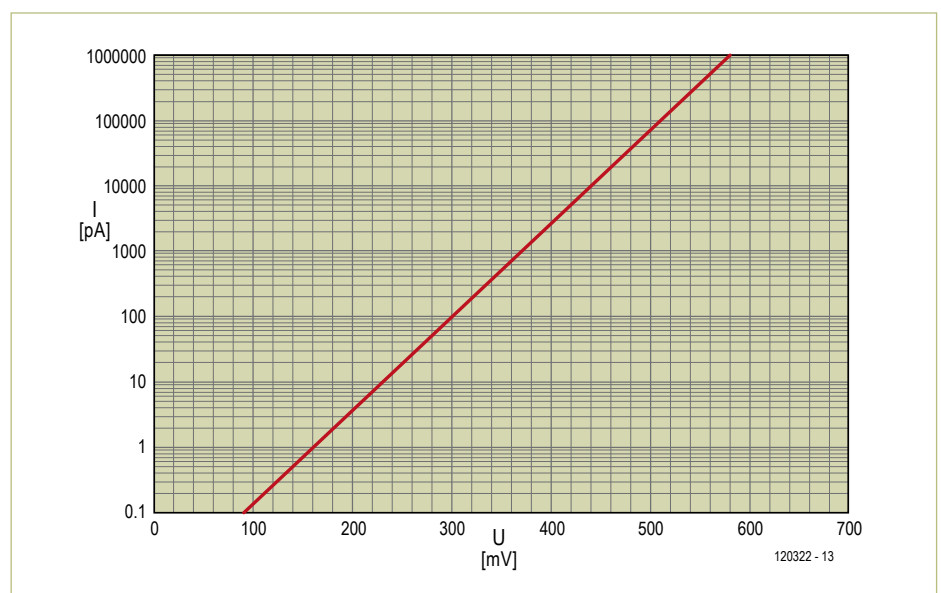
From these last two results we can see that quartz glass is a considerably better insulator than ordinary glass.

(120322)

A particularly good diode with a very low reverse current is the gate-source diode inside a BF245 JFET (see lower circuit). Using this we can measure currents of less than 1 pA. To calibrate the circuit we use known currents at the input and measure the output voltage:

1 μA	580 mV
100 nA	510 mV
10 nA	440 mV
1 nA	370 mV
100 pA	300 mV

The above values closely follow a logarithmic curve: each increase in current by a factor of ten gives an output voltage increment of 70 mV. We have already covered four decades: the graph shows how



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

DLP-USB1232H: USB 2.0 UART/FIFO

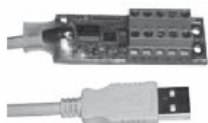
HIGH-SPEED
480Mb/s

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



DLP-IO8-G

8-Channel Data Acquisition



Only \$29.95!

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

DLP-IOR4

4-Channel Relay Cable

DLP-TH1b

Temp/Humidity Cable

DLP-RFID1

HF RFID Reader/Writer

DLP-FPGA

USB-to-Xilinx FPGA Module



www.dlpdesign.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



Programmable Controllers:

Item #1336: Wixel Programmable USB Wireless Module



Motor Drivers:

Item #2502: Dual VNH5019 Motor Driver Shield for Arduino



Custom Laser Cutting:

Design Your Own Chassis



Finding the right parts for your robot can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products - from actuators to wireless modules - that can help you take your robot from idea to reality.

Pololu
Robotics & Electronics

www.pololu.com

IMMERSE YOURSELF



Santa Clara Convention Center Oct 30-Nov 1, 2012

Welcome to the premier annual industry event for engineers and developers of next-generation smart, connected devices. Explore chip and system design. Connect with software development pros. Ponder the next phase of innovation in IT for the enterprise. It's all here at ARM® TechCon™, where the future starts today.

Register online and **Save 15%** with promo code ARM15.
Register here: www.ARMTechCon.com

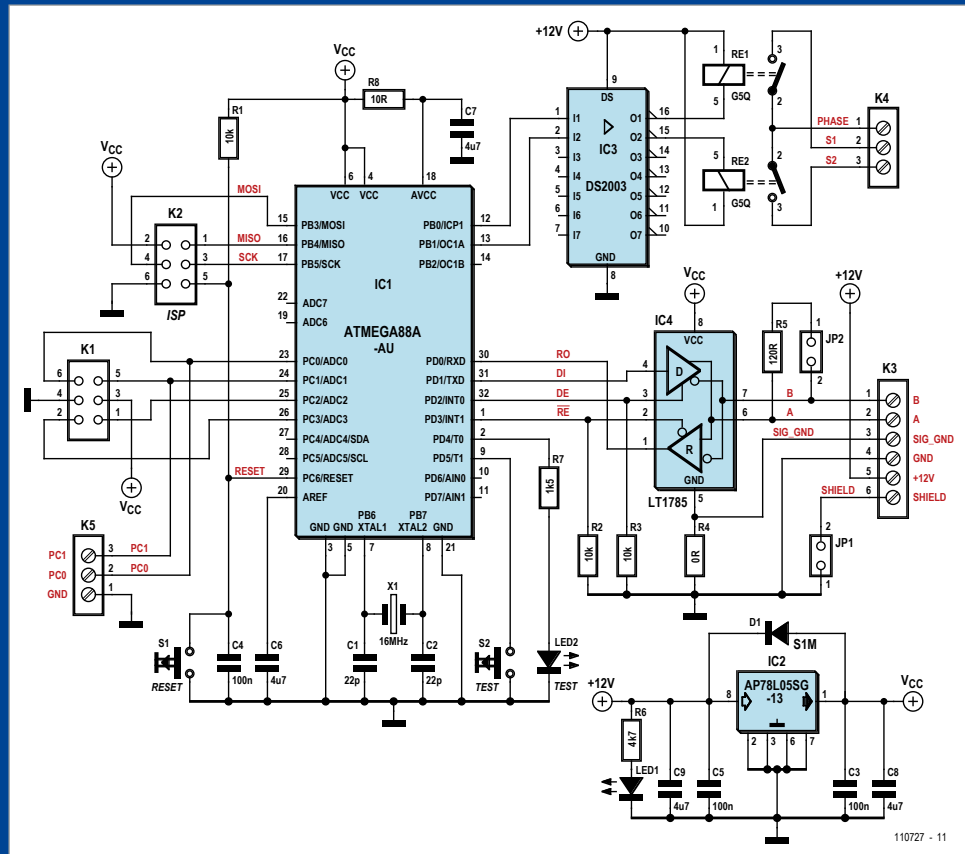
**ARM®**
2012 TechCon™
Join the community defining the future

Shifting RS485 Grounds

By Jens Nickel
(Elektor Germany Editor)

Good news for all automation enthusiasts in general and ElektorBus friends in particular: more bus hardware is in the pipeline, including a larger Xmega board with display and web server module that you can use as a control center. The C firmware library is on the way. More at www.elektor-projects.com! Meanwhile, here's a small E-Labs Inside installment, hopefully with yet more educational value. When we designed our RS485 relay Board [1] an extra access point was provided to the RS485 driver's signal ground (SIG_GND = signal ground). The idea was to pass on the RS485-driver ground from node to node, in parallel with the ElektorBus ground for the supply of each node (GND). This was explained in the article: "When loads are switched it is possible that there will be local interference generated on the bus ground; and if the ground potentials at the drivers differ too widely this can result in spurious bits being detected on the bus. One solution might be to connect signal ground directly to bus ground at only one point on the bus, while in the other bus nodes the two signals are linked via (for example) a 100 Ω resistor. ... We have not yet tested this possibility..."

We received an email on this subject from Elektor reader Andreas Schoenberg, who has been professionally engaged with RS485 bus systems (among others) for about 25 years. Our desire to have a 'reference ground' that's always at the same level for all the bus drivers, appeared reasonable. However with the LT1785 the common-mode voltage at A/B is allowed to be in the range of -7 to +12 V (w.r.t. ground). Even when the ground potentials of two bus nodes differ by 7 V, all bus levels would still be interpreted correctly. Andreas Schoenberg continued, "Statically this cannot occur in the circuit. If 7 V appears to drop on the GND line to some high-current bus node, then an equal amount applies to the supply line, meaning a total of 14 V would be lost." Surely that's not possible, because the bus supply voltage is only 12 V. He continued: "Dynamically, it certainly looks different — if a relay is energized in a module, this can cause a short voltage dip. If so, this should be counteracted with local capacitances at the supply pins of IC3. Provided sufficient local buffering is available, I believe that ground shift problems cannot occur in your system or in the transceivers used, to



the extent of disrupting the communication.”

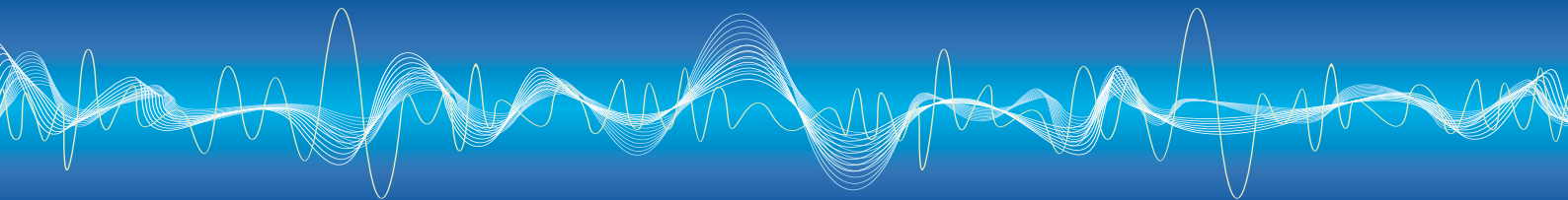
However, if you actually have a bus node that draws so much power and/or is connected using long, thin wires, that GND is at 3 V above SIG_GND due to a voltage drop, other problems can be expected. The LT1785 at its V_{CC} pin invariably ‘sees’ a voltage that’s 5 V above the local GND potential — hence in this case it is powered at 8 V. However the DE pin is pulled to GND via R3, and DE cannot be pulled lower by the CPU because that too is referred to this ground potential only. For the transceiver however, that’s still 3 V above its own ground connection (SIG_GND), i.e. always High. The Driver Enable pin is then out of control by the software!

So much for theory. It would be interesting now to know how you got along with the wiring of the ElektorBus hardware and if you ran in to any problems. Here at Elektor we already established that the communication is very robust, even with long, el-cheapo cables and four lines [2]. Which cables did you use? Tell us on editor@elektor.com

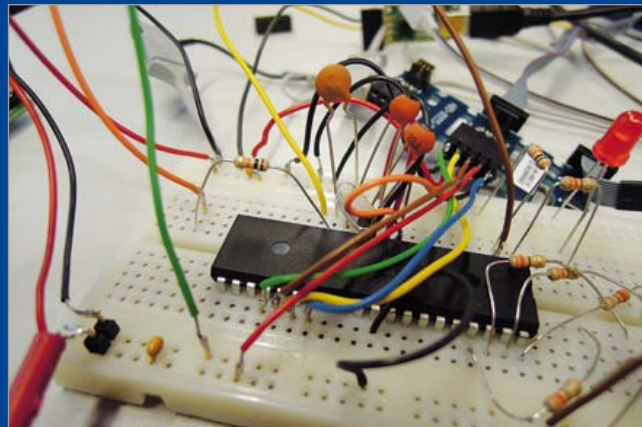
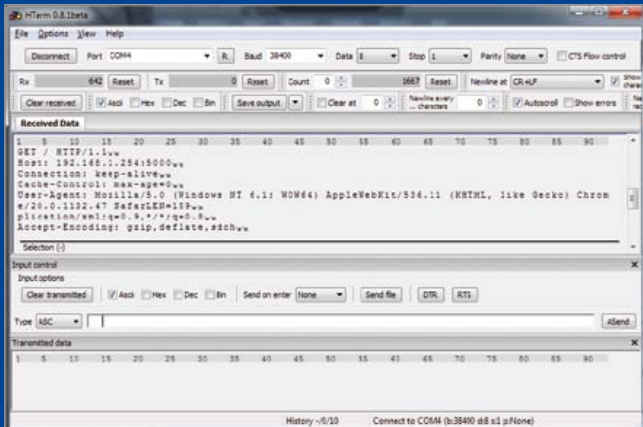
(120501)

[1] www.elektor.com/110727

[2] www.elektor.com/120198



AVR reset-out-of-the-blue issue



By Koen Beckers (Elektor Labs)

Perhaps you're familiar with this next problem: during communication between your PC and an AVR microcontroller in your new circuit the microcontroller resets itself abruptly for no particular reason. This happened to me while I was testing my design for a WiFi LED strip.

Having tried several configurations and done a number of tests, I determined that it happened when I tried to have the microcontroller receive a large packet of data via its UART port. Sometimes it seemed to work, but most of the time the microcontroller was resetting itself and occasionally it crashed instantly after the transmission started. Very occasionally it kept resetting itself and the only thing to resolve this was to shut off the power supply.

'Consulting' Dr. Internet, I learned this was a very common issue with AVR microcontrollers. Luckily a simple solution is available:

The UART connection needs to be fitted with a series resistor of 10 kΩ and a bypass capacitor of 100 pF essentially creating a low pass filter at about 159 kHz.

My issues were with the reception of data, so I inserted a series resistor in the RX line and added a capacitor from RX to GND. A few lines of test code looping back the data received on RX to my PC (see the screenshot) confirmed the correct functioning of my circuit. All data got received correctly without the microcontroller crashing.

Just to make sure it wasn't my freshly written chunk of test code that fixed the problem, I removed the RC filter. Running my test code again resulted in an immediate crash of the microcontroller. After reinserting the RC filter all worked flawlessly again. I guess it's safe to say this simple RC filter solved my issue. Perhaps it can solve yours too!

(120386)

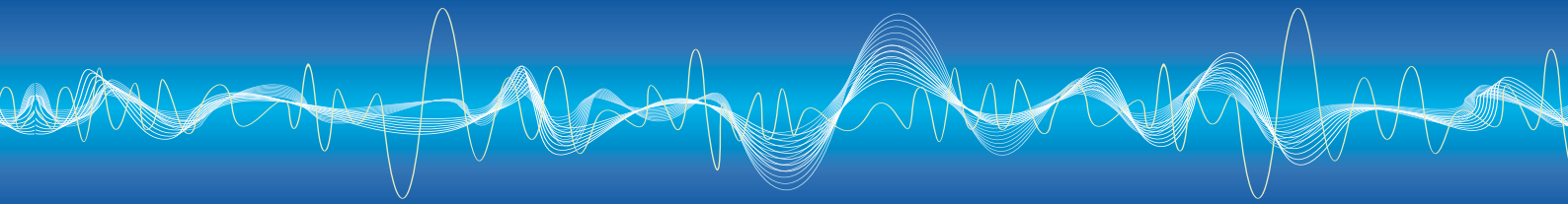
E-Pollution!

By Thijs Beckers (Elektor Editorial & Labs)

As always testing and measuring is part of the daily lab activities and very rarely problems pop up of the really unexpected type. When fellow lab worker Ton Giesberts was testing a DCF77 (European time signal) receiver for our popular AVR-SDR article series (series started in March 2012 edition [1]), he ran into some serious issues.

It proved plain impossible to receive the radio signal inside the lab. He even tried different locations inside our castle but to no avail. Admittedly, the castle walls are quite bulky (approx. 80 cm, 2.5 feet) and may prevent the signal from entering, but even when the antenna was mounted on a 2 m (7 ft) long scrap





batten held out through a window so the antenna was well outside the castle walls, there was still zero signal. Suspecting the WiFi routers, PCs, laptops, fluorescent lighting, switching wall warts and DECT telephones could be interfering with the signal, Ton brought an old portable AM radio with long wave band reception from home and started walking around in the building, trying to receive the old faithful BBC4, the purpose of this being to find the most 'silent' place in the castle. This radio station transmits on long wave at 198 kHz. But nowhere in the castle did he find a sign of the signal. It was all dirt and noise — e-smog (Hey, didn't we have a handy e-smog tester shortly? Yes, we had. It's the TAPIR. Go check it out [2]!). Only after taking the radio more than 5 m (15 ft) outside of the castle walls, a weak signal of the station could be heard.

When entering the building again — with the radio switched on — Ton noticed quite some interference signals when he held it next to the fluorescent lighting. Turns out they generate quite some noise exactly in the 198 kHz range. So one (very) early Monday morning, when there were very few staff present and all the lights were still off, he tested again. But still there was far too much noise coming from PCs, WiFi-routers, DECT telephones, printers and other electronics that never goes silent.

So much for testing sensitive equipment inside the castle. A proper worst case scenario test though; when a circuit passes testing inside our labs, it sure is interference-proof...

Fellow lab worker Luc Lemmens experienced the same problems with a design soon to be released. He decided to take his testing matters elsewhere: home. As soon as he powered up his prototype (also with a DCF77 receiver) everything came to life and functioned as it should — right out of the box! Ton ended up sending the prototype to an esteemed external author for testing. And it worked instantly when he tested it. So the lesson to be learned is: Do you want to test a prototype with radio receiving capabilities? Don't be too disappointed when it doesn't work when testing it in your lab environment. Chances are your circuit works a treat, but e-smog from your electrical devices is interfering with the reception of the signal.

(120504)

Internet Links

[1] www.elektor.com/100180

[2] www.elektor.com/120354

Your all-time favorite component

By Thijs Beckers (Elektor Editorial & Labs)

As an electronics engineer you often get to deal with electronic as well as electrical components. Some are new and just hit the market, others are old and have been here for ages.

Like with so many things, you probably have your own preferred voltage regulator, Schottky diode, microcontroller, capacitor type or even vacuum tube. But with vast numbers of Elektor readers out there, there are bound to be differences. So here's a call to explain to us why this Gunn Diode you always refer to

is your favorite. Tell us your experiences with that BRY39 and elaborate on your favorite IC. Does the LM729 hold any secrets for you? And why do you still use that 6502 for almost every job?

We look forward to a flood of emails in our inbox (myfavorite-part@elektor.com) and hope to be surprised by striking stories and notable narrations on your preferred electronic component.

(120503)



Energy Harvesting with the LTC3108 & LTC3588-1

By Raymond Vermeulen (Elektor Labs)

Imagine you have a sensor, which only has to report its results very infrequently and is located far away. A direct power supply is not practical because of the distance, a battery does not last forever and will eventually have to be replaced. Now what? Energy harvesting! Modern sensors can be very frugal and just like small microcontrollers they often have a power-saving mode of operation. If the average energy requirement in this power-saving mode is very small, then it would be possible to charge a capacitor during this long interval. This capacitor can then power the sensor for a brief moment, after which it returns to a sleep state. During this sleep state the capacitor is charged once again. Below I describe two ICs capable of deriving energy from various ambient sources. The LTC3108 is specifically intended to operate in combination with a Peltier element. The LTC3588-1 is more suited for piezo elements. A circuit idea could be to power the ams lightning detector from last month, together with a microcontroller and RF module, using energy harvesting.

LTC3108

The LTC3108 can obtain energy from voltage sources that supply very small voltages, starting from about 20 mV(!). This enables the use of a small Peltier element with only a very small temperature differential. The manufacturer's datasheet lists a number of other potential sources. The operation is the same as that of a boost/step-up converter with an adjustable output voltage.

An LDO with an output of 2.2 V is also present. There is an additional Vout output that can be switched on and off and can be used, for example, to power a small circuit that needs to be turned on only very briefly and does not have to be placed in some kind of power-down mode. An alternative to this chip is possibly the bq25504.

LTC3108 Datasheet
<http://cds.linear.com/docs/Datasheet/3108fb.pdf>

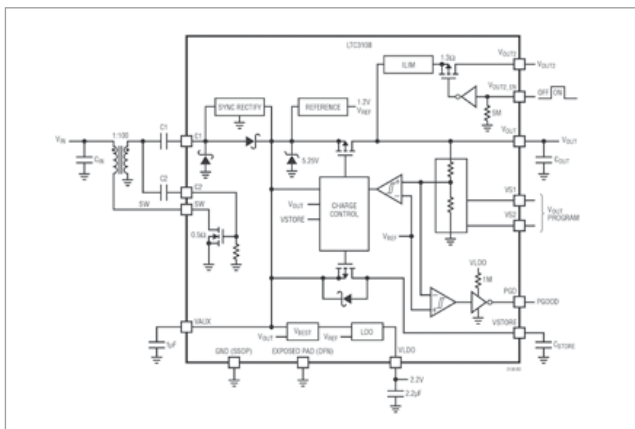


Figure 1. LTC3108 block diagram.

LTC3588-1

The LTC3588-a can obtain energy from a piezo element, but also, for example, from two metal plates positioned close to a fluorescent light tube. The datasheet offers additional options. A piezo element can generate a relatively high voltage, this is internally limited by the IC to 20 V. This voltage is then reduced to an adjustable voltage by a buck/step-down converter.

The prescribed piezo-element in the datasheet is very expensive, but perhaps the IC will also work in combination with a piezo buzzer? If so, is it possible to obtain sufficient energy? This is an invitation for some experimenting!

