

+ Time Domain Reflectometry

# eLektor

Exposure levels in check?

Cut-rate  
Radiation  
Meter

for alpha, beta and gamma-radiation

## Super Arduino

Getting started with  
the chipKIT™ Max32



DesignSpark  
chipKIT™  
Challenge

## OnCE/JTAG Interface

Program and debug Freescale DSPs

## Simple Bat Detector

Inexpensive, sensitive and easy to build

US \$ 7.95 - Canada \$ 7.95



# ● **Subscribe** to *audioXpress* magazine!

Do your **electronics speak** to you? Are the words "**audio**", "**vacuum tubes**", and "**speaker technology**" music to your ears?

Then you should be **reading audioXpress!**

Recently acquired by The Elektor Group, *audioXpress* has been providing engineers with incredible audio insight, inspiration and design ideas for over a decade. If you're an audio enthusiast who enjoys speaker building and amp design, or if you're interested in learning about tubes, driver testing, and vintage audio, then *audioXpress* is the magazine for you!

## What will you find in *audioXpress*?

- In-depth interviews with audio industry luminaries
- Recurring columns by top experts on speaker building, driver testing, and amp construction
- Accessible engineering articles presenting inventive, real-world audio electronics applications and projects
- Thorough and honest reviews about products that will bring your audio experiences to new levels

Choose from print delivery, digital, or a combination of both for maximum accessibility.

Subscribe to *audioXpress* at  
**[www.cc-webshop.com](http://www.cc-webshop.com)**  
today!

**audioXpress**



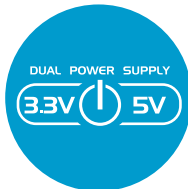


# IT'S HERE!

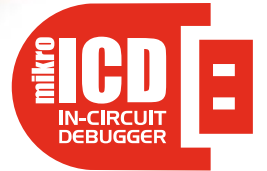


JUST PLUG IN ONE OF YOUR **CLICK BOARD™**, AND IT'S READY TO WORK. SPECIALIZED **MIKROBUS™** **COMMUNICATION INTERFACE** MAKES YOUR DEVELOPMENT EASIER, AND ALLOWS SIMPLE, YET HIGHLY EFFECTIVE CONNECTIVITY. EVERYTHING IS NOW **JUST A CLICK AWAY!**

## EasyPIC<sup>v7</sup> CONNECTIVITY



**\$149<sup>00</sup>**



POWERFUL ON-BOARD MIKROPROG PROGRAMMER AND IN-CIRCUIT DEBUGGER CAN PROGRAM ALL **PIC10, PIC12, PIC16 AND PIC18 MICROCONTROLLERS.**

OUTSTANDING PERFORMANCE AND EASY OPERATION WILL BLOW YOUR MIND. YOU WILL NEED IT, WHETHER YOU ARE A PROFESSIONAL OR A BEGINNER.



EASYPIC V7 IS THE ONLY DEVELOPMENT BOARD IN THE WORLD TO SUPPORT BOTH 3.3V AND 5V MICROCONTROLLERS. REVOLUTIONARY ENGINEERING ALLOWED US TO SUPPORT **OVER 250 MICROCONTROLLERS** IN A SINGLE BOARD. IT'S LIKE HAVING TWO BOARDS INSTEAD OF ONE!

Best selling PIC development board in the world enters its 7th generation of development. It is state of the art in design, functionality and quality. With 4 connectors for each port EasyPIC v7 has **amazing connectivity**. Ports are logically grouped with their corresponding LEDs and Buttons. Powerful on-board **mikroProg** In-Circuit Debugger and programmer supports over 250, both **3.3V and 5V devices**. Three types of displays, Serial EEPROM, two temperature sensors, Piezo Buzzer, USB connector, RS-232 and FTDI, Oscilloscope GND pins, as well as **mikroBus support** make this board an irreplaceable PIC development station.

## Great Expectations

Some call it “disappointing”, others “problematic to adapt to active marketing strategies” but I find it fascinating and a constant source of inspiration and amusement: projects and articles that start out insignificant, then taking off in a grand way and eventually reaching blockbuster status, apparently “self-propelled”. Elektor’s *Hexadoku*, the *Audio DSP Course*, *E-Weekly*, *The Chaos Machine*, *Pico C*, *Retronics* and the *ElektorBus* are fine examples of Elektor readers determining the rate of success rather than us editors and publishers.

Eight installments ago, the *ElektorBus* was no more than a vague idea loosely mentioned in an E-Labs Inside installment. Since then it has gained momentum and thanks to many active thinking caps out there we’re now at the stage of describing an ElektorBus control center running on a Smartphone. Likewise *Retronics*, kind of my own section in the magazine, started as a joke, but six years on it’s gathered a large group of loyal followers—and contributors! I’ve several ‘sensors’ available to gauge the success of published projects. Magazine and board sales reports are the obvious monitors that come to mind, but as an editor I find my weekly website statistics quicker and more up to date, as well as telephone calls and just plain enthusiastic emails. In good engineering spirit I find all manner of feedback useful and rewarding.

We kick off a global Challenge again, this time brought to you jointly by RS Components, Elektor and Circuit Cellar. It’s all about designing a project around the Digilent chipKIT™ Max32 hardware and software using RS’ DesignSpark PCB design tools, where ideally the three culminate in an add-on board that helps save or reduce energy any way you can think of. After the official launch of the Challenge on November 26 at the Elektor Live! event, a limited number of chipKIT™ boards will be given away to participants. Starting on page 16, Clemens Valens reports his initial findings with the chipKIT™ hardware and software components and you might find his story useful to be in pole position by the time the Challenge starts. I do expect your entry too!

Enjoy reading this edition,  
Jan Buiting, Editor



### 6 Colophon

Who’s who at Elektor.

### 8 News & New Products

A monthly roundup of all the latest in electronics land.

### 14 Super Arduino

This article should put you in pole position for the DesignSpark chipKIT™ Challenge organized by RS Components, Elektor and Circuit Cellar.

### 18 Improved Radiation Meter

Gamma, beta and alpha radiation levels get measured with this instrument based on a cheap and cheerful PIN photodiode.

### 24 Simple Bat Detector

Build this circuit over the winter period and next spring you’ll be able to listen to bats flying to and fro.

### 28 Audioguide

The first application of the Elektor Platino board is an RFID triggered rMP3 (no typo) player for use as a personal guide in museums

### 32 Audio DSP Course (5)

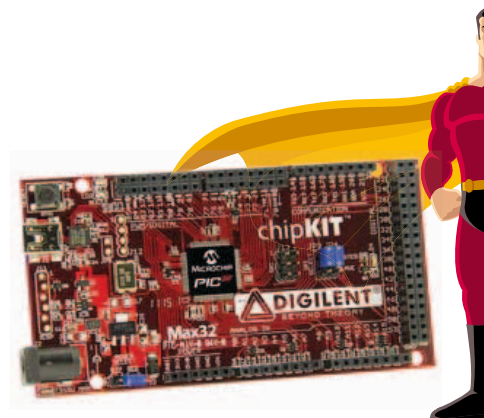
This month we take a look at the important matter of structuring your DSP programs.

### 40 OnCE/JTAG Interface

This adapter was originally designed for the Elektor DSP courseware board but is compatible with other Freescale DSPs too.

### 44 E-Labs Inside: Working with Stencils

A step by step guide to making perfect SMD boards as far as applying solder paste is concerned.





# CONTENTS

**Volume 3**  
**November 2011**  
**no. 35**