LTC3588-1 Datasheet
<http://cds.linear.com/docs/Datasheet/35881fa.pdf>

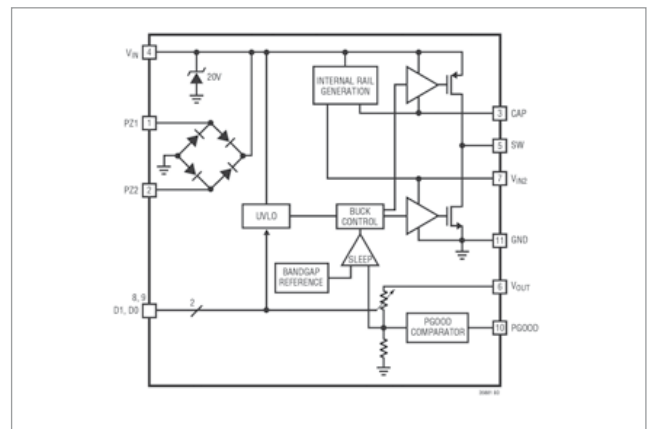


Figure 3. LTC3588-1 block diagram.

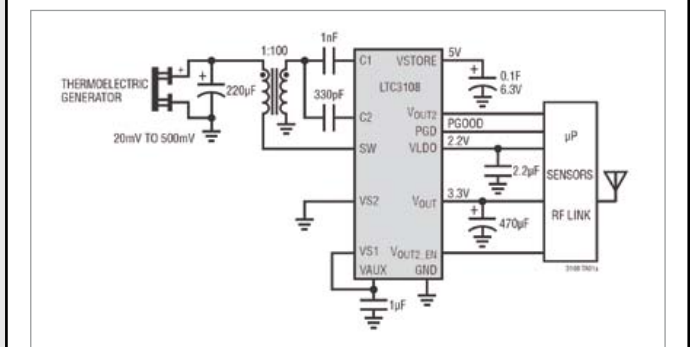


Figure 2. Application example using a Peltier element.

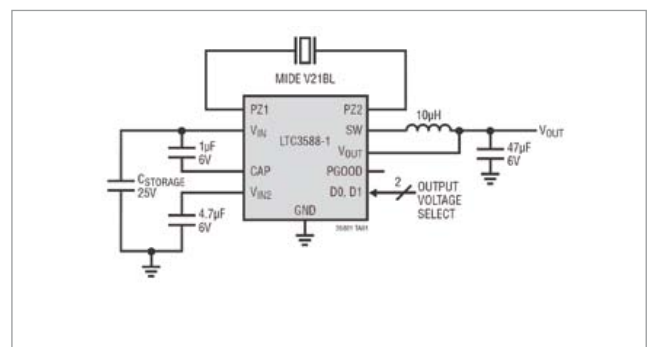


Figure 4. Application example using a piezo element.

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!



Vote for your Favorite Proposal

1 vote		Horloge du bureau à tubes Numitron et composants classiques Connaissez-vous les "Numitrons"? Ces petits tubes incandescents à 7 segments sont utilisés ici pour réaliser une horloge de bureau avec un...
0 votes		Numitron desktop clock - Using only classical components Do you know "Numitrons"? Those incandescent 7 segments bulbs are used here to make a 6 digits clock with a very nice orange glow: I used...
3 votes		ATX to bench top lab power supply adapter This open-frame design can optionally be mounted in an enclosure of the user's choice. There are many "how-to"s out on the web encouraging people to...
5 votes		Carte de prototypage rapide pour circuits analogiques linéaires Cette carte permet de tester rapidement des circuits à base d'amplificateurs opérationnels (filtres, oscillateurs ...). Le circuit imprimé...
2 votes		efficient solenoid valve We have in our house a reverse osmosis water filter for filtering the tap-water up to drinking water quality. The filter fills a little reservoir of...
4 votes		Displayless WLAN Speaker with Web User Interface WLAN Speaker for everywhere usage. Maybe a the top of a shelf. Configure it with a Web-Browser or a Smartphone-App to set the Web-Radio Station, the...



Arduino on Course (2)

Part 2: read (awfully) remote sensors using GSM / GPRS

This month we'll explore the possibilities offered by GSM and GPRS communication for your projects. We will link concepts like the Internet of Things (IoT) or the Machine To Machine (M2M) business in order to understand what you can achieve all by yourself in a short time.

By David Cuartielles (Spain)

Shopping list

In terms of hardware, here's what you need to be able to replicate the experiment discussed in this installment:

- Arduino Uno board;
- Arduino GSM shield (made in collaboration with Telefonica I+D);
- a SIM card for the GSM/GPRS network;
- a cellphone ('mobile') with its own SIM card;
- TinkerKit shield;
- TinkerKit LED module (x3);
- TinkerKit Slider module.

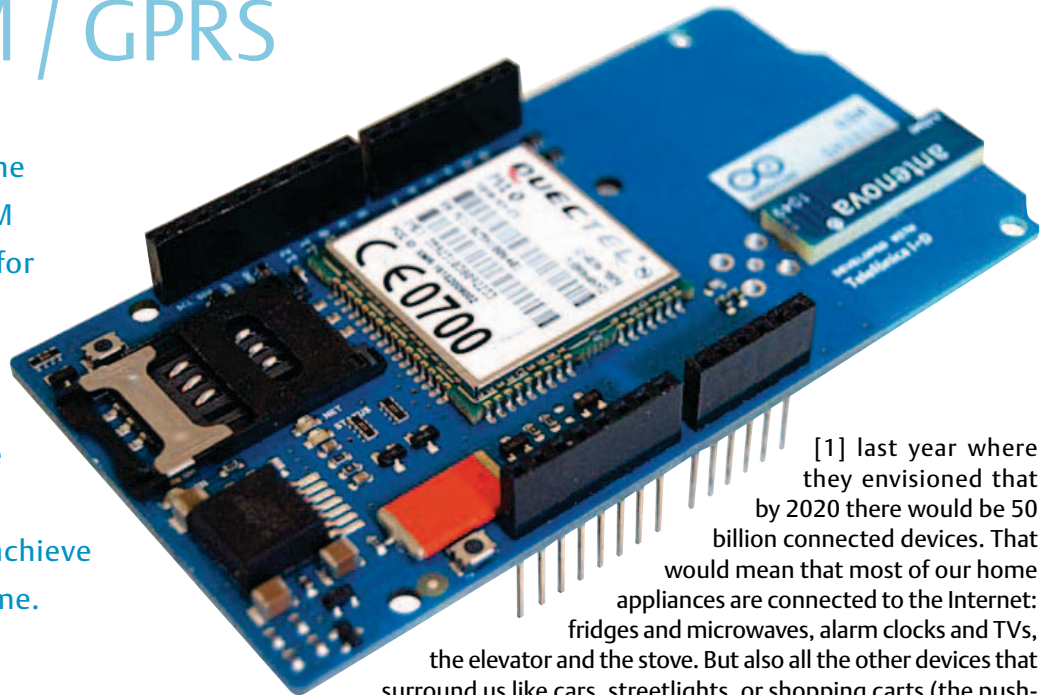
The last three items in the list can be substituted by a breadboard and a combination of various components. However, to keep the focus on the operation of the GSM library for Arduino, I have opted to use pre-assembled parts.

Now, before you read on, I strongly recommend you download all the code used this month. Due to the length of the example programs, only excerpts are printed here.

Introduction to M2M and IoT

Putting it in easy terms, IoT is a computing paradigm that is being explored lately by many research laboratories and universities around the world. It tries to bring to life the idea that everything can be connected to the Internet: your microwave and your car, passing by your door lock and your bike at the gym. All those objects should be able of talking to each other in some way and exchange data to offer people better ways to handle different situations.

Ericsson, the Swedish communication giant, presented a memo



[1] last year where they envisioned that by 2020 there would be 50 billion connected devices. That would mean that most of our home appliances are connected to the Internet: fridges and microwaves, alarm clocks and TVs, the elevator and the stove. But also all the other devices that surround us like cars, streetlights, or shopping carts (the pushable type, not the icon or applet).

M2M (machine-to-machine) is concerned with the connectivity between devices using wireless, wired or hybrid communication. In a sense, M2M makes IoT (Internet-of-things) possible as it offers the infrastructure for devices to gather data and transmit it through a network to a remote location. For some people M2M is a synonym of telemetry in whatever form, but in recent years the concept has mutated to refer mostly to wireless communication and to be more specific it is mostly referring to the use of cellular (mobile) communication for remotely controlling devices.

For the sake of simplicity, when talking about the IoT I mean the connection of devices to the Internet, while M2M will be a way to make this possible through the cellular telephony network. You will find many articles and videos on the Internet that follow the same approach, the reason being that the most extended network right now is the one called GSM (global system for mobile communications).

I assume that you are reading this article because you plan on making a connected device. Imagine that you create an object that's going to connect to the Internet and send data back to you from anywhere in the world — what, do you think, is the best way to do so? The answer is obvious: cellular telephony. GSM is a great alternative as it allows you contact your device almost anywhere in the world knowing that it will work, basically like at home.

The Arduino - GSM/GPRS Shield

From Arduino we want to simplify as much as possible the way you build prototypes and learn about technology. Therefore we have spent quite some time thinking about how to help you getting hooked to this trend of IoT in many ways. One of them is

through M2M and for that we teamed up with Telefonica's *Physical Internet Laboratory*.

We have been collaborating in making an official Arduino shield that easily handles GSM/GPRS (also known as 1G and 2G) communication. We also wrote a library that simplifies the process of texting (SMS transmission), setting up calls or publishing some data on a website. The design is open source and the software can easily be ported to support other shields (with potentially other series of AT-Commands; more about this later). Finally, the shield comes with a SIM card, courtesy of Telefonica, that will offer you the possibility to hook up to send/receive text (SMS), set/get voice calls, and connect to the Internet through the GPRS network.

Note: This shield is **not** locked to the accompanying SIM; you could use any other GSM/GPRS SIM card available in your country. The SIM is optimized for data communication; if you wanted to make something controlled over Text (SMS), you should consider a SIM card operating on your preferred network. If you were about to use the card coming with the board, you should follow the online activation instructions that come with it.

Controlling a modem - AT commands

Modems are the devices behind the expansion of the Internet. In order to automate the way they would operate, a company called Hayes [2] invented in 1981 what would later be called the AT-Command Set. The idea behind it all is for modems to change between Command and Data operation modes. The Command mode would allow performing operations on the modem itself like calling a certain number, changing the baud rate (the speed at which the data would be transferred), etc.

The Hayes Command Set became a de-facto standard for the way devices should communicate over a serial port. That standard is still operating inside many devices. When in Command mode, devices start the communication sending the string "AT" which stands for "attention". That would be followed by other strings that would translate in different operations to be performed at the other side. On the other hand, when in Data mode, the data will just be proxied to/from the Internet through the modem.

The GSM/GPRS shield has a radio modem manufactured by Quectel [3] that can be operated using a series of AT commands. Most of those commands are specific to the modem on the board, but the library that controls the shield has been written in a way that makes very easy to port it for other modems made by any manufacturer. Also, the library hides all the complexity behind this mode of operation. Therefore, we are not going to refer to them in the rest of the article. However, if you happened to use the GSM library in debug mode, you would be prompted with many of those commands.

Installing the GSM Library for Arduino

At the time of writing this article the Arduino IDE had reached revision 1.0.1 and did not include the GSM library we're about to discuss. Thus, when checking the menu 'Sketch / Import Library' if you don't see the library named 'GSM3' in the list, you will have to down-

load it from the official Arduino website hosting this project [4] and install it as explained under the 'Library' link. 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 [4] directly there. After that, you should restart the Arduino IDE and the library will then show up in the above mentioned menu. 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 modem is working properly;
- text, i.e. send/receive SMS;
- place/get phonecalls (voice);
- open a TCP/IP connection and exchange data over the Internet, using GPRS;
- post/collect sensor information to/from Cosm (an online data service);
- collect and show twitter messages.

The way to access the examples is simple: just use the menu to navigate through "File / Examples / GSM3".

Checking your status

Likely, the first thing you want to do is checking whether your board is working properly and whether your SIM card can be used by the shield for you to experiment with. To do so, open the example under the menu: "File / Examples / GSM3 / Tools / TestModem".

```
// libraries
#include <GSM3ShieldV1ModemVerification.h>

// modem verification object
GSM3ShieldV1ModemVerification modemTest;

// IMEI variable
String IMEI = "";

// serial monitor result messages
String oktext = "OK";
String errortext = "ERROR";

void setup()
{
  // initialize serial communications
  Serial.begin(9600);

  // start modem test (reset and check response)
  Serial.print("Starting modem test...");
  modemTest.begin();
  Serial.println(oktext);
}
```

```

void loop()
{
  // get modem IMEI
  Serial.print("Checking IMEI...");
  IMEI = modemTest.getIMEI();

  // check IMEI responded
  if(IMEI != NULL)
  {
    Serial.println(oktext);
    // show IMEI in serial monitor
    Serial.println("Modem's IMEI: " + IMEI);
    // reset modem for check booting
    Serial.print("Reseting modem...");
    modemTest.begin();
    // get and check IMEI one more time
    if(modemTest.getIMEI() != NULL)
    {
      Serial.println(oktext);
      Serial.println("TEST COMPLETE!");
    }
  }
  else
  {
    Serial.println(errortext);
  }
}
else
{
  Serial.println(errortext);
}
while(true);
}

```

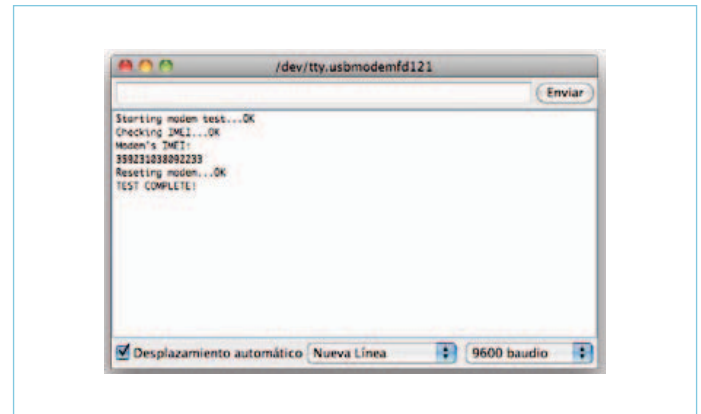


Figure 1. Positive answer from the TestModem example using the Serial Monitor.

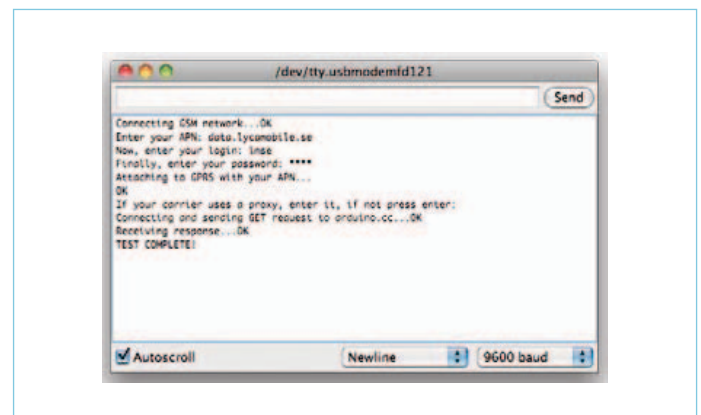


Figure 2. Configure your GPRS connection using the Serial Monitor.

This code example allows you to check whether Arduino can detect your radio modem. It also opens a connection to the radio modem on your GSM shield and attempts to read the IMEI (International Mobile Equipment Identity) from it – that’s a unique number for every device capable of connecting to the GSM/GPRS/3G/4G network. For example, your cell phone identifies itself via the IMEI. Every device on the GSM network has a unique ID. If everything goes fine, after uploading the example on your Arduino Uno, and opening the Serial Port Monitor, you should see something like in **Figure 1**. If your modem was working correctly, the next thing is to check whether you can connect to the GPRS network. To do so, we will open the example: *File / Examples / GSM3 / Tools / TestGPRS*. I will not show the code here, as it is very similar to the previous one. This example is used for testing that your SIM card allows connecting to the GPRS network (the one provided with the shield does), but also to test your settings.

Note: this shield is using GSM/GPRS technology and not 3G. 3G and GSM SIM cards look the same, but remember that some of the modern operators do not offer 2G (the way we call GPRS) services. Connecting to the GPRS network calls for a tad more configuring than just texting (sending an SMS) or placing a call. There are four parameters that need to be configured:

- APN: acronym for Access Point Name or the name of the domain the shield will connect through to reach the Internet. It is specific for the operator providing you with connectivity through the shield;

- login: some operators require a login;
- password: some operators require a password;
- proxy: this is the address to a server that will channel all your communication inside the operator’s network. Most likely it won’t be used.

The way this example works is by asking you for the information the shield needs to connect to the network.

Tip: remember activating the option ‘Newline’ in the dropdown menu at the bottom of your Serial Port monitor. In this way, when clicking on ‘Send’ on the monitor, the system will add an end-of-line (EOL) character that is needed for the modem to know the information package came to an end.

If everything goes fine, you will see a window like in **Figure 2**. Those messages mean that your shield can connect to the Internet using your SIM. Please note that even if you couldn’t connect to the Internet, it may still be possible for you to text-out, get text (send/receive SMS) and set up calls.

Texting (sending an SMS)

Texting or receiving text (sending or receiving SMS) with the GSM shield couldn’t be any easier. The following lines of code could be part of a program sending a string to a certain number.

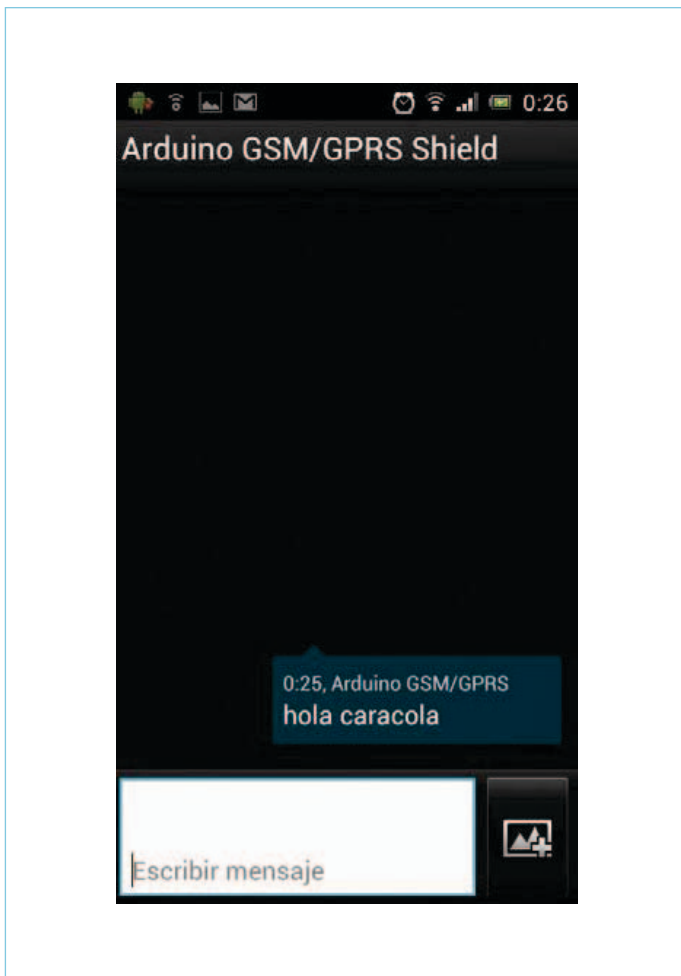


Figure 3. Screenshot of my phone after I was texted.

```
// send the message
sms.beginSMS("0755442200");
sms.print("hola caracola");
sms.endSMS();
```

Note that I am using a fictitious cell phone number! And the result on the screen of my phone would be like shown in **Figure 3**. The library comes with examples covering both sending and receiving. For example, the code at *File / Examples / GSM3 / Tools / SendSMS* will be sending an SMS (texting out) from the board to a phone via interaction through the Serial Port Monitor. It will first ask you for the phone number to text to (SMS) and then for the string to send. Try it out!

Remotely controlling a device via Text (SMS)

Controlling a device remotely implies receiving messages, parsing the data, and operating actuators depending on the different commands. In this case we are going to build a quick prototype including three LEDs: one red, one green and one blue. You can imagine that instead of using LEDs we could be attaching relays, and we could be controlling any kind of device on distance. The program is going to read the first character sent in the SMS and turn on the corresponding light: R for red, G for Green and B for blue. This pro-

gram is, as you can see, very simple. It doesn't execute huge parsing operations; it simply reads the first character in the text message (SMS) and decides what to do. After that, the text message (SMS) will be erased from the memory using the command `sms.flush()`. The following is a part of the example `DecodeSMS` you can find at the download link [5] for this article.

```
// If there are any SMSs available()
if (sms.available())
{
  Serial.println("Message received from:");

  // Get remote number
  sms.remoteNumber(remoteNumber, 20);
  Serial.println(remoteNumber);

  // This is just an example of message disposal
  // Messages starting with # should be discarded
  if (sms.peek() != '#')
  {
    Serial.println("Discarded SMS");
    sms.flush();
  }

  // Read message bytes and print them
  int count = 0;
  while (c = sms.read()) {
    if (!count) {
      digitalWrite(pinR, LOW);
      digitalWrite(pinG, LOW);
      digitalWrite(pinB, LOW);

      switch (c) {
        case 'R':
          digitalWrite(pinR, HIGH);
          break;
        case 'G':
          digitalWrite(pinG, HIGH);
          break;
        case 'B':
          digitalWrite(pinB, HIGH);
          break;
      }
    }
    count++;
  }

  Serial.println("\nEND OF MESSAGE");

  // delete message from modem memory
  sms.flush();
  Serial.println("MESSAGE DELETED");
}
```

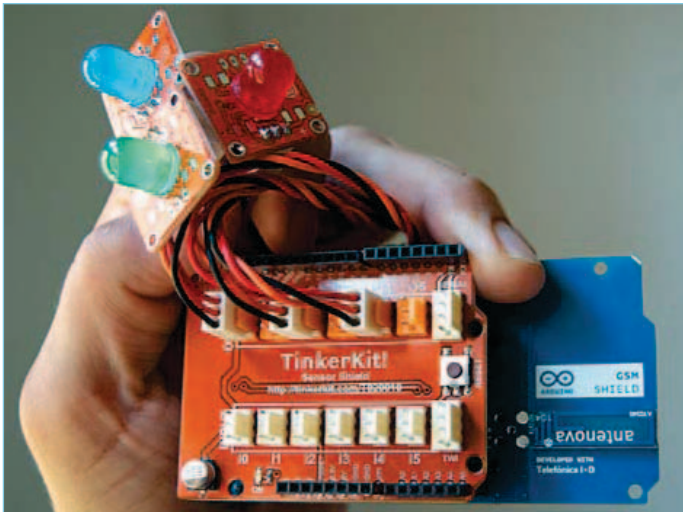


Figure 4. All in hand: Arduino Uno + GSM/GPRS shield + TinkerKit shield + TinkerKit LED.

The actual circuit boards are shown in **Figure 4**.

Device working over GPRS

Let's now see how to use an analog sensor (a slider in this case) and send data from it to a website. On the server side I am using Apache and a very simple PHP script that will be taking the data sent by the board and will update a part of the script containing plain HTML. This is what I like to call a simple 'online memory cell'. It is a PHP script that can take the data and display it. It will be possible to access that same page using a browser to see the data changing over time. You will need to use the same APN/login/pass combination you used in the previous GPRS example for this code to work properly. Here's how to send an analog value to a server every 10 seconds:

```
void loop()
{
  client.connect(server, 80);
  Serial.println("sending data...");
  client.print("GET ");
  client.print(path);
  client.print(query);
  client.print(analogRead(A0)); // take the value
                               // on A0 and send it

  client.println(" HTTP/1.1");
  client.print("HOST: ");
  client.println(server);
  client.println();
  client.stop();
  delay(10000);
}
```

You need to have a server to run the php code that will be reading from your Arduino board. The server needs to be publicly accessible as you want your Arduino board to post information to it and to read that data using a browser, a phone or any other web-enabled device. *Important:* keep in mind that this code is just a proof of concept, it is not safe in any way and you should probably disable the PHP script once you are done trying out the example. The only thing left to do is uncompress the code of the PHP mem-

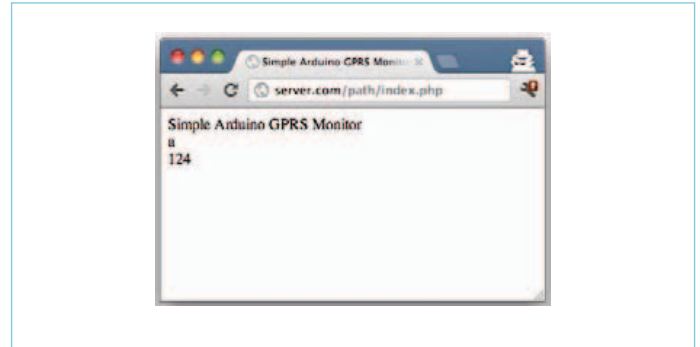


Figure 5. Information being displayed on a browser as sent from the GSM shield.

ory cell in the given file [5] at the root folder, to your http server. Make sure the file has read-write-execute permissions to allow it to rewrite itself. You can then type the URL to your server (for example: <http://server.com/path/index.php>) in a browser to see the information update every 10 seconds. Like in **Figure 5**.

Closing words

I hope you found this brief introduction to the use of GSM/GPRS technology in your projects both educational and fun. I want to highlight how relevant it is that we, as makers, can benefit from the existence of an omnipresent wireless network. If you don't need very data-hungry devices, you could start controlling lots of things over the cellphone network. As long as there is a way to power up your board, you can gather and process data from almost anywhere. (120506)

Internet Links

- [1] **Ericsson's memo on 50 billion connected devices:**
www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf
- [2] **on the AT-Command set:**
http://en.wikipedia.org/wiki/Hayes_command_set
- [3] **Quectel AT-Command description:**
<http://datasphere.eu/en?t=/documentManager/sfdoc.file.supply&fileID=1285079825955>
- [4] **Arduino website hosting information about the GSM shield:**
<http://labs.arduino.cc/GPRS/Index>
- [5] **Compressed file with all the examples and images:**
www.elektor.com/120506

Acknowledgements

My sincere thanks are due to the whole Physical Internet Laboratory team at Telefonica Research and Development, for their support in making this new Arduino shield possible. Special thanks go out to F. J. Zorzano for being there debugging code until late many nights.



10th International System-on-Chip (SoC) Conference, Exhibit & Workshops

October 24 & 25, 2012 – Hilton Irvine/Orange County Airport

www.SoCconference.com

EARLY BIRD REGISTRATION IS NOW OPEN!

Don't Miss Out!

Platinum Sponsors



- Table-Top Exhibit (Free Passes)
- Memory Trends & Technologies
- Analog & Mixed-Signal Designs
- Green Chips
- Sub 20nm Design Challenges
- 3-D ICs
- Heterogeneous Multi-Core Designs
- Multicore Software Development
- SoC Design & Verification
- Many Door Prizes

- SOI vs. CMOS
- FPGAs –Trends & Designs
- Multicore SoC Platforms
- Memristors
- Network-on-Chips (NoCs)
- Informative Panels
- Mixed-Signal SoCs
- IEEE Student Design Contest
- Networking Opportunities
- And Much More. . .

Keynote Speakers



Altera
Dr. Mike Peng Li,
Fellow,
Altera Corporation.



HP
Dr. Janice Nickel,
Cognitive Systems Laboratory
at Hewlett Packard
Laboratories.



Microsemi
Jim Aralis,
Chief Technology Officer
(CTO),
and Vice President of R&D.



JPL
Dr. Reza Ghaffarian,
Jet Propulsion Laboratory,
California Institute of
Technology.



UCI
Dr. Payam Heydari,
University of California
Irvine.

**Promote Your Company, Technology, Products & Services at
The Most Informative, Targeted & Affordably Priced
Chip Design Conference, Exhibit & Workshops of the Year!**

Selected Participating Companies & Universities



For More Information or Questions, Please Contact the SoC Conference Organizing Committee at:
SoC@SoCconference.com or (949) 851-1714

www.SavantCompany.com & www.SoCconference.com

Radiation Meter Reloaded

Simple modifications open up new possibilities

By Burkhard Kainka (Germany)

The radiation meter published in the November 2011 edition of *Elektor* is currently being used by many readers, mainly in making long-term measurements and in investigating weakly radioactive samples. There are several ways in which we can modify and improve the design, and we look here at a few examples.



The 'Improved Radiation Meter' was published in *Elektor* in November 2011 [3]. The device is available as a kit of parts [2], the ATmega88 microcontroller being equipped with a bootloader that makes it easy to program your own code into it. Some applications will require just a small change to the code in the basic program, and below we will look at a couple of examples of this. Other applications will require firing up the soldering iron to make more permanent changes: we will look at some of the possible hardware modifications first.

Calibration

The following question often crops up in our readers' forum: can the radiation meter be calibrated in standard units such as microsievert per hour ($\mu\text{Sv/h}$)? This turns out not to be so easy, as several variables enter into the equation. The most significant of these is the screening used.

Experience with the radiation meter shows that it is not easy to screen the sensor completely from light without simultaneously also blocking the radiation we are interested in to a significant extent. Two options are open to us: we could put the sensor and the sample being studied together in

a metal can connected to ground, or we could wrap the whole sensor board in aluminum foil, which again must be grounded. Some readers have experienced problems with the latter approach arising from a microphony effect: the sensor forms a kind of condenser microphone with the foil shield. The effect can be mitigated by gluing the foil directly to the sensor. Self-adhesive copper foil, 0.035 mm thick, has also been found to work well in practice: it also has the advantage that it is easier to solder the ground connection to it.

We tried an experiment using a BPW34 sensor with a copper foil shield. The radiation source was a thorium gas lantern mantle in its paper envelope. Because this source has a large area the radiation level at close quarters is approximately uniform, which means that the readings obtained are largely independent of the distance between source and sensor. The *Elektor* radiation meter gave approximately 40 pulses per minute at a distance of 1 cm. For comparison we used a professional Geiger counter with a tube sensitive to beta and gamma radiation, again at a distance of 1 cm. This gave a reading of 800 pulses per minute, corresponding to 20 $\mu\text{Sv/h}$.

This gives us our first unofficial calibration result:

$$1 \text{ pulse/min} = 0.5 \mu\text{Sv/h}$$

In principle the unit is capable of measuring the level of background radiation if readings are taken over a sufficiently long period. Over very long periods it is possible to see a clear change in readings with variations in solar activity. The somewhat longer measurement period required than with a Geiger counter does of course make some applications less practical: for example, if you try to measure the radiation from a salad you may well be lucky enough to discover that it is safe to eat, but by the time you know that your rocket will probably have wilted!

The device is better suited to investigating small, weakly radioactive, samples than to analyzing the background radiation. Here the small sensor sizes gives the device an advantage over the Geiger counter, as it makes it easier to discriminate between radiation from the sample and the background.

AC coupling

There is a further technique we can use to

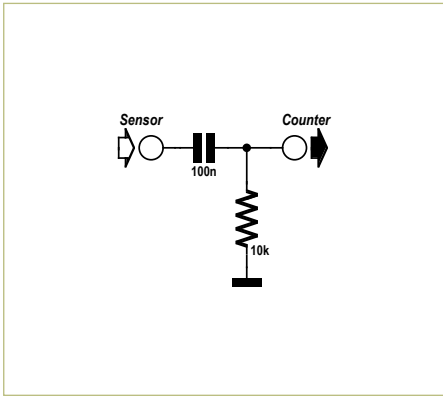


Figure 1. AC coupling.

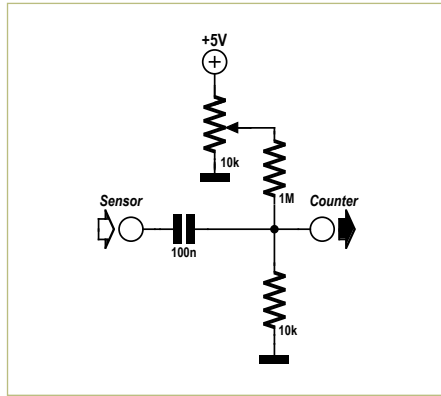


Figure 2. Setting the trigger threshold.

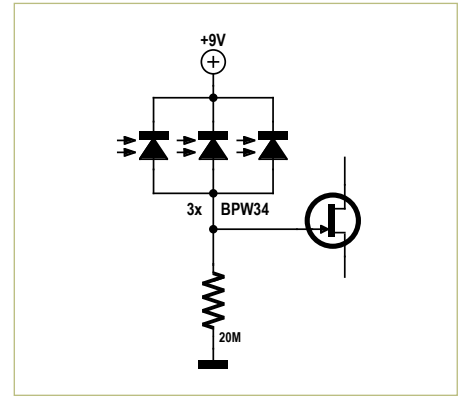


Figure 3. Three BPW34s in parallel.

reduce the sensitivity to ambient light: we can add AC coupling between the sensor board and the counter. A 100 nF capacitor and a 10 kΩ resistor will do the job (Figure 1). The circuit operates as a high-pass filter with a corner frequency of 160 Hz. It now does much less harm if light should fall on the sensor and shift its operating point. Flickering from artificial incandescent lighting at a frequency of 100 Hz is also somewhat attenuated. However, it does little to reduce the effect of fluorescent lights as these produce sharper flashes of light. Nevertheless, the filter modification does make the circuit more stable generally and considerably less sensitive to light.

If desired, a potentiometer can be added to the circuit to allow adjustment of the counter trigger threshold (Figure 2). Although this facility was already available in the PC software accompanying the design, some readers use the unit to make measure-

ments ‘in the field’ without a PC attached: if you are taking readings while scrambling around in a quarry, you will probably prefer not to risk lugging an expensive laptop about as well. The suggested circuit shifts the average threshold by up to 50 mV. By default the software sets the trigger threshold to 10, which corresponds to approximately 50 mV. At start-up the microcontroller measures the average voltage at its input and adds this value to the threshold: any pulse with amplitude exceeding this value will be counted. With the new AC coupling circuit the microcontroller will measure the average noise level on the input, obtaining, for example, a value of 5. The trigger threshold will then be 15. With the addition of the potentiometer the quiescent level can be increased to as much as 50 mV. At start-up the potentiometer should be set to its lowest point. Then the threshold can be adjusted carefully towards the noise floor to determine the point where sensitivity is greatest. A press of the ‘Zero’ button will then start a fresh series of readings with the new threshold setting.

Sensors and results

In its standard configuration the device uses a single type BPW34 PIN diode as its sensor. However, it is possible to wire several BPW34s in parallel (see Figure 3). The effect of this is to produce meaningful readings in a shorter time. It is possible either simply to count the pulses or to use the more sophisticated program that analyses the energy spectrum.

Before carrying out a measurement it is a good idea to do a calibration measurement to estimate the background radiation level. Figure 4 shows such a calibration measurement with three BPW34s wired in parallel. The trigger threshold used was 5. Forty pulses of varying energies were detected over the course of one hour. These results can be subtracted from the subsequent measurement to improve accuracy when using weakly radioactive samples.

Potassium chloride (or more accurately the potassium-40 in it) exhibits weak beta activity. Perhaps easier to obtain is potassium carbonate (or ‘carbonate of potash’), which is sometimes used in baking. A sample of this substance was placed in a small plastic bag next to the sensor. Figure 5 shows the results, taken over one hour. A total of 251 pulses were counted, some six times the background level.

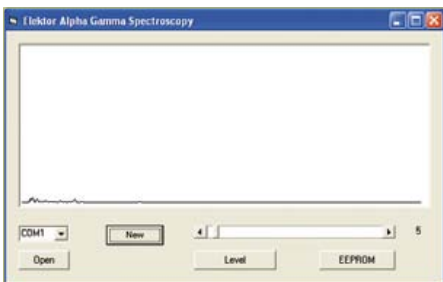


Figure 4. Calibration measurement using three BPW34s in parallel.

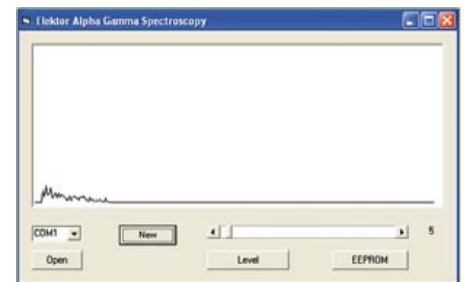


Figure 5. Radiation from potassium carbonate.

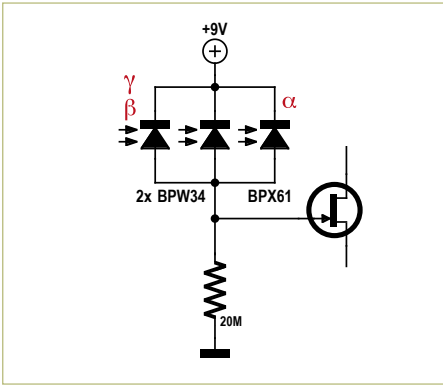


Figure 6. A combination of different sensors.

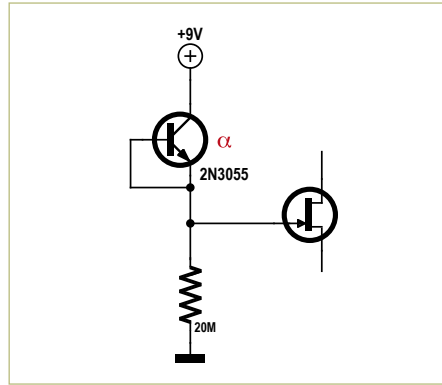


Figure 7. Using a power transistor as an alpha particle sensor.

Alkaline batteries do not normally leak; however, when they do, the concentrated potassium hydroxide in them reacts with the CO₂ in the atmosphere to form potassium carbonate. This is seen as a white residue where the leak has occurred. This residue should be a weak radioactive source, and indeed this can be confirmed using the radiation meter.

With the sensor set up for maximum sensitivity, including for low-energy pulses, we can use the radiation meter with samples of even lower activity. This can be seen in an experiment with charcoal ash from the barbecue, which contains potassium carbonate. We measured a radiation level of approximately 150 pulses per hour from a sample. We can even find radioactiv-

ity in cigarette ash: in one experiment we obtained 80 pulses per hour, double the background level. Again, the main source of the radiation is probably potassium, although polonium, a product of uranium decay, may also play a part as it is accumulated in tobacco plants.

Alpha decay

Using an exposed BPX61 photodiode lets us also detect alpha particles. In fact, we can use both types of diode together: a BPW34 to detect beta and gamma radiation, and a BPX61 for alpha particles (see **Figure 6**).

To test this combination we first measure the weak radiation from a sample of potash or potassium carbonate and set the trigger threshold appropriately. Once this has

been done successfully, we can move on to making the more difficult measurements, including on various building materials whose level of activity may be of interest. Alpha particles are distinguished by their higher energy: we can often find noticeable levels of radiation in plasterboard, because the gypsum used in its manufacture is made by the desulphurization of flue gases from coal-fired power stations and, in turn, the coal contains a small amount of uranium and its various decay products.

The 2N3055 as an alpha particle detector

When alpha particles hit a silicon junction a relatively strong signal is produced. That is true not just for photodiodes, but for any type of diode or transistor. Under normal circumstances alpha particles do not get as far as the junction as they are absorbed by the devices package or shielding. However, we can get around this in the case of certain power transistors in a TO-3 package: if we remove the metal cap the silicon chip itself is revealed, with its relatively large surface area exposed to the elements.

This is usually the case for the famous 2N3055; for other devices there is sometimes an additional protective layer, and so sometimes a little luck is needed to find one that is suitable. A fairly recently-produced transistor from ST was found to work well (see **Figure 7**). The chip is completely exposed and makes an excellent alpha particle sensor. Beta and gamma radiation, however, go completely undetected. **Figure 8** shows the result with a sample of pitchblende (uranite). When tested with an americium-241 source from a smoke detector more pulses were produced than the counter could cope with.

Serial interface

The counter board includes a three-pin connector which gives access to the serial port on the microcontroller, the RXD and TXD signals being at TTL levels. The connector is compatible with the Elektor breakout board ('BOB') [1], based on the FT232R, and so allows connection to a PC over USB using a virtual serial port.

A very simple extension allows the device to be connected to a standard RS-232 inter-

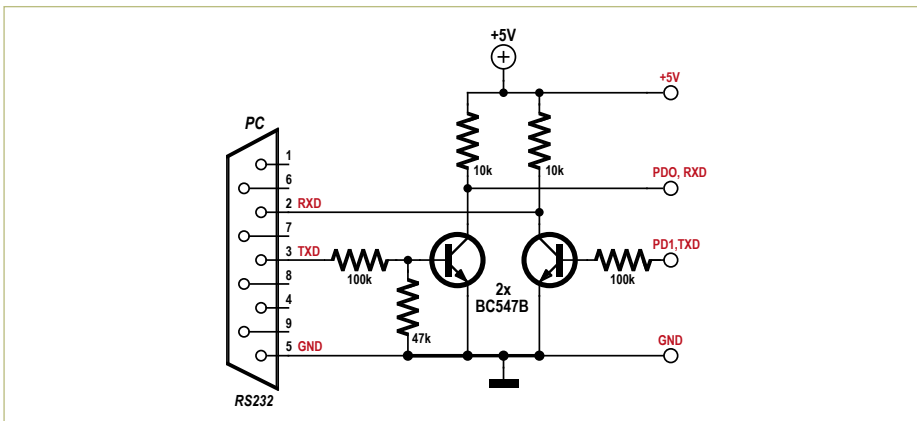


Figure 9. RS-232 level shifter.

face. Two NPN transistors do the job of shifting the voltage levels and inverting the signals (see **Figure 9**). With this circuit you can connect the radiation meter to a PC with an ordinary serial port or to a USB-to-RS-232 adaptor.

The PC-based program can be installed to provide the ability to plot energy spectra and to adjust the trigger threshold. It also provides the additional useful function of uploading program updates (or even completely new programs) into the microcontroller over the serial port.

Program upload

Suppose we would like to make a change to the way the counter works. The bootloader facility in the microcontroller makes this easy, and it even works with the 'Demo' version of BASCOM (**Figure 10**). First open the BASIC source code and then, after making any changes to it, compile it. Select the programmer to be used in BASCOM. In this case it is 'MCS Bootloader'. The COM port and baud rate (19200) also have to be set correctly (see **Figure 11**). Now start the programmer, for example by using the small green circuit board icon. The MCS Bootloader will now try to establish a connection with the microcontroller. The connection can only be established shortly after the microcontroller is powered

up or reset. So, switch the device off and then on again, and programming should commence (**Figure 12**).

When all the code has been uploaded the microcontroller immediately starts to execute the new program.

Software variants

In-depth BASIC programming skills are not needed to make small changes to the program and to upload the new code to the microcontroller. Here are a couple of examples.

1. Constant trigger threshold (Listing 1).

Suppose we always wanted to have the trigger threshold equal to 5. All that's required is to add the line 'L = 5' at the right point in the program. We can also get rid of the 'Counter' message at start-up by simply commenting out the relevant lines by adding a single quotation mark at the beginning. A further change is also shown in Listing 1, but commented out for now so it has no effect: the line 'Um = 0' sets the average level to zero, which is better suited to use with the AC coupling circuit shown in Figure 1.

2. Measuring the energy distribution (Listing 2)

Our second example shows how to count

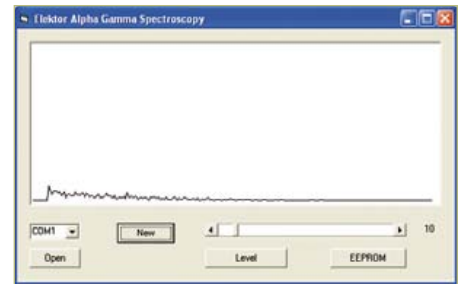


Figure 8. Measurement using an old faithful called 2N3055.

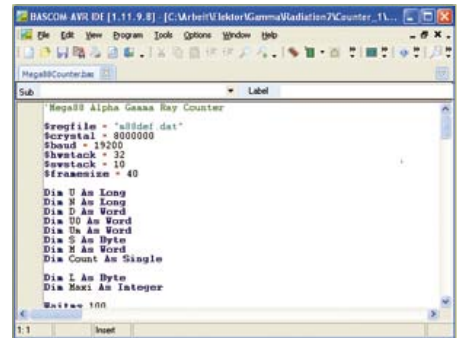


Figure 10. The BASCOM development environment.

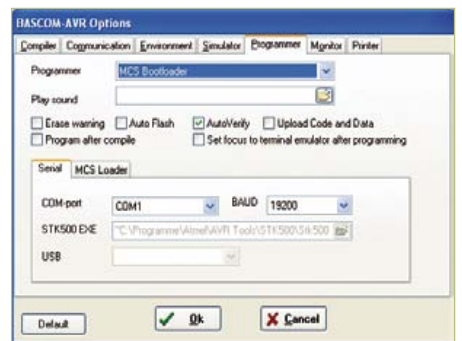


Figure 11. Selecting the programmer.

```

Listing 1: Changes in Counter_1

'Print "Counter"
Locate 1 , 1
'Lcd "Counter"
'Waitms 500
Readeeprom L , 1
If L = 255 Then L = 5
U = 0
For N = 1 To 1000
    D = Getadc(0)
    U = U + D
Next N
U = U / 1000
Um = U
'Um = 0
U0 = Um + L
N = 0
    
```

```

Listing 2: Changes in Counter_2

Do
    D = Getadc(0)
    If D > Maxi Then Maxi = D
Loop Until D < U0
Maxi = Maxi - Um
If Maxi > 255 Then Maxi = 255
N = N + 1
If Maxi > 40 Then N2 = N2 + 1
If Maxi > 80 Then N3 = N3 + 1

Locate 1 , 1
Lcd N
Lcd " "
Locate 1 , 8
Lcd N2
Locate 1 , 14
Lcd N3
    
```



Figure 12. Programming is successful!

pulses separately according to their energy. We will have three counters, one for all results (N), one for pulses with a level exceeding 40 (N2), and one for the highest-amplitude pulses, with a level exceeding 80 (N3). All three counts will be displayed on the top line of the display. The values give a broad indication of the energy distribution without the need to connect a PC to plot a graph. The fact that we can discriminate energy levels in this way is a significant advantage of the semiconductor sensor over a conventional Geiger counter.

Other substances to examine

The question often arises of what substances can be used to test the operation of the device. In the original article we suggested using a sample of uranite, a luminous watch, and potassium chloride. Below we give a few more ideas. Uranium glass is still made. It is used in decorative fluorescent glass beads. Type WT20 tungsten inert gas (TIG) welding electrodes are made of tungsten with the addition of a 2% thorium oxide component. The electrodes are weak alpha, beta and gamma sources.

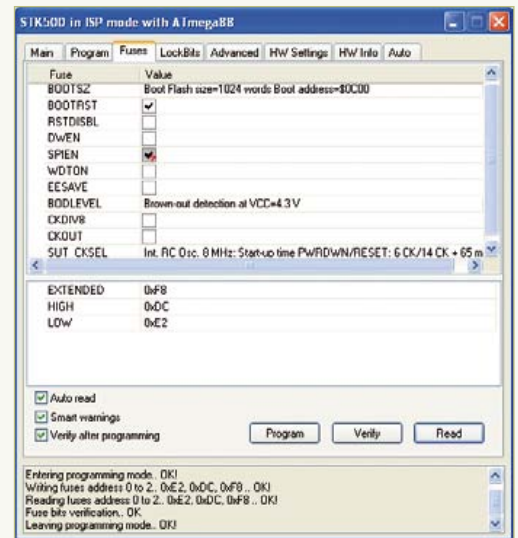
Tungsten with added thorium is also used in the directly-heated cathodes of large radio transmitter valves, although admittedly such devices are not exactly found on every street corner. However, the same material is used in the magnetron in a domestic microwave oven: a salvaged magnetron cathode makes an effective test source.

Thorium salts were used in the mantles for gas and paraffin lamps in the past, although modern mantles use different substances. If you can manage to get hold of an old mantle, you will find it makes a particularly powerful test source, including of high-energy gamma rays.

Ionization smoke detectors use a strong (perhaps 33 kBq) alpha source. Such detectors are common in the USA and in the UK but much less popular in mainland Europe. The gamma rays produced by the substance are relatively weak, and so a BPW34 only produces about 30 pulses per minute. However, the alpha particles also produced are easily detected by an exposed BPX61, giving many thousands of pulses per minute.

Program the bootloader yourself

If you programmed the microcontroller yourself you probably did it without the help of the bootloader. It is, however, possible to switch to using the bootloader. The program can be found along with the BASCOM development system examples, and requires a little adjustment to make it compatible with the ATmega88. The modified version is provided with the download files accompanying this article [2]. The compiled file BootLoader88.hex needs to be programmed into the microcontroller using (for example) the STK500 programmer, which will overwrite the counter program already stored in the device. (We will reload it later using the newly-programmed bootloader.) It is important not to use the hardware programmer from this point on, as otherwise the bootloader code will be overwritten. Note also that the configuration fuses have to be correctly set (see Figure): in particular the boot area must be enabled and its size set to 1024 words. When everything has been programmed correctly, the counter program can be reloaded over the serial interface.



Let us look in a bit more detail at the bootstrap process. When the bootloader is called up from within the development environment, the PC sends a continuous stream of bytes with value 123 and waits for the microcontroller to reply with the same byte value. Once communication is established all the programming data are transferred over the serial port and then programmed into the microcontroller's flash memory.

The microcontroller sees things as follows: upon reset or when power is applied it first branches into the boot area, an action enabled by the fuse settings mentioned above. The program stored in this area is responsible for determining whether a bootloader operation is required. If not, it branches to address zero and starts the program stored there. If, on the other hand, it detects the byte value 123 on the serial port, it echoes the byte and then waits to receive further data, which it writes into program memory. When programming is complete the new program is executed from address zero.

As you can see, a couple of simple modifications to the radiation meter circuit open up a range of new possibilities. The kit [3] is easy to build as only leaded components are used. The *Elektor* BOB [1][2] is also recommended to simplify connection to a PC. (120381)

Internet Links

- [1] <http://www.elektor.com/BOB>
- [2] <http://www.elektor.com/120381>
- [3] <http://www.elektor.com/110538>

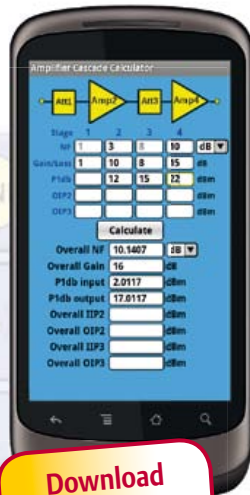
Elektor RF & Microwave Toolbox

➔ for Android

The RF & Microwave Toolbox contains 43 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

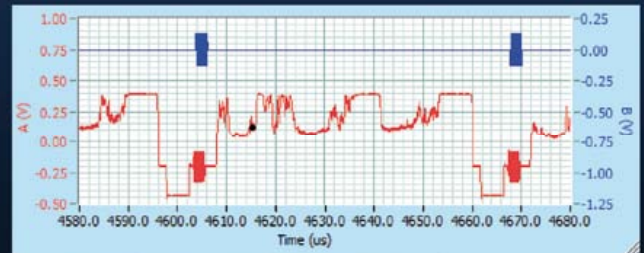
CS328A-XS



100MHz Mixed Signal Oscilloscope + Signal Generator

Maths + Matlab + Filters

What other scope has multi-line symbolic Maths including configurable filters and live export to Matlab and Excel ?



Example: use Maths to extract color burst from Video and send to Channel B

14 Bits 100 MSPS MSO



www.cleverscope.com

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

Electronics for Starters (8)

Audio Preamplifier

One of the most important applications for transistors is the amplification of audio signals. The device providing this function is the audio amplifier or audio frequency (AF) amplifier. This might be a microphone preamplifier or perhaps part of a radio circuit but the result is the same: something quiet is made louder!

by Burkhard Kainka (Germany)

The principle of a (preferably) linear amplifier is straightforward: an audio signal modulates (varies in volume and frequency) the small base current of a transistor, so that the amplified collector current produces corresponding audio signals of greater intensity. The most commonly employed amplifier circuit is the emitter circuit, the name implying that the emitter is connected to the common ground potential of the circuit. Whilst the base voltage and collector voltage can vary, the emitter voltage always remains constant, namely 0 V. In circuit diagrams we sometimes draw components with individual ground symbols and at other times a continuous ground connection, but the meaning is the same. Generally the ground connection is joined to the minus (negative) pole of the current source.

Our goal for an amplifier of this kind is to provide the greatest possible dynamic range before distortion occurs and under-

mines the satisfyingly faithful amplification achieved up till then. For this reason the quiescent collector current should be capped at just half the maximum possible current, with the collector resistor and supply voltage selected carefully to achieve this. In this way the current can be varied significantly in either direction to an equal extent without hitting limitations. All the same, achieving the correct quiescent current with a suitable base resistor is not entirely simple. If we take a look at **Figure 1**, we see here some optimal values for the components, assuming that the BC547B transistor happens to exhibit current amplification of exactly 300-fold. The base current amounts to

$$I_B = U / R = (9 \text{ V} - 0.6 \text{ V}) / 560 \text{ k}\Omega = 15 \mu\text{A}.$$

The collector current is then $I_C = I_B \times V = 15 \mu\text{A} \times 300 = 4.5 \text{ mA}$. The voltage dropped across the collector resistor is $U = I \times R = 4.5 \text{ mA} \times 1 \text{ k}\Omega = 4.5 \text{ V}$. The remainder of the 9 V, also 4.5 V as it happens, is developed between the emitter and collector.

This is the ideal case, in which the output can now be driven an equal extent in both directions. The voltage on the collector can therefore vary between 0 V and 9 V, meaning for example that a pure sinewave tone will be amplified at maximum output voltage without distortion. The voltage amplification is in this situation more than 100-fold.

However, if you build this circuit using a variety of transistors you will get a different result each time, because their current gain varies wildly. In the worst-case scenario, a transistor might have an amplification factor of 600, which is so great that the transistor is already switched hard-on to full saturation. Then you can forget about getting low-distortion output signals. Fortunately there is a perfect remedy: negative feedback.

Negative feedback

The simplest form of negative feedback occurs when the base resistor is connected not to the supply voltage but to the collec-

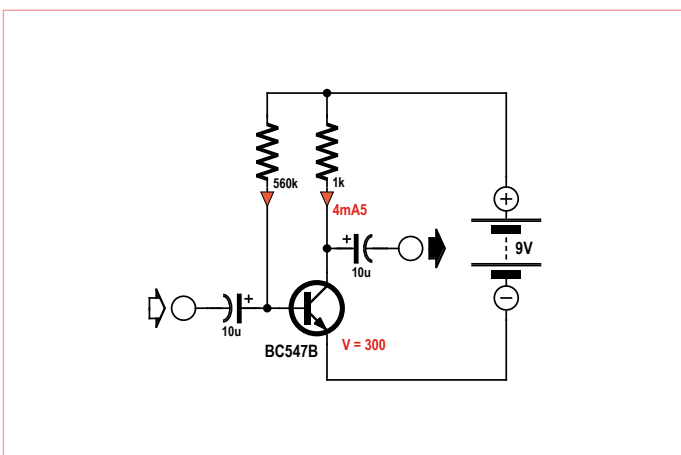


Figure 1. Setting the bias point for exactly 300-fold gain.

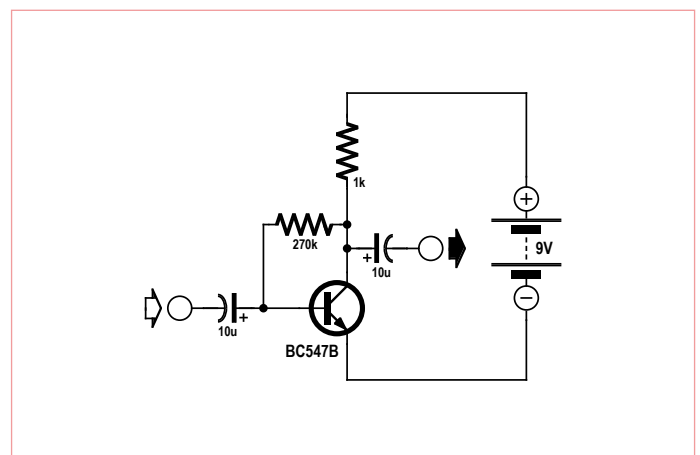
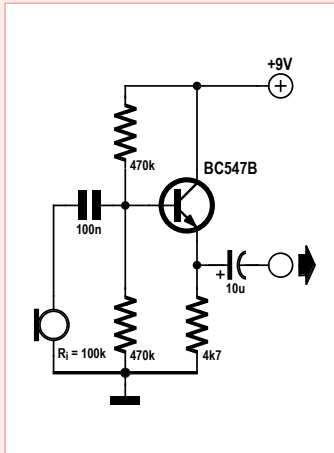


Figure 2. Setting the bias point with negative feedback.

The Emitter Follower



The emitter follower is also known as the common collector circuit, as the transistor operates with a constant collector voltage. The output of the amplifier is at the emitter. Every change of voltage on the input is reflected automatically at the output because even a very small variation in the base-emitter voltage is sufficient to alter the emitter current significantly. The emitter voltage

lies always around 0.6 V below the base voltage.

A change in the base voltage of 1 V is therefore reflected in one of almost 1 V in the emitter voltage as well. The precise value might be 0.99 V if a change in collector current required a variation in base-emitter voltage of 10 mV. The voltage amplification is thus almost unity (1), meaning the input voltage is not amplified even though the input current certainly is. The advantage of the emitter follower is its high input resistance or impedance. Whilst the input resistance of an emitter follower circuit is about 1 k Ω (according to its bias point), the collector circuit achieves 100 k Ω and more. For example, you can connect a crystal microphone or a crystal pick-up cartridge direct. This circuit also lets you use a simple piezo-ceramic transducer as a microphone (or for picking up sound waves in solid objects). It can be used like this for monitoring pulse beats for instance.

tor (**Figure 2**). The rule of thumb is this: $R_B = R_C \times V$, where V is selected for the middle range of current amplification expected. For 300-fold current amplification and a collector resistance of 1 k Ω we need to use a base resistor of around 300 k Ω .

This time our circuit reacts in a more measured manner to differing current gain situations. Greater gain leads to larger collector current and therefore to greater voltage drop across the collector resistor. The collector-emitter voltage falls at the same time and with it the voltage on the base resistor too, which in turn leads to a reduction in base current. The nett result is higher gain offset in part by smaller base current, so that the collector current rises less. We call this negative feedback because the increase in collector current is counteracted. In this situation the negative feedback leads to reduced amplification but with correspond-

ingly less distortion. Most importantly, you can use absolutely any NPN transistor in your junk box to build this circuit and it will work each and every time. Or expressed more scientifically, the circuit can tolerate a wide variation of samples.

Achieving a still more accurate setting of the bias point calls for more effort. For this purpose we provide the transistor with an emitter resistor (**Figure 3**). By providing the base with a voltage divider we can ensure a fixed voltage on the base, in which the current flowing through the voltage divider is around 10 times larger than the base current, thus avoiding any reaction arising from variations in base current. The emitter voltage stabilizes itself at a value in which the base-emitter voltage is below the base voltage. This means the emitter current remains stable and consequently the collector current too.

In the sample circuit a base voltage of 3 V is given. This determines the emitter voltage of 2.4 V. An emitter resistor of 1 k Ω produces an emitter current of 2.4 mA. A collector resistor of 1 k Ω causes a voltage drop of 2.4 V. The resulting collector voltage is 6.6 V, along with a collector-emitter voltage of 4.2 V.

The circuit displays a strong relationship between voltage and negative feedback. A small variation in the emitter voltage has a direct effect on the base-emitter voltage and, on account of the steep characteristic curve of the base, leads to a significant change in the collector and emitter currents. Since only very small variations in base-emitter voltage are desired, the emitter voltage is set at a value that is always about 0.6 V below the base voltage. The function resembles that of the so-called emitter follower (see panel).

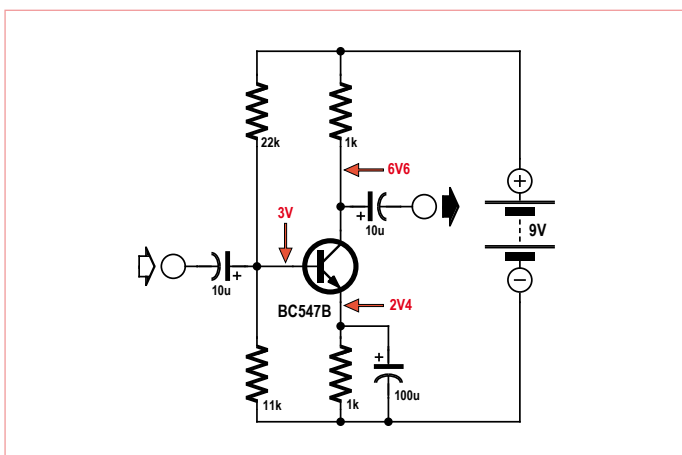


Figure 3. Stabilizing the bias point.

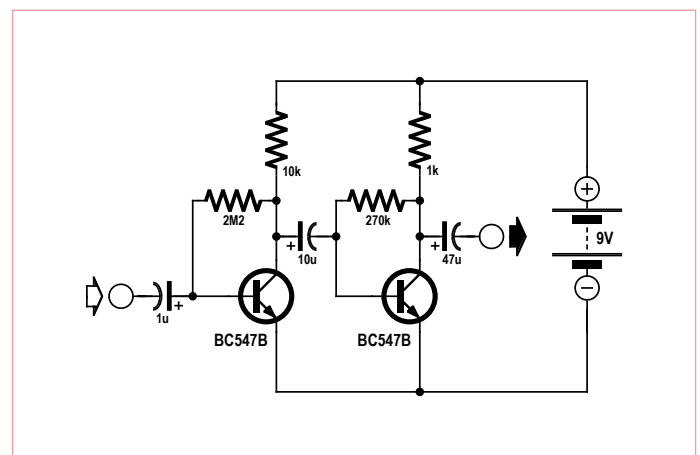
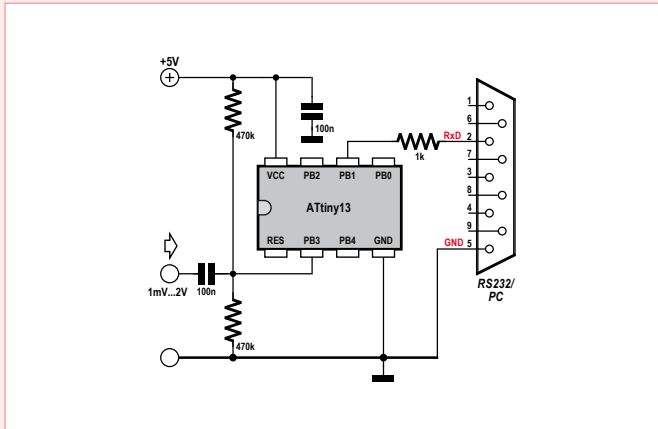


Figure 4. A two-stage audio amplifier.

An audio millivoltmeter

An audio millivoltmeter would be very handy for investigating the amplifiers described here. The A-D converter in the ATtiny13 can in fact measure only DC voltages. All the same, by elevating the mean voltage to 2.5 V and making rapid measurements, AC voltages can be investigated as well. The result is sent serially to the PC and can then be displayed using a terminal program.

The principle of the measurement program is straightforward. First



we determine the mean voltage by averaging. There then follows a rapid series of multiple individual measurements, from which the mean voltage is subtracted, creating an absolute value for each instance. The smallest measurable voltage step for the A-D converter is actually 5 mV. Even so, by averaging many separate readings we can get down to 1 mV. In the actual measurement loop process we carry out, process and add up 2,780 readings. Incidentally, the reading points are entirely asynchronous with regard to the test signal. However, the large number of test measurements made and the random factor combine to produce an adequate result, so long as the input signal remains between roughly 50 Hz and 50 kHz.

How do we arrive at this ominous figure of 2,780? This figure is based on a reference voltage of 5 V and takes into account the difference between arithmetic averaging and true RMS measurement, so that the result in the case of a sinusoidal signal is displayed as actual effective values in mV_{eff} . In order to determine accurate rms (root-mean-square) values, you would have to actually add up the square of the voltage and afterwards calculate the mean value from the square root. This would overwhelm the ATtiny13, however. For this reason we derive the arithmetic mean from the absolute voltage. This is around 10% too low, or more precisely by a factor of the

root of 2 divided by pi / 2, i.e. 0.9003. One A-D step is $5000 \text{ mV} / 1023 = 4.8876 \text{ mV}$. In addition the measured value is multiplied by 8 and then divided by 4096, thus effectively dividing by 512. In order to display everything correctly in mV, we need to multiply by $512 \times 4.8876 / 0.9003 = 2780$, and this is simplified when 2,780 measurements are totalled.

The result is remarkably accurate. Even values like 1 or 2 mV_{eff} are displayed stably!

```
'Millivoltmeter 1 mVeff ... 2000 mVeff
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4
```

```
Dim U1 As Integer
Dim U2 As Integer
Dim U3 As Long
Dim N As Integer
```

```
Config Adc = Single , Prescaler = Auto
Start Adc
```

```
Open "comb.1:9600,8,n,1,INVERTED" For Output As #1
```

```
Do
```

```
    U2 = 0
```

```
    For N = 1 To 64
```

```
        U1 = Getadc(3)
```

```
        U2 = U2 + U1
```

```
    Next N
```

```
    Shift U2 , Right , 3 ' /8
```

```
    U3 = 0 ' zero point
```

```
    For N = 1 To 2780
```

```
        U1 = Getadc(3)
```

```
        Shift U1 , Left , 3 ' *8
```

```
        U1 = U1 - U2
```

```
        U1 = Abs(U1)
```

```
        U3 = U3 + U1
```

```
    Next N
```

```
    Shift U3 , Right , 12 ' / 4096
```

```
    Print #1 , U3
```

```
Loop
```

```
End
```

In principle a harsh level of negative feedback has the disadvantage of much reduced voltage amplification. To achieve adequate amplification of audio signals we need to boost the negative feedback for AC currents, specifically by using an emitter capacitor. Its value is determined by the lowest frequency being handled, without skimping (avoid using too small a value).

Two stages

If the level of amplification provided by a single transistor is insufficient, a multi-stage amplifier will be required. **Figure 4** shows a version with two stages and R-C coupling. In most situations the simple form of negative feedback, with one resistor between base and collector, will be entirely adequate for obtaining an appropriate bias point. The

circuit also demonstrates a further principle of amplifier development: from left to right, the circuit becomes successively lower in impedance. In this way we achieve high input resistance and low output resistance. The latter would not be up to connecting a loudspeaker but with headphones on the output, it would be fine for many applications.

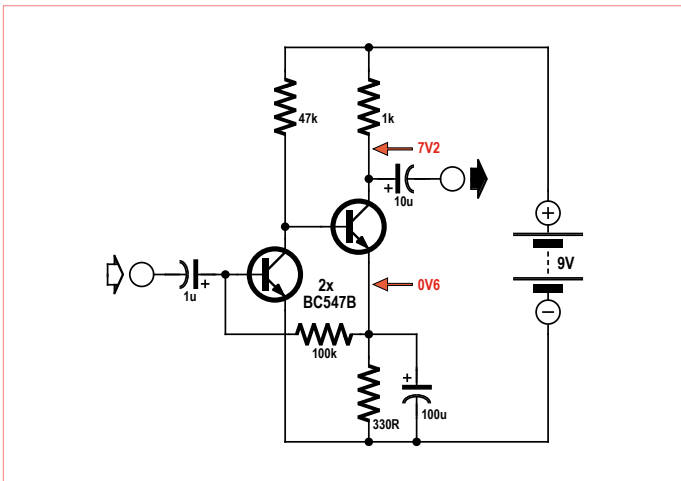


Figure 5. A two-stage audio amplifier with DC coupling.

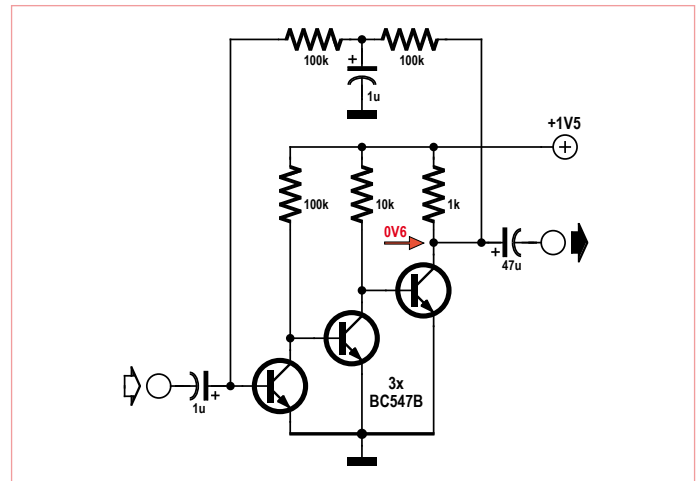


Figure 6. A three-stage, direct-coupled amplifier.

DC-coupled stages

Multi-stage audio amplifiers can be created at reduced cost if you forego capacitor coupling between the individual stages and employ DC coupling instead. The collector resistor of the first transistor is now simultaneously the base resistor of the second one (Figure 5). Frequently this also simplifies setting the bias point, with the base current being appropriate for both transistors. Figure 5 shows a two-stage, directly coupled amplifier. The base current of the first transistor is branched off from the emitter of the second stage. This provides negative feedback and stabilization of the bias point. The voltage drop across the relatively small base resistor of 100 kΩ can be disregarded and consequently the emitter voltage at the second transistor amounts to around 0.6 V. In the process the emitter resistor of 330 Ω determines the emitter and collector currents of the second stage, which are largely independent of the supply voltage.

The negative feedback should not lead to any reduction in AC voltage amplification. An additional emitter capacitor takes care of ensuring that only DC current is fed back. In practice the amplifier has a low frequency limit that can be set even lower with a larger emitter capacitor.

Three stages

The position is even simpler with a three-stage amplifier because negative feedback can be applied direct from the output to the input (Figure 6). Each stage reverses the signal phase by 180 degrees. With a two-stage amplifier we needed a flip-flop to achieve feedback. On the other hand, one

or three stages produce negative feedback, which stabilizes the bias point.

Figure 6 shows a circuit optimized for small supply voltages from 1 V upwards. Here, as in Figure 2, we see how greater amplification in the transistors leads to increased collector current flow in the third transistor and hence to larger voltage drop across its collector resistor. The collector-emitter voltage falls and consequently also the voltage dropped across the two 100 kΩ negative feedback resistors. This results finally in reduced base current, as explained previously above.

The collector-emitter voltage here amounts consistently to merely 0.6 V or so, enabling each stage to deliver relatively modest amounts of audio amplification. All the same, this is more than compensated by the high number of amplifier stages.

The negative feedback element of the circuit includes a low-pass filter with a capacitor to ground (bypass capacitor), which boosts the negative feedback for higher frequencies. It is essential for achieving high levels of audio amplification. The circuit is, for example, ideal as a sensitive microphone preamplifier or as audio amplifier in simple radio receivers. A headset can be connected directly in place of the collector resistor in the third stage. Incidentally the circuit was used in the 'Tapir' RF Sniffer described in the Elektor July & August 2012 edition (www.elektor.com/120354).

(120008)

Quiz solution from the June 2012 edition

The correct solution code is 'CDG'. Here is how it is worked:

Solution 1:

The LED drops around 1.8 V, with U_{CE} at the transistor about 0.1 V. This leaves 7.1 V over. 1 kΩ and 470 Ω form a voltage divider of around 3.1:1. $7.1 \text{ V} / 3.1 = 2.3 \text{ V}$ (Solution C).

Solution 2:

For the left-hand transistor to start conducting its base voltage must be around 0.6 V above the emitter voltage, hence $2.3 \text{ V} + 0.6 \text{ V} = 2.9 \text{ V}$. The correct solution is D.

Solution 3:

If the right-hand transistor is fully cut off and the left-hand one is hard on, we will find approx. $9 \text{ V} / 11 = 0.8 \text{ V}$ at the emitter resistor. This gives a base-emitter voltage of 0.6 V, which the left-hand transistor requires to conduct. Therefore the lower switching point lies around 1.4 V (G).

Vocal Annunciator

How about making your next project speak?

By Christian Picard (France)

When your attention is occupied elsewhere, it's sometimes impossible to watch a display — to keep an eye on a changing value, for example. Here's just one example: you're keenly watching your favourite R/C model, on the ground or in the air, which just happens to be fitted with telemetry. Now wouldn't it would be a great idea if you could keep track of its speed or altitude, or the drive battery voltage, just by listening. What do you say to that?

Module description

The voice of the whole thing comes from a module by 4D Systems, the SOMO-14D [1], which is capable of reading files from a micro SD memory card of 2 GB maximum. The module's audio output can be to a speaker or headset.

As for the brain of this project, that's an Atmega8AU microcontroller, clocked by an 8 MHz crystal, whose mission is to drive the sound module that is going to perform the voice playback of the number or message analysed. The sound module is driven via a 2-wire (data and clock) serial connection, managed by the microcontroller which receives its own orders as a slave on the I²C bus.

This circuit will be connected to the I²C bus of an application and will make it possible, for example, to read out loud a string of ASCII characters representing a whole or floating-point number, positive or negative, resulting from a calculation and stored in a 'string' variable.

It is possible to use the SOMO module directly within an application by using the 2-wire bus — for example, just for playing sound files — but this would tie up the main program and memory space, which may be at a premium in any case if the microcontroller is being used in 'lean' mode! A further advantage of the unit described here is its transportability to another application.

Firmware

The microcontroller's main program consists of little more than a wait loop that analyses the string received as soon as the whole of the I²C data frame has been decoded. Within this loop, the received characters trigger an interrupt per character. The program will leave the loop at the end of the string, indicated by the 'null' character \$00, followed by two hex characters representing the num-

eight point
sixty three
volts



ber (name) of the file in the micro SD card containing the units of the measurement value that has just been read out (or any other message).

Some clarification may be in order here. Let's suppose that the number to be read out is -235.12 V. The ASCII data string to be transmitted is assembled as follows (hexadecimal): \$2d \$32 \$33 \$35 \$2c \$31 \$32 \$00 \$00 \$69. Now you'll have recognized the minus sign in the character \$2d and the comma, \$2c. I'm sure you've followed the rest, except perhaps for the last two. Remember, there's a unit to be read now: 'volt'. In point of fact, \$00 \$69 (= 105 in decimal) is the number of the file "volt" on the micro SD card. Note that this applies to the author's own files — it may well have a different number in your system, since it's up to you to record your own sound files.

Returning to the analysis, the first check is to determine if the received string corresponds to a number or a direct address to the SOMO-14D module. If the first character received is the null character (\$00), the program treats the next two hex bytes as a file number and this file is read. If the first character is not the null character, the string is taken as an ASCII representation of a number. In this case, the programme looks to see if it is a whole number, a decimal, positive or negative, how many digits before the decimal point, how many after it (1, 2, or 3 digits maximum), etc. These latter elements will determine which of the various subroutines will be called. Among the bytes received that are neither numbers nor file numbers, there may be direct commands to the sound module, for example adjusting the volume (\$FF \$F0 to \$FF \$F7).

A minor adaptation will be necessary for using English (or other languages), just for the numbers. For example, the specific differences between the French and English language way of expressing "hundreds" (FR: cent, EN: one hundred) and "thousands" (FR: mille, EN: one thousand). Also, the decimal point should replace the comma.

It's a subtle point, but does justify a slight modification in the structure of the software.

To note...

- Leading non-significant zeroes in the string will be suppressed.
- If the string starts with a zero (ASCII \$30), the number will be interpreted as being less than one and a maximum of three digits after the decimal point will be read.
- When a number is zero, the string will be \$30 \$00.
- For a number with no units, the null character at the end of the string will be followed by \$C0, so no unit will be spoken and the sound display will be stopped.

A few examples (after adaptation to English)

- 12 will be read out as "twelve".
- 527 will be read out as "five hundred twenty-seven".
- 13,689,222 will be read out as "thirteen million six hundred eighty-nine thousand two hundred twenty-two".
- 0.155 will be read out as "zero point one hundred fifty-five"
- 0.155 will be read out as "minus zero point one hundred fifty-five"

Circuit

The project circuit diagram is shown in **Figure 1**. The module is relatively compact (25 × 40 mm) and should be easy to fit into applications. Constructing the board takes a bit of dexterity, but not too much. Manual soldering of the ATmega8AU (SMD type in a TQFP32 package) is made easier by using solder paste, the same as is used for reflow soldering. Here's the method I use: the IC is positioned accurately and held in place using a spring clip, then a moderate amount of paste is applied to the IC pins. A fine-tipped soldering iron will do the job with no risk of bridging between pins. There's space for an HE6 socket to allow the microcontroller to be programmed.

A few resistors and capacitors in SMD packages are required. The two 1N4001 diodes (D2 and D3) are recommended by the manufacturer to allow the sound module to be powered from 5 V (rated at 3.3 V). The module can be soldered directly into the board, or mounted by means of IDC socket strips.

The output at J9 is intended to drive a speaker or headset of 8, 16, or 32 Ω; the sound module is capable of supplying up to 250 mW. D4, a 3 mm LED, shows when the SOMO module is busy. Communication with the I²C bus can be achieved by hard-wiring or SIL strip connectors.

The pin assignments for J11 are as follows:

Pin	Function
1	not connected
2	+5 V
3	0 V
4	SCL
5	SDA

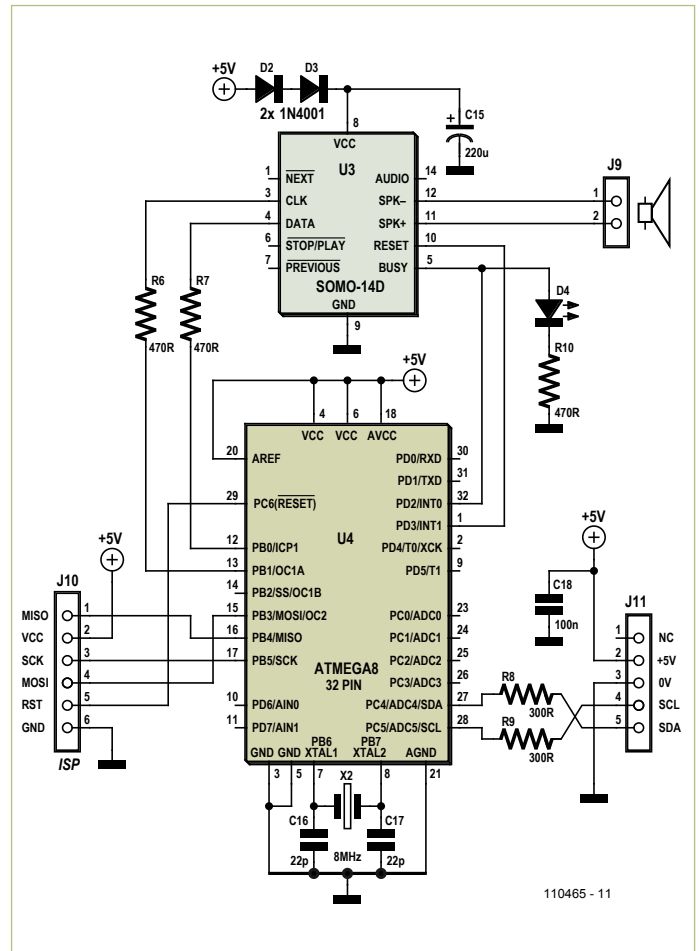


Figure 1. Circuit diagram for the Vocal Annunciator.

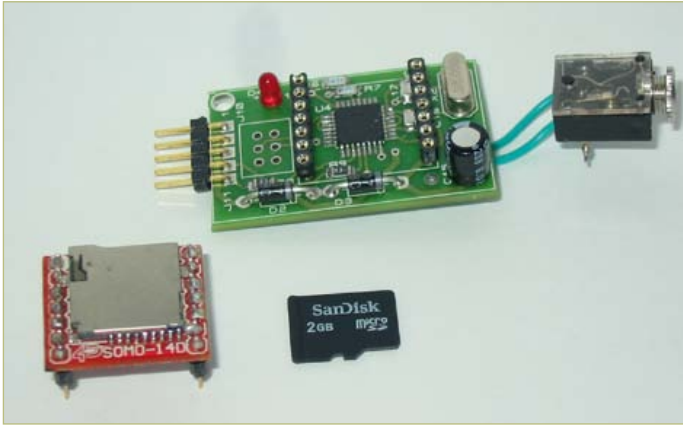
Everything's stored in the micro SD card

The memory contains the files required for making up numbers and messages. The relatively high capacity of the micro SD memory, 2 GB maximum, means it can be used in various different fields. Around 105 files are needed to produce numbers in French. So that leaves the user with 407 files that can contain various units, words, phrases, music, alarm messages for the widest range of applications.

I²C commands

The module's address is \$58. The access protocol is standard, a START followed by the module address, then the data representing the characters of the string to be processed, ending with a STOP command.

I haven't tested all the direct commands. The command for setting the volume has been validated, along with the direct addressing of the sound files.



The main components of the Vocal Annunciator.



The assembled circuit.

SOMO-14D commands according to the manufacturer's data.		
Command code	Function	Description
0000h – 01FFh	Audio file number	Addresses one of the pre-recorded audio/sound/voice files in the micro SD memory card (from 0 to 512 files max.)
FFF0h – FFF7h	Volume	Code for setting the volume, 8 levels in all.
FFFEh	Play/pause	Plays or pauses the current audio file.
FFFFh	Stop	Stops playing of the current audio file and puts the module into idle mode.

File details

The elementary files in French used by the analysis program may be downloaded from [2] – they are all ready for loading into a micro SD card. However, users can create their own messages and the elementary files can be replaced by these user files.

The elementary files represent the figures from 'zero' to 'hundred', 'thousand' (0101d), 'mega' (0144d), 'comma' (0129d), 'point'(0130d) and 'minus' (0102d). Because of the requirements of the software, the figures from 0 to 99 will occupy files 0 to 99, which will be converted into hex characters by the software in order to address the sound module. This behaviour only applies to the part involving the reading of the numbers, since as explained above, the direct message commands will be sent in hex.

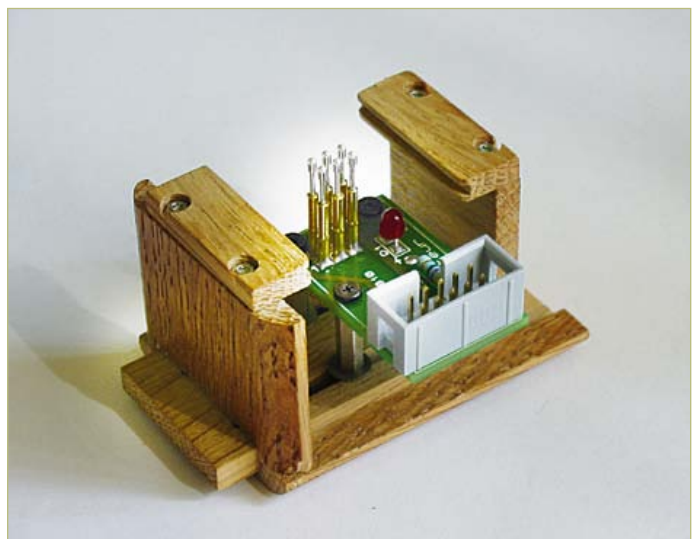
The voices used can be synthesized or recorded live. The only vital point is that recordings must have a minimum length in order for them to be recognized, as the SOMO module will simply ignore files that are too short.

You can download a free utility (SOMO Tool) [3] from the manufacturer's website, which lets you convert .wav or .mp3 sound files into

.ad4, the only format the SOMO module can recognize. The manufacturer warns that certain micro SD cards are not suitable. The author uses SanDisk cards, which up till now I've found perfectly satisfactory, as long as you respect the minimum file length. (110465)

Internet links

- [1] SOMO module: www.4dsystems.com.au/prod.php?id=73
- [2] Software and author's sound files (in French): www.elektor.com/110465
- [3] SOMO audio converter utility: www.4dsystems.com.au/prod.php?id=74



For small circuits, fitting a programming connector can be awkward. To get round this problem, the author has created this programming device.

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**

**SCIENCE,
ROBOTICS &
ELECTRONICS**
IMAGES Scientific Instruments Inc.
Microcontrollers
Servo Motor Controllers
Artificial Vision
Speech Recognition
Flex Sensors
www.imagesco.com
Tele: (718) 966-3694 Fax: (718) 966-3695

**THE NEXT
GENERATION OF
MAXSONAR**

The HRLV - MaxSonar Sensors

- Amazing One-Millimeter Resolution
- Simultaneous Multiple Sensor Operation
- Superior Noise Rejection
- Target Size Compensation for Accuracy
- Temperature Compensation (\$4.95)
- Outputs now include TTL Serial

\$34.95 (MSRP) www.MaxBotix.com

Pololu
Robotics & Electronics

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

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

www.pololu.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

**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

MICROCONTROLLER KITS



**LEARN
CREATE
EXPLORE**



EX-MORSE-1
Pictured

ATRIA Technologies Inc

www.AtriaTechnologies.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

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

VISA MasterCard PayPal IPC MEMBER

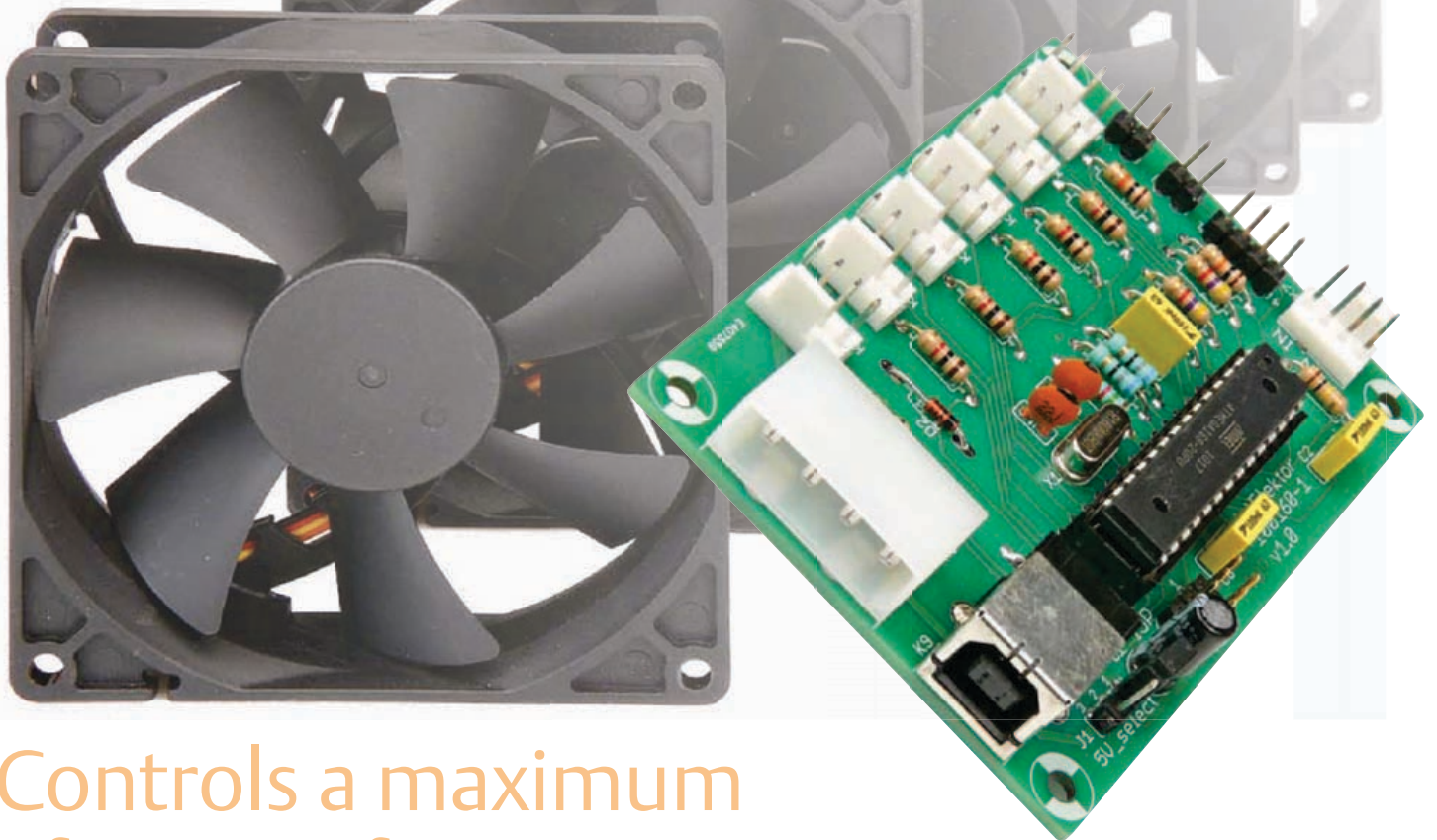
**TO BOOK YOUR SHOWCASE SPACE
CONTACT STRATEGIC MEDIA MARKETING INC.**

Tel. 1-978-281-7708

Fax 1-978-281-7706

Email ElektorUSA@smmarketing.us

Intelligent PC Fan Controller



Controls a maximum of 6 PWM fans

In a modern PC you will find a number of fans for removing the heat that is generated. These can't always be controlled individually by the PC motherboard. With the help of the circuit described here you can actively control up to six fans, while the temperature can be measured in various places inside the PC case using a number of sensors. A PC program is used to configure and monitor the fans, which communicates with the fan control board via a USB link.

By Ivo Pullens (The Netherlands)

A number of years ago Intel created a specification for 4-wire PWM driven PC fans (see [1]). These fans are hardly more expensive than the well-known 2-pin and 3-pin

varieties, although they have the advantage that their rotational speed can be controlled using a PWM signal. The 3-pin and 4-pin connectors also include a tachometer signal that can be used to feed the instantaneous speed of the fan back to the motherboard. A PC motherboard often has only a single

fan connector that includes a PWM signal, although a PC usually contains several fans. There is no choice but to run these other fans at a fixed speed. And even if there is a facility to connect several PWM fans, it usually isn't possible to control them individually from the motherboard.

The circuit described here is capable of controlling the speed of up to six fans in a PC in a safe and flexible manner, so that they will run at their minimum required speed and hence keep the noise level as low as possible.

The circuit

It can be seen from the circuit diagram in **Figure 1** that the circuit itself is very straightforward. It consists of not much

more than an Atmega microcontroller and a number of connectors for connecting the fans, sensors, USB and power supply. The 4-wire fans are connected directly to the microcontroller, since they are able to cope with the generated PWM signals without any further processing. The internal pull-ups of the microcontroller make sure that the open-collector tachometer signals from the fans are pulled towards the positive supply voltage.

R3/C2 is a low-pass filter that smooths (integrates) the incoming PWM signal from the motherboard, so that an analog input of the microcontroller can be used to read the PWM control signal from the motherboard.

For the analog temperature sensors you can use two 10 kΩ NTC sensors, which are connected to K11 and K12. Together with the NTCs, R8 and R9 form two potential dividers that are connected to ADC2 and ADC3

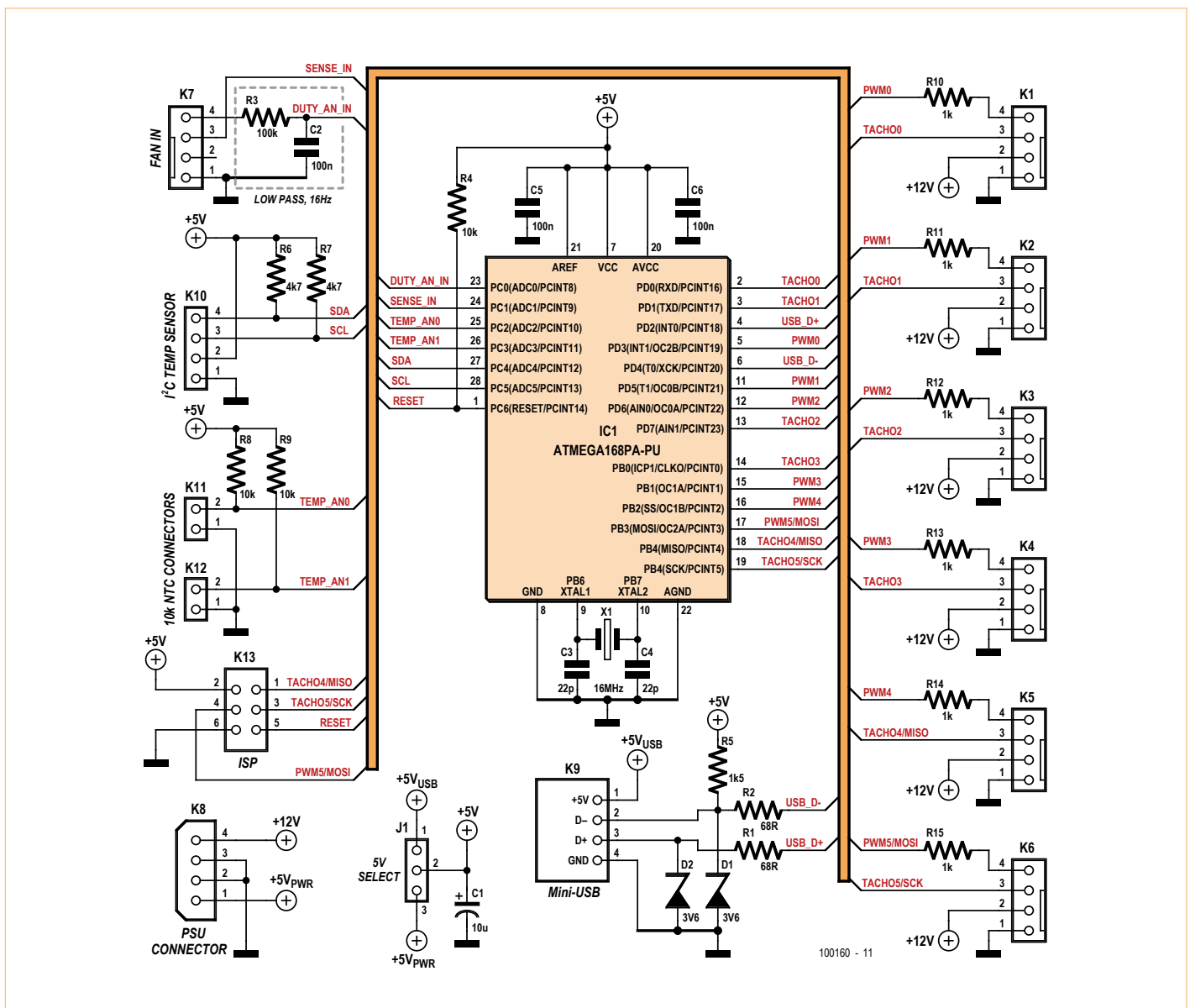


Figure 1. The circuit consists mainly of an Atmega microcontroller and a large number of connectors.

COMPONENT LIST

Resistors

R1,R2 = 68Ω
 R3 = 100kΩ
 R4,R8,R9 = 10kΩ
 R5 = 1.5kΩ
 R6,R7 = 4.7kΩ
 R10-R15 = 1kΩ

Capacitors

C1 = 10μF 16V radial, lead pitch 2.5mm
 C2,C5,C6 = 100nF, lead pitch 5mm
 C3,C4 = 22pF

Semiconductors

D1,D2 = 3.6V zener diode, 400mW
 IC1 = ATmega168PA-PU, programmed,
 Elektor # 100160-41

Miscellaneous

X1 = 16MHz quartz crystal
 J1 = 3-pin pinheader with jumper
 K1-K6,K7 = 4-pin fan header
 K8 = 4-pin PC supply plug, PCB mount
 K9 = Mini USB-B connector, PCB mount
 K10 = 4-pin pinheader
 K11,K12 = 2-pin pinheader
 K13 = 6-pin (2x3) boxheader
 PCB # 100160-1 [3]

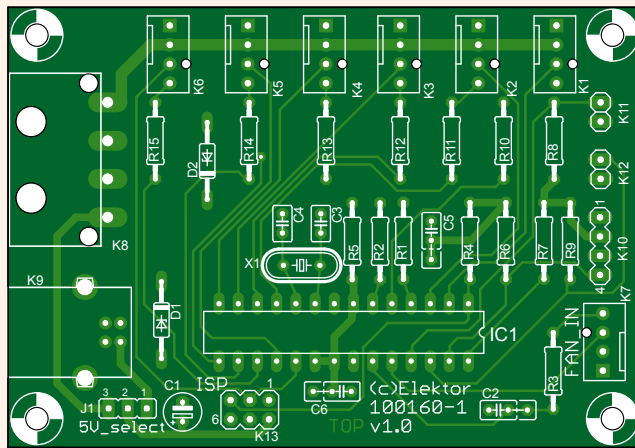
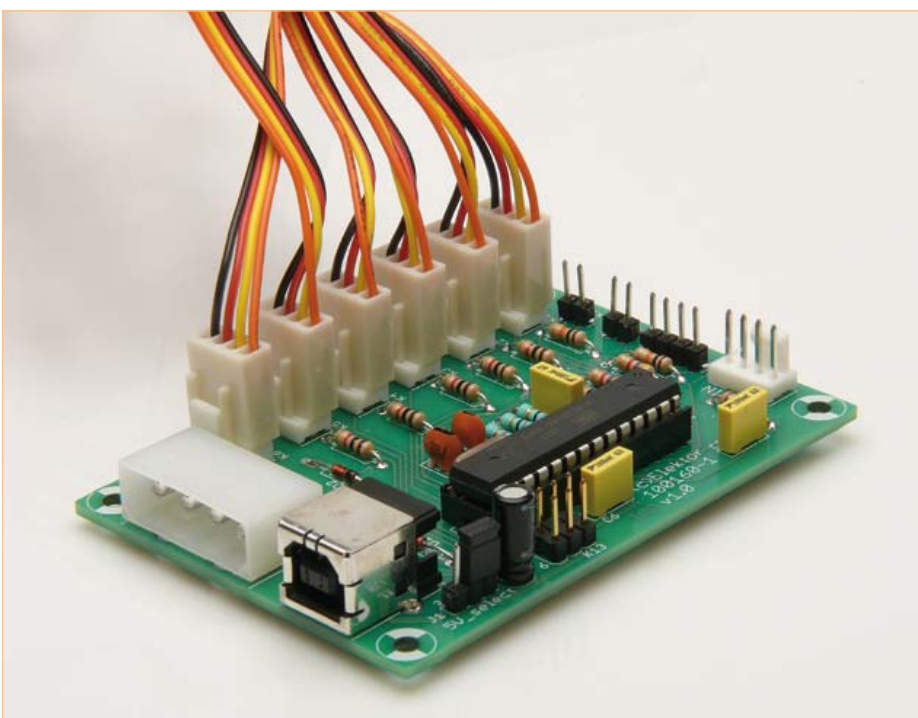


Figure 2. With this PCB the construction of the circuit becomes child's play!



of the microcontroller.

If you wish, you can connect a maximum of eight I²C temperature sensors to K10. R6 and R7 are the required pull-ups on the bus. The firmware has been tested with Microchip MCP980x sensors, but similar sensors such as the TCN75 should work either directly or with a small modification. When the firmware starts it figures out itself how many sensors are connected to the I²C bus. Jumper J1 is used to select either the USB bus or the PC as the source of the power supply for the microcontroller. During firmware development, when no fans are connected, it can be convenient to power the circuit via the USB bus, but normally the PC supply would be used (jumper on pins 2 and 3).

Construction

Since the circuit doesn't have any SMDs the construction is quite easy, especially if you use the PCB designed for this project (Figure 2). The large connector (K8) is the male version of the well-known PC hard drive connector, made by Molex. If required, the power supply cable could also be soldered directly to the board.

The official 4-pin fan headers (K1 to K7) may be difficult to get hold of in small quantities. The 3-pin version, however, is much more widely available. The form and connections of both connectors are identical, so that 3-wire fans can be connected to 4-pin headers. Conversely, if you mount a 3-pin fan header and in the space for the fourth pin mount a single header pin you end up with the equivalent of a 4-pin header (see Figure 3). This makes it easy to connect PWM fans without the risk of connecting them the wrong way.

Firmware

The software for the microcontroller is written in C and makes use of a software-based USB stack V-Usb (see [2]) to communicate with a PC. The software was initially developed for an ATmega88, but this was soon found to be too small to accommodate all the functions it was required to perform. For this reason we changed over to the ATmega168, which has double the amount of flash memory. However, it is still possible to compile the software for the ATmega88,

Specifications

- Autonomous individual control of up to six 4-wire PWM fans.
- Measurement of the rotational speed of these fans and detection of stalled fans.
- Support for a maximum of 8 sensors to control the fan speed:
 - 2 NTCs
 - I²C temperature sensors, MCP980x/TCN75 compatible
 - Rotational speed of fans can also be used as a 'sensor' signal
 - External 'sensors', where the value can be set via USB
 - The duty-cycle of an external fan control signal, such as emanating from a motherboard
- The rotational speed can be set in three ways:
 - Constant
 - Linear control where the rotational speed depends on a sensor signal
 - PI-controlled that uses a sensor signal and a required sensor value to calculate the rotational speed
- Configuration and monitoring via USB.
- Software library for use in PC applications to let you communicate with the circuit and a PC application to configure and drive the circuit, for Windows as well as Linux.

although with a reduced functionality (refer to the file config.h in the download available from the Elektor website [3]).

As mentioned in the Intel specification for 4-wire fans, they should be controlled with a PWM signal with a frequency of about 25 kHz, where the rotational speed should follow the duty cycle linearly from 20% upwards (at lower duty cycles the speed is undefined). The ATmega contains six PWM outputs, but only two of these (which are driven by the 16-bit Timer1) can be set to exactly 25 kHz. The other PWM outputs are driven by 8-bit timers and can be set to either 7.8 kHz or 62.5 kHz. A little trick is used to make them operate at the required frequency. They are set to operate at the lower frequency of 7.8 kHz, but then the overflow interrupt of Timer1 is used to restart the 8-bit timers, which causes all PWM outputs to operate at 25 kHz.

The pulses from the tachometers, which are used to measure the rotational speed of the fans, are counted using the pin-change interrupt. In practice the fans rarely went faster than 6000 revolutions per minute, where the tachometer outputs two pulses per revolution. When all fans run at full speed this means that the software has to deal with about 1200 pin-change interrupts per second. This may seem like a lot, but the main loop is quite simple which leaves plenty of time for calculating the duty cycles.

This main loop checks every second if any of the fans has stopped, for example because of a fault or obstruction. Depending on the settings, a stalled fan can cause the simulated tachometer signal to stop, thereby signalling to the motherboard that a fan has stopped. Any function on the motherboard that raises an alarm when a fan fails

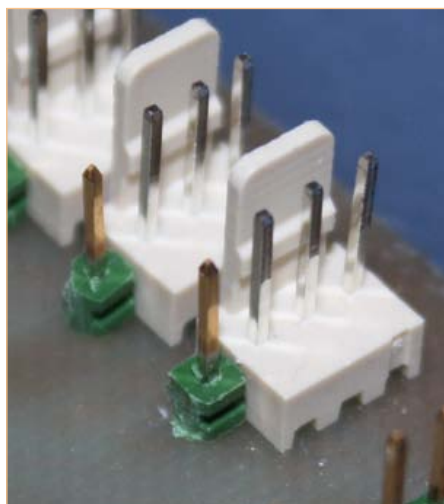


Figure 3. With the addition of an extra pin you can use a 3-pin fan header to make up a 4-pin connector for a PWM controlled fan.

will therefore continue to work.

Next, the current sensor values are read, all the calculations are carried out and the values for the duty-cycles are written into the output-compare registers for the timers. The interval of one second for calculating the new duty cycles may seem like quite a long time in the world of microcontrollers, but it is quite suitable for controlling a slow-changing process such as the temperature inside a PC.

The simulated tachometer signal for the motherboard has to be able to provide a maximum of about 1200 pulses per second, where the timing doesn't have to be that accurate. For this reason the pulses are generated from the main loop, where the tachometer pin is made to go alternately in

a low state and a high-impedance state in order to reproduce the required open-collector output.

With the help of the PC software the current configuration for the controller can be stored in the EEPROM, so that it can be automatically loaded when the circuit is turned on again.

A large part of the software consists of the USB stack and the routines for exchanging messages with the PC. The 'realtime' communication with the PC is taken care of by the INT0 handler, which has the highest priority.

Software

The PC software has been written in C++ and was developed using the free Microsoft Visual C++ 2010 Express. The graphical interface makes use of wxWidgets, which means that it can be used both under Windows as well as Linux (and OS X too, but this has not been tested in practice by the author).

LibUsb has been used to communicate with the circuit via a USB link. A separate library has been used for the creation and exchange of messages. This hides the details of the communication between the PC and the circuit and promotes the re-use of the routines. From here you can directly set the rotational speed of fans or read the sensor values.

The graphical interface consists of the following tabs:

• Admin

Here you can manage the EEPROM, such as reading or writing the configuration. Remember to store a good configuration into the EEPROM at the first opportunity, otherwise the circuit will have a blank configuration when it restarts!

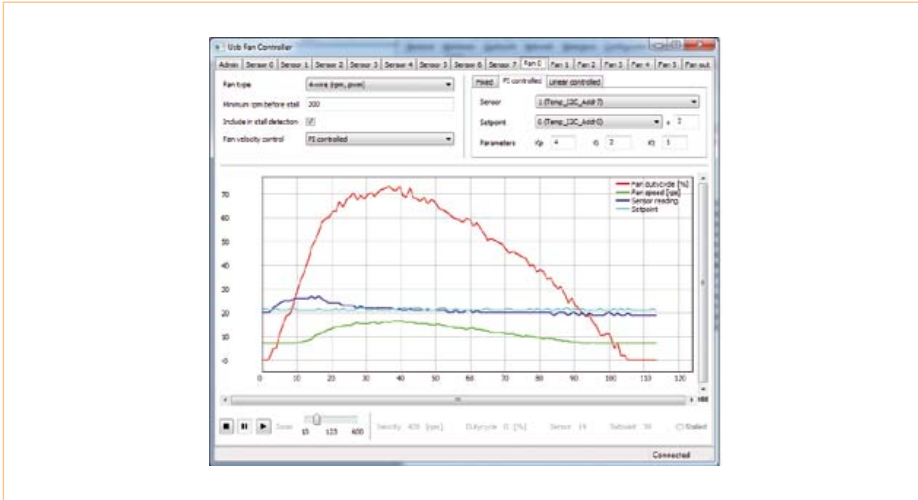


Figure 4. A screendump of the associated PC program that is used for the configuration and monitoring of the fans.

• **Sensor0 to 7**

The sensor tabs let you assign the connected sensors to a sensor input and inspect their value graphically over time.

• **Fan0 to 5**

Here you can configure the individual fans.

• **Fan out**

This is used to configure the simulated tachometer signal to the motherboard.

Before a sensor can be used as a controller in the circuit it needs to be assigned to a sensor input.

Figure 4 shows an example of the configuration for Fan 0, which is connected to K1. At the top-left the type of fan (so the soft-

ware knows if a tachometer and/or PWM signal can be used) and the type of control can be selected. In this example the I²C sensor with address 0 (Temp_I2C_Addr0) measures the temperature of the outside air that is sucked into the PC case and the sensor with address 7 (Temp_I2C_Addr7) measures the temperature inside the case itself. Since the PC generates heat itself, it would be unrealistic to expect to get the inside temperature the same as the outside temperature. The controller will therefore be configured such that the inside temperature will be at most 2 °C higher than the outside temperature (the setting +2 after the set-point). The PI (Proportional-Integral) control selected here continuously calculates the difference between the required tem-

perature (outside temperature +2 °C) and the measured inside temperature. The controller will increase the rotational speed of the fan proportionally (using factor Kp) and also take account of integrating errors over time (factor Ki).

In the figure you can see that the fan starts off slowly at a duty-cycle of 0%. When the inside temperature increases (the dark blue line) above the required temperature (the light blue line), the controller ensures that the rotational speed of the fan increases (the green line, in revolutions per second). When the required temperature has been reached again after some time the rotational speed gradually decreases.

Another possible application for this circuit is to synchronize the speed of several fans in order to keep vibrations in the PC case to a minimum. To do this, use a PI controller with the rotational speed of one fan as the sensor for all other fans and negative factors for Kp and Ki.

On the final tab you can configure the source for the simulated tachometer signal to the motherboard, such as the fan with the lowest rotational speed or a copy of one of the connected fans. For each fan a minimum required rotational speed can be specified. If it falls below this limit the software considers that the fan has stalled and the simulated tachometer signal to the motherboard will be stopped.

In the Elektor download [3] you will find both the firmware for the microcontroller as well as the PC software.

(100160)

About the author

Ivo Pullens is a freelancer who has specialized in embedded software development.

He recently worked on a large industrial assignment for the production of semi-conductors and the industrial fieldbus. In his infrequent spare time he develops solutions to everyday problems, usually based on microcontrollers.

More information can be found on his website (in Dutch only): www.emmission.nl



Internet Links

- [1] www.formfactors.org/developer/specs/4_Wire_PWM_Spec.pdf
- [2] www.obdev.at/products/vusb/index.html
- [3] www.elektor.com/100160


Elektor Print

Classical reading: on paper



Elektor Digital

Contemporary & innovative reading:
on PC, notebook or tablet



Elektor PLUS

Comprehensive reading: at home and on the road

**Read Elektor with the
premium PLUS membership!**

Join now or upgrade: www.elektor.com/usa

Current Transformers

Professional current transformers often use toroidal cores and are wound to high precision. The purpose of this article is to demonstrate how to make your own current transformers for projects, using readily available parts. This will allow safe, isolated measurements of current in power supply conductors.

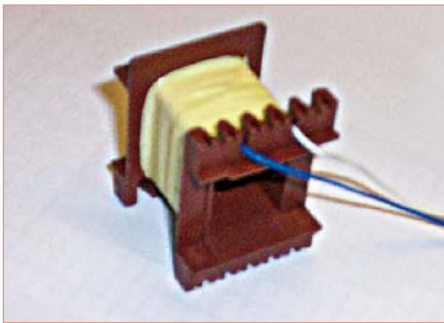


Figure 1. Current transformer with secondary winding on bobbin.

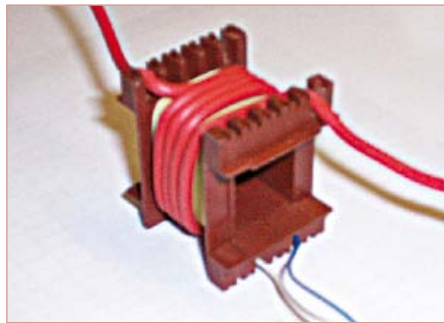


Figure 2. Current transformer with primary winding added.

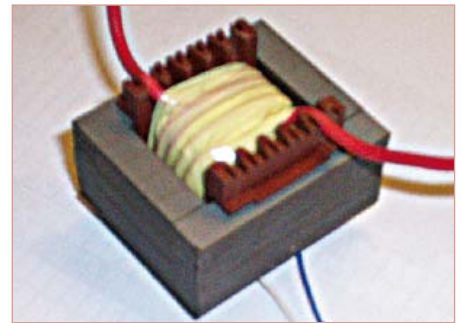


Figure 3. Completed and laminated current transformer.

By Ed Dinning (United Kingdom)

Current transformers (CTs), as their name implies, transform a high current to a smaller one (usually) for metering and control purposes. They also provide isolation between primary and secondary windings; as such they are used extensively in the power infrastructure. Their great advantage over more modern devices such as Hall effect and Rogowski coils is that they are self powered and will operate meters

and protective relays without additional supplies.

Setting up parameters

As with all transformer designs, it is customary to define the load, or burden, as it is mostly also known. For most professional CTs this is usually no greater than 10 VA (loads are often expressed in VA rather than watts due to phase shifts being involved; this is not really a concern in the simple designs we introduce in this article).

For the purposes of this design we will be driving a meter or feeding a comparator as part of a protective circuit and a load of 0.5 VA will be sufficient.

We will assume that we wish to monitor the current in a circuit with a maximum value of 10 A, and that we want to lower this to 50 mA, i.e. a 200:1 ratio.

With a 0.5 VA load and 50 mA of current we calculate a voltage of 10 V ($U = P / I$) and a resistive load value of 200 ohms ($R = U / I$).

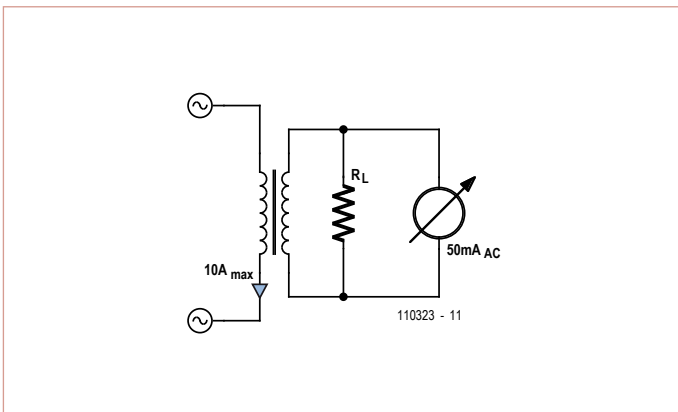


Figure 4. Test circuit for our self-wound transformer.

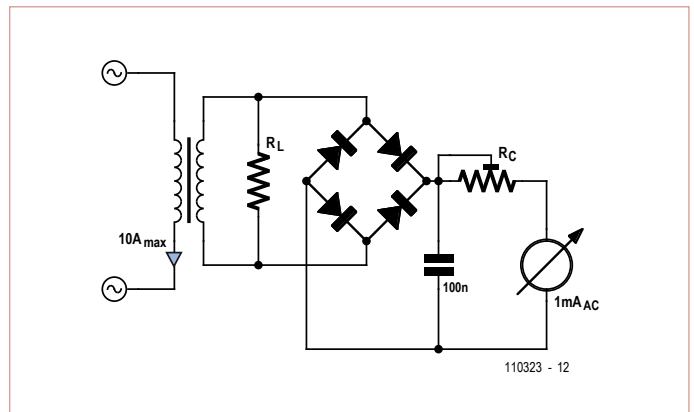


Figure 5. A CT in a circuit with a moving coil meter.

We need to calculate the secondary turns required to go on the transformer. For this application we will use a small transformer that has a core area of, say 300 to 400 mm². These can be obtained cheaply and even recovered from old equipment. The larger the core area, the fewer turns that will be required.

Many commercial transformer laminations are classified with an EI number. We are using an EI48 (long dimension of laminations equals 48 mm) by a 20 mm stack. The center leg of the lamination is 16 mm wide, and with a stack of 20 mm of these the area works out at:

$$0.016 \times 0.02 = 0.00032 \text{ m}^2.$$

We then use the transformer equation:

$$N = E / (4.44 f B A_e)$$

where N is the number of turns, E the applied voltage, f the system frequency, B the flux density in Tesla and A_e the cross-sectional area of the core in square meters. The AC line frequency is 50 Hz (US: 60 Hz). It is common to operate at the relatively low flux density of 0.2 Tesla (unlike a voltage/power transformer operating at >1 Tesla, depending on lamination material). Entering these values into the equation gives:

$$N = 10 / (4.44 \times 50 \times 0.2 \times 0.00032)$$

$$N = 704 \text{ turns (US: } N = 586 \text{ turns)}$$

We're rounding this of to 700 turns. We know we need a turns ratio of 200 to 1, so we would need to use a 3.5 turn primary (US: 2.93). This is not practical, so we increase secondary turns to 800 (US: 600) (giving a lower flux density and greater

accuracy) and use a 4 turn primary (US: 3 turns). Now we have the theory sorted. On to the practicalities of winding.

Winding the core

The secondary is to carry 50 mA. It is normal to work at 3 A/mm² with copper wire. Working backwards this gives a 0.15 mm diameter wire (approx.), i.e. 0.16 mm (#34 AWG) including its enamel coating. The transformer bobbin winding area is 20 mm wide and 6 mm deep. This will allow 125 turns per layer and 6.4 layers (rounded to 7); or a winding depth of 1.2 mm, leaving a space of 4.8 mm for the primary. See **Figure 1**.

Note that the primary is made from one core of 10 A rated mains cable, either flexible or solid core, see **Figure 2**. The primary insulation is provided by the insulation on this wire. Four turns are added on top of the secondary and the windings are held in place with an insulating tape (polyester transformer tape is best suited for this). The transformer is then laminated with alternate 'E' & 'I' laminations, so that there is virtually no air gap, see **Figure 3**.

The unit can now be tested using the circuit shown in **Figure 4**. Note that the load resistor (R_L) **must** always be connected, or the windings can go short-circuit, as current transformers develop very high voltages when run into an open circuit. A DVM in its AC current range can be used to measure the secondary voltage. Its internal impedance relative to the load determines the deviation: the lower the internal impedance, the better.

Example #2

A more practical use of a CT is to drive a

moving coil meter through a rectifier, like in the circuit shown in **Figure 5**. A moving coil meter typically has a full-scale deflection of 1 mA; if this is used in parallel with the load (carrying 50 mA), it will make little difference to the accuracy of the system.

As can be seen from the circuit, the transformer develops a maximum of 10 volts across the load resistor when 10 A are flowing in the primary. If this voltage is fed into a bridge rectifier, we can read a DC voltage across the output of the bridge.

When the 1 mA meter is connected across this DC output, it can be calibrated to read AC amps when a series resistor (R_C) is used. Ohms law tells us that the resistor should have a value of 10 k Ω minus the internal resistance of the meter (typically 100 Ω or less). If the 10 k Ω resistor is made variable, the meter can be accurately calibrated.

Any vibration of the meter pointer can be dampened with a small capacitor across the bridge output. 0.1 μ F should be sufficient. Higher values will cause the meter to 'peak read' and have a longer response time.

If the meter is accurately calibrated, it will be found to be non-linear near the bottom of the scale. To some extent this is due to the forward voltage loss of the bridge rectifier diodes. The use of Schottky or Ge (germanium) diodes here will reduce this effect. The design can also be extended to switch mode and audio circuits to measure current; the use of a ferrite transformer is recommended where the frequency is above about 1 kHz to reduce losses. With a ferrite the flux density should be reduced to about 25 mT to maintain accuracy.

(110323)



Marconi Instruments TF801D/1 AM RF Signal Generator

“They don’t make ‘m like that anymore”

By Stephan Germann (Switzerland)

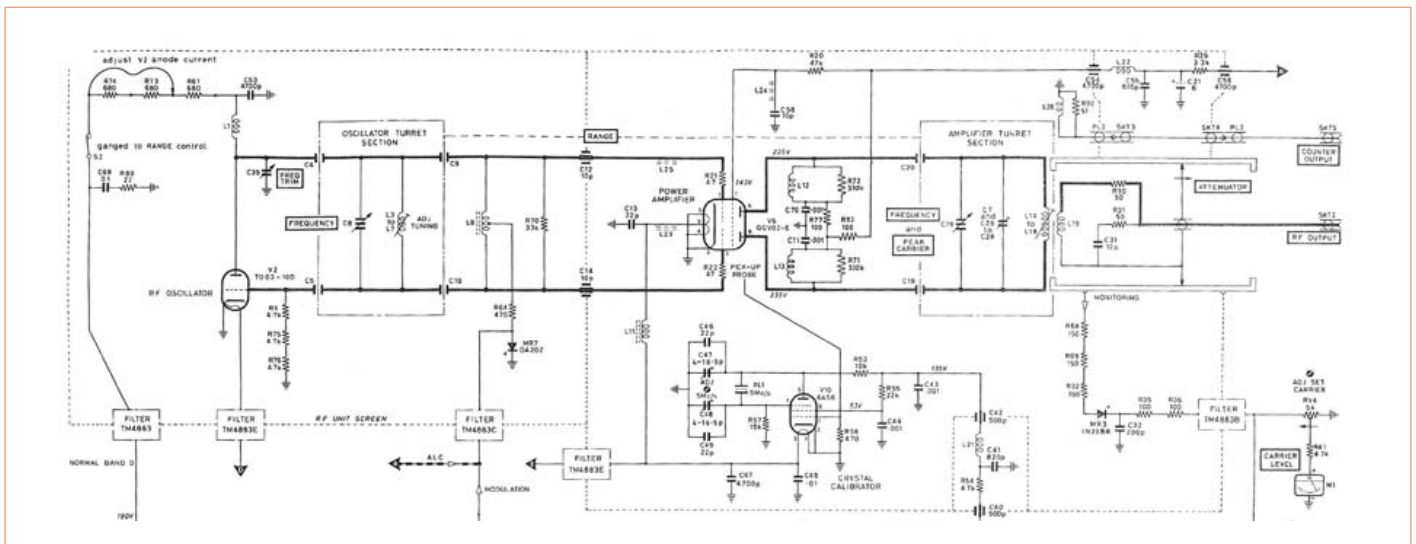
Manufactured by UK-based Marconi Instruments, the type TF801D/1 is an RF generator covering frequencies from 10 MHz to 470 MHz. Built at around 1960, the instrument is likely to have been pretty expensive if only for its highly complicated mechanical assembly. I got this generator ‘thrown in’ when I bought a second-spectrum analyzer about 15 years ago. Since then, it has provided steady service in my workshop set up for the construction or repair of amateur radio transceivers. The electrical data are also very good by today’s standards, the only weak point being the frequency stability, which of course cannot compete with a DDS Generator. With some patience and skills though, even an SSB receiver for the 2 m band can be checked.

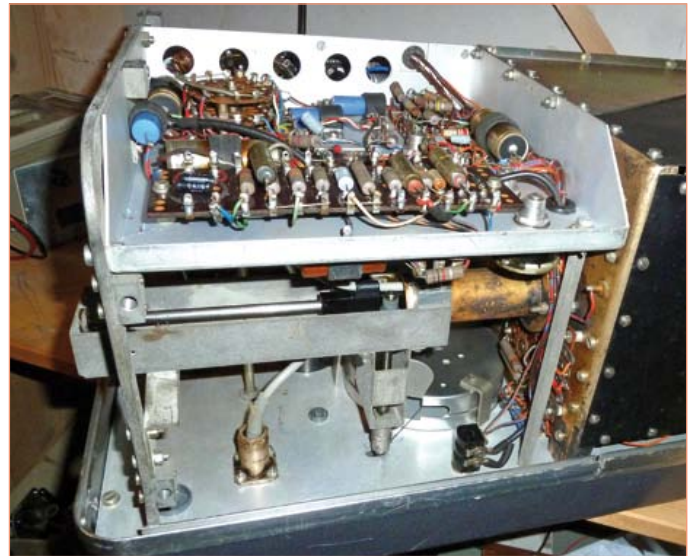
I am using this equipment for adjusting and checking amateur radio equipment, as well as for troubleshooting, not forgetting tuning checks on homemade antennas, filters and matching networks. Thanks to the continuous frequency range, amateur bands from 30 m to 70 cms are catered for, as well as all the regular intermediate frequencies (IFs). The very large output voltage range from 100 nV to 1 V enables both sensitive receivers to be checked, and driver stages of transmitters or passive impedance measuring bridges to be driven. Thanks to vacuum tube technology, the equipment is electrically very robust. The precision of the output voltage and the total absence of RF leakage never cease to amaze me. The operation too is very convenient when

compared with today’s devices. In fact, it beats many of today’s menu-driven controls, or jog-style rotary controls with multiple functions, mainly because each function has its own button, and superfluous functions are absent.

Barring one defective electrolytic capacitor, no repairs were necessary, clearly pointing to the use of superb quality components. The pots for example are wound from resistance wire and sealed tightly. The modulation depth and RF output voltage each have their own moving coil meter on the front panel. The controls are sparsely arranged over the front panel, and all controls and knobs are of ‘comfortable’ size. The frequency scale is so large and accurate that tuning to less than 100 kHz error is possible up in the UHF frequency range without additional measuring equipment. The drive assembly is accurate, and fine-tuning is also provided.

The construction of the TF801D is marked by extreme stability, which becomes manifest if you lift all 30 kgs (66 lbs) of it. UK-made, the unit has imperial screws and bolts only. The internal structure consists of a sturdy chassis, on which the modulator, the control circuit and the power supply are constructed. Attached to it is the RF section in a RF sealed double casing made of silver plated copper sheet. The external RF housing only contains the tuning mechanism as well as the RF filters for passing the control voltage. The RF circuitry itself is located in an internal RF-proof housing that’s divided into two parts for the oscillator and the final stage. Thanks to this method of construction RF voltage appears at the output jack only; in other places it is simply not detectable, which is no mean feat





when considering both the oscillator and the RF power amplifier have an RF power capability of the order of 100 mW.

The circuit itself consists of ten tubes, a couple of semiconductor diodes and a single Germanium transistor (OC71), and is not terribly exciting from today's perspective. Sections of the circuit diagram are reproduced here, due to their size they cannot be printed in full. The power supply consists of a transformer, a 5R4GY full-wave rectifier tube and a voltage stabilizer comprising a 5651 reference tube, an EF95 (6AK5W) control amp and a KT66 audio output tube as the regulating element. The filament voltage for the oscillator tube is stabilized with a transducer (current-controlled inductance). The RF signal is generated 'straight off', i.e. the oscillator oscillates at the desired frequency (= sig gen output frequency) and its output signal is buffered by an RF tetrode. The oscillator employs a TD03-100 UHF triode with five switchable coils around it for the frequency ranges. It is tuned with a variable capacitor. Unusually a heat sink is found attached to the anode terminal of the triode. The oscillator signal directly drives the symmetrical push-pull output stage built around a QV02-6 UHF dual tetrode. The tube oper-

ates in class A, resulting in maximum gain and low distortion (hence, low harmonics). The DC level at the control grids allows the gain to be controlled, or amplitude modulated if required. Thanks to the gain control, the output voltage is independent of load as well as frequency, making the measuring and matching of filters very convenient indeed.

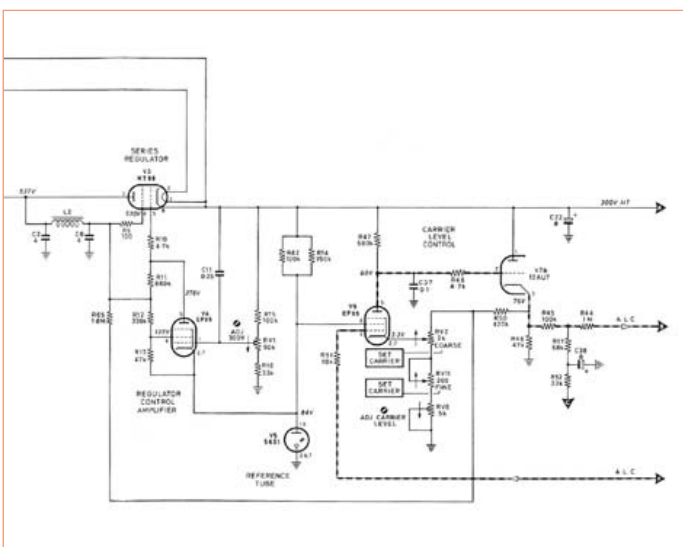
The anode circuit of the amplifier also comprises five switchable coils. Together with a second variable capacitor, the coils act as tunable resonant circuits tuned in synchronism with the oscillator. To prevent tracking errors, the variable capacitor of the output circuit can be slightly detuned with respect to the oscillator — mechanically, a not so simple design. The linear amplification afforded by the push-pull output stage keeps harmonics generation down to a minimum. The output signal is inductively coupled out through the attenuator having a single calibrated range of 135 dB, and providing a resistor network for a frequency-independent, rock steady output impedance of 50 ohms.

The 5-position frequency range selection switch is also worth mentioning. Remarkably, the switch does not work as expected from one marker to the next (to get to the next frequency range), but requires an additional full turn. The switch itself is contactless, the connection with the coils being capacitive. This has the advantage of avoiding contact problems, which is highly relevant in view of the tuned circuits' Q (quality) factors.

Besides the everyday use of the instrument, the manual describes maintenance and common repair steps in great detail, both mechanical (like Bowden cable replacement) and electrical. I especially liked one tip for faultfinding: "Initially, look for burnt resistors to locate a faulty circuit section".

(120169)

Retronics is a monthly section in Elektor magazine covering vintage electronics including legendary Elektor projects. Contributions, suggestions and requests are welcomed; please send an email to Jan Buiting (editor@elektor.com).





By Gerard Fonte (USA)

No, this is not a column about designing cars. This is about the proliferation of various programming packages that will automatically turn you and your whole department into experts in hardware and software design. From what these products imply, education and work experience can be immediately realized by buying their merchandise. Instant intelligence!

No Thinking Required

Labor saving devices have been around since the invention of the machine. And whether it's a wheel or a dishwasher, machines can and do perform better and more consistently than humans in terms of physical actions. However, in mental actions, machines are clearly inferior. Yes, there are chess-playing programs that can beat grandmasters. And IBM's Watson did very well on Jeopardy. But a chess program has a very limited set of possible moves (although the number of the permutations of the moves is large). And Watson is basically an information-retrieval, or data-base, program. Neither of these requires creativity or cognition. And, to the best of my knowledge, no machine can currently imagine or think as humans do (But, there is considerable research into these areas).

There are the inexpensive Printed Circuit Board (PCB) auto-routers that lay out your PCB in minutes. And they do just that. But they don't understand the difference between a sensitive input signal line and a high-power drive line and may route them right next to each other. Can you say oscillation and crosstalk? (There are some PCB layout/simulation programs — mostly for RF work — that can do this, but they can cost tens of thousands of dollars. And you still have to define the important characteristics of the traces).

It is true that many times, even most times, these cheap PCB layout programs will perform adequately and produce a workable product. However, the user gains neither experience nor education when using this software. "So what?" you ask.

The problem is: what happens when the program fails? What happens when mysterious problems are seen with the PCB, but are not seen on the breadboard? How does the inexperienced user approach this difficulty? That person is now required to think. Which is something that hasn't been needed before. Generally, in such cases, the user has no idea in how to proceed. It can be an insurmountable problem for the hobbyist — who abandons the project. And it can be a major obstacle in business — which may require outside consultation and a budget overrun.

Fundamentally, we see that the proper user of this type of software is the expert who can anticipate and recognize the limitations of this product. That is, the person who can best use it, is the person who has the least need for it.

Cut and Paste

There are other products that 'write your code' for you. You provide a few specifications, and it generates the body of your software automatically. It eliminates the tedium of all the details so you can

Auto Engineering

concentrate on the big picture.

In some ways, assemblers and compilers fall into this category. They take high-level language and convert it into assembly or machine language. The big difference between assemblers/compilers and the code-writing programs is that assemblers/compilers are pretty much line by line translators. They take a line of high-level code, break it down into lower level actions and then convert those actions into assembly or machine language. They go from high level to low level or from a complex statement to a series of simpler, equivalent statements. This is a relatively uncomplicated and straightforward procedure. The complications mostly arise in the error checking.

The code-writing programs are different. They generally have blocks of pre-written code which can be varied by the parameters you define. Then these blocks are combined to form a program. As you can see, this is the opposite of a compiler/assembler. The code-writers are creating higher-level content from a lower-level definition. They are assuming 1) that the functions that they have supplied will provide everything you will ever need and 2) all of these functions will work properly in all combinations and variations (This last assumption may be impossible to test because the number of permutations may be astronomical). Again, everything is fine until there is a problem and then everything comes to a crashing halt. Worse, the manufacturers of these systems sometimes keep important aspects of their software confidential. This may make a difficult debugging task simply impossible.

Losing Control

The basic problem of auto-engineering is that the designer is giving up control to someone else. There is no reason to believe that this 'someone else' has any better skills or insights than the designer. And it is certainly true that the designer has far superior knowledge and experience concerning his (or her) own project requirements. Automatic Braking Systems ('ABS' and AKA anti-lock brakes), common in the US (and required in the EU) can significantly reduce the number and severity of accidents. In most cases, vehicles with ABS stop faster than those without. But the driver is giving up some control of the braking function to a machine that has to manage every possible braking scenario. Quite simply, it can't.

On loose surfaces such as newly fallen snow, sand and gravel, the braking distances with ABS can be much, much longer than cars without ABS. On these surfaces, locking the wheels causes them to dig into the loose material and grab better. Locked wheels also push loose material in front of the tires to create a wedge. With ABS, the wheels just roll and roll and roll. Deficient braking on loose surface with ABS can come as a rude (and dangerous) shock.

All of these auto-engineering examples have good points and bad points. Often the good points outweigh the bad. But all limit the performance of the user. You can't learn to lay out a PCB with an auto-router, or create a program with an auto-coder, or stop quickly on snow with ABS. All users of the same auto-engineering system perform at the same level. Syndrome, from The Incredibles movie, put it this way: "When everyone is 'super', no one will be."

(120543)

Hexadoku

Puzzle with an electronics touch

Past page 75 or thereabouts and fed up with all that designing, programming and soldering? Then take a break deservedly and give this puzzle a go. 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 November 1, 2012, send your solution (the numbers in the gray boxes) by email to:

hexadoku@elektor.com

Prize winners

The solution of the July & August 2012 Penta-Hexadoku is: **D89276**.
 The Elektor \$125.00 voucher has been awarded to Norbert Breuker (Germany).
 The Elektor \$60.00 vouchers have been awarded to Donald Hoogendorp (Netherlands), Jean-Marie Bellet (France), and Mike Parker (United Kingdom).
 Congratulations everyone!

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

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

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



1



2



3



Limited Time Offer for Elektor Magazine Members!
20% DISCOUNT
www.elektor.com/october

More than 75,000 components

1 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 • \$40.20

A whole year of Elektor magazine onto a single disk

2 DVD Elektor 2011

The year volume DVD/CD-ROMs are among the most popular items in Elektor's product range. This DVD-ROM contains all editorial articles published in Volume 2011 of the English, American, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy

of PCB layouts at printer resolution, and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-276-1 • \$37.90

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
 \$47.60

Elektor Linux Board

4 Embedded Linux Made Easy

Today Linux can be found running on all sorts of devices, even coffee machines. Many electronics enthu-

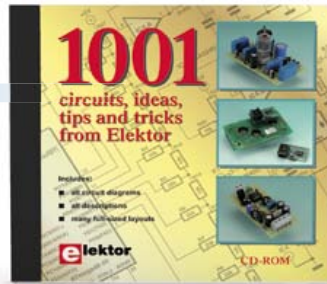
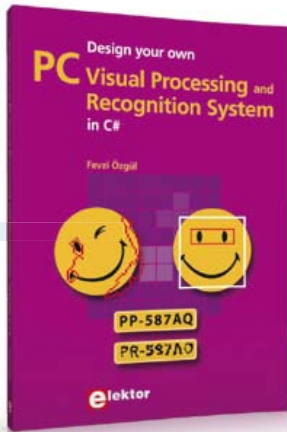
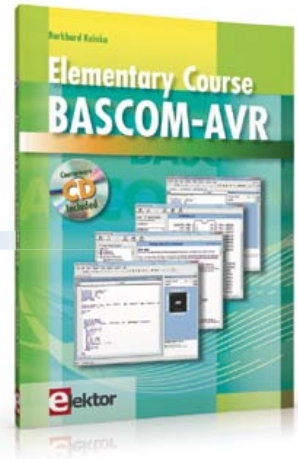
siasts will be keen to use Linux as the basis of a new microcontroller project, but the apparent complexity of the operating system and the high price of development boards has been a hurdle. Here Elektor solves both these problems, with a beginners' course accompanied by a compact and inexpensive populated 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 • \$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 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
\$57.30

Free Software CD-ROM included
6 Elementary Course BASCOM-AVR

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
\$56.40

7 TAPIR Sniffs it Out!

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 • \$21.50

8 CD 1001 Circuits

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 & measure-

ment and of course a section miscellaneous for everything that didn't fit in one of the other sections.

ISBN 978-1-907920-06-6 • \$55.70

9 Package Deal: 12% off AVR Software Defined Radio

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.

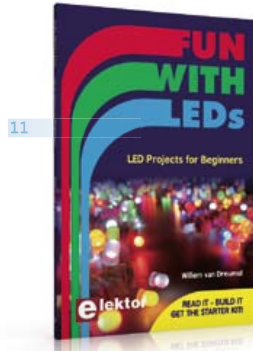
Signal Generator + Universal Receiver + Active Antenna: PCBs and all components + USB-FT232R breakout-board
Art.# 100182-72 • \$133.00

10 Avoid interference and earth loops USB Isolator

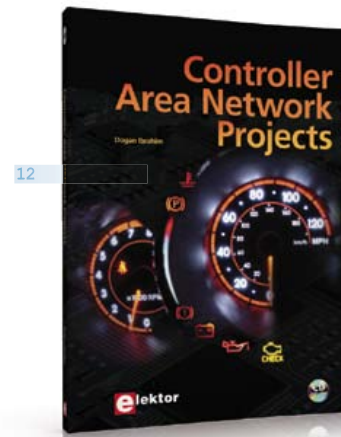
If your USB device ever suffers from noise caused by an earth loop or if you want to protect your PC



10



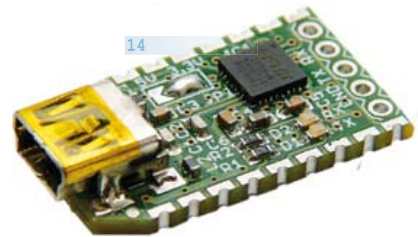
11



12



13



14

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 • \$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
 \$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
 \$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 • \$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 • \$20.90

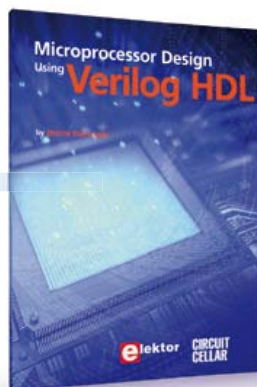
110 issues, more than 2,100 articles
15 DVD Elektor 1990 through 1999

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 alphabetically 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 • \$111.30



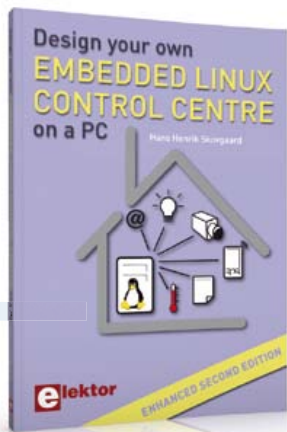
15



17



19



16



18

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
\$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 and serves as a thorough introduction about reducing a computer architecture and instruction set to practice. 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
\$45.00

Counter for alpha, beta and gamma radiation

18 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 • \$57.30

19 Dual-layer DVD: 165 mins. video 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.

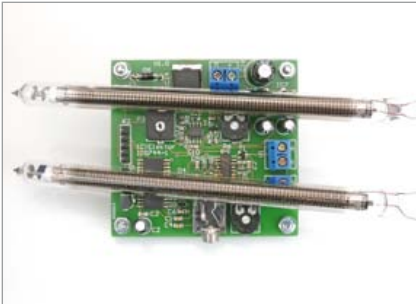
ISBN 978-1-907920-10-3 • \$40.20

More information on the Elektor Website:

www.elektor.com/store

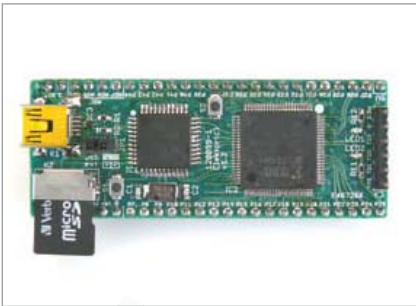
Elektor USA
4 Park Street
Vernon, CT 06066 – USA
Phone: 860-875-2199
Fax: 860-871-0411
Email: order@elektor.com





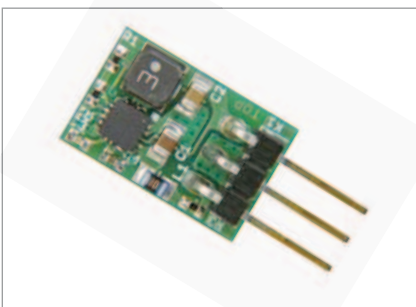
Nixie VU Meter

After various Nixie clocks, thermometers and a hygrometer, a Nixie VU meter gets described in Elektor next month. This circuit is designed to embellish a (tube) amplifier. It supplies beautiful light effects but it is not calibrated to represent instantaneous output power in any exact way. At the heart of the circuit is a Cypress PSoC type CY8C27443, which determines the level of two audio channels and looks after the driving of two Russian Nixie tubes type IN-g.



Elektor FPGA Board

Always wanted to start working with FPGAs, but found nothing but hindrances in your way? Elektor now has a great option for you: an affordable FPGA developer's board that's easy to use, as well as to extend by way of a standard breadboard. The microcontroller on the board takes care of the communicating with the PC via USB, while the configuration data for the FPGA are stored on a micro-SD card located on the board too. Using Windows a configuration file can easily be transferred on to the SD card; when the card is reset this configuration file gets written into the FPGA.



Cool! 7805 Drop-in

The 7805 is unquestionably the most widely used fixed voltage regulator. Unfortunately this golden oldie has a habit of turning all the excess energy into heat. Wouldn't it be nice to have a switching power supply controller as a drop-in replacement for the 7805? The switched device is sure to offer much higher efficiency as well as better specifications. In the Elektor labs, a small printed circuit board was designed around a TPS62150 buck converter capable of turning an input voltage of 5.5 to 17 V into a 5 V stabilized rail at up to 1 A. By adjusting a number of resistance values, other output voltages are also possible. (we are sorry this article could not be published in the October 2012 edition as planned)

Article titles and magazine contents subject to change; please check the Magazine tab on www.elektor.com

Elektor UK/European October 2012 edition on sale October 18, 2012.

Elektor USA October 2012 edition published October 15, 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:

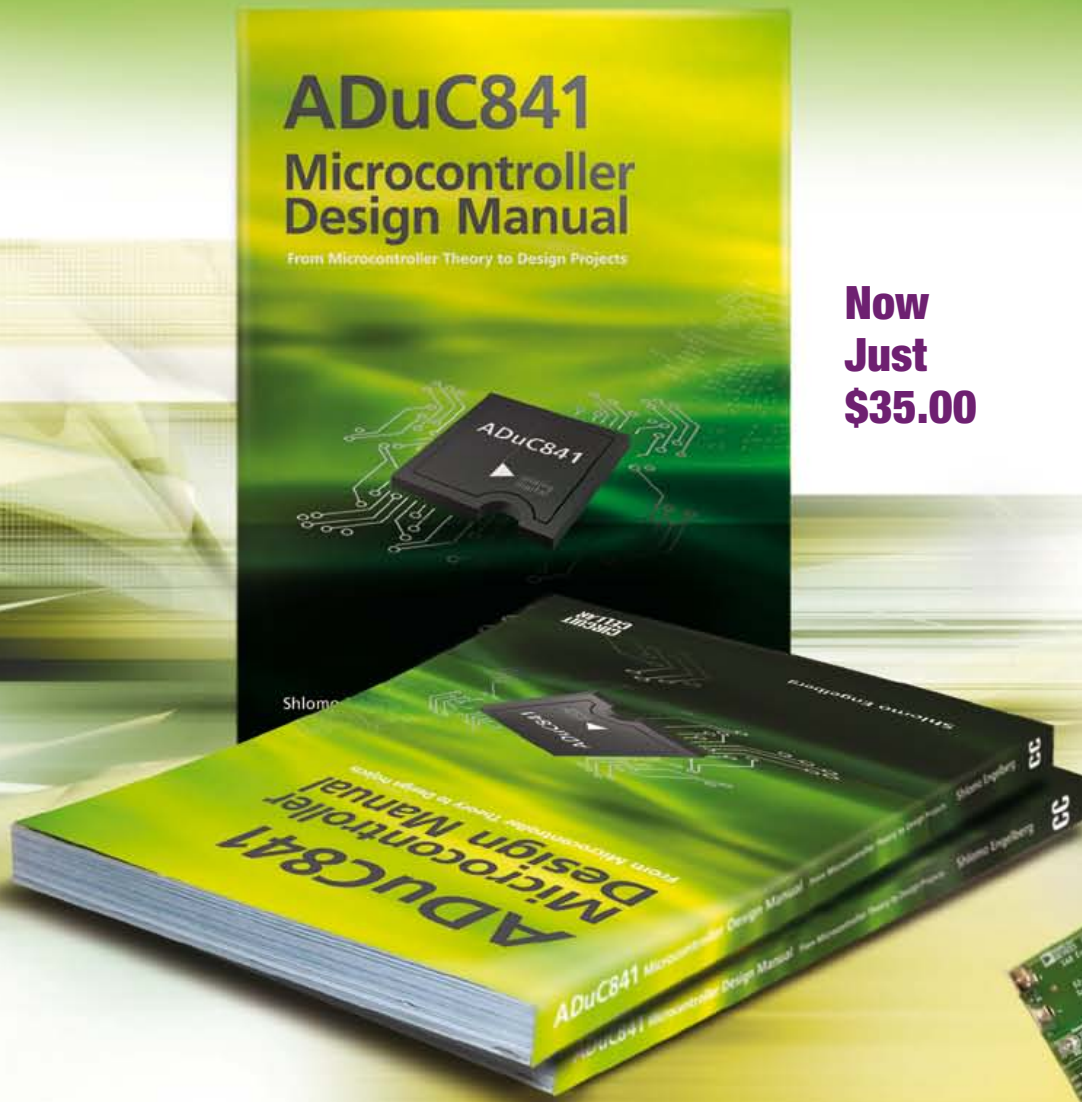
- Electronics news and Elektor announcements
- Readers Forum
- PCB, software and e-magazine downloads
- Time limited offers
- FAQ, Author Guidelines and Contact



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.

MEMBERSHIPS (US & CANADA ONLY)

Membership Rates (1 Yr.)

Standard Membership: \$40 (Canada \$55)

Plus Membership: \$60 (Canada \$70)

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

Order memberships on-line at www.elektor.com/members

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



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.

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 OFFER!

Sign up today and you'll also receive the **Special 25th Anniversary Edition** with your subscription!



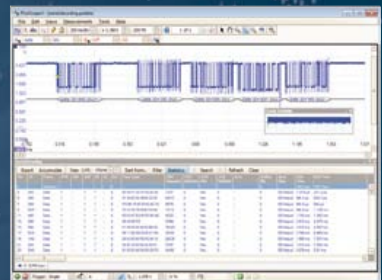
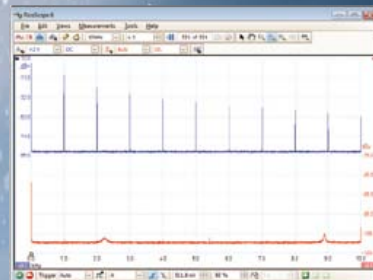
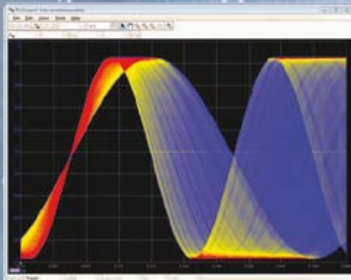
GET MORE WITH THE
CLASS LEADING OSCILLOSCOPES

PicoScope 6000 Series

MORE MODELS • MORE CHOICE
MORE FEATURES AS STANDARD



HIGHEST SAMPLING RATE AND
DEEPEST MEMORY OF ANY
SCOPE FOR THIS BUDGET



	6402A	6402B	6403A	6403B	6404A	6404B
Channels	4					
Bandwidth	250 MHz		350 MHz		500 MHz	
Sampling Rate	5 GS/s					
AWG or Function Generator	FG	AWG	FG	AWG	FG	AWG
Memory	128 MS	256 MS	256 MS	512 MS	512 MS	1 GS
Price	\$3292	\$4117	\$4942	\$5767	\$6592	\$7417



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

For more information visit www.picotech.com/pco483 or call us at: 1-800-591-2796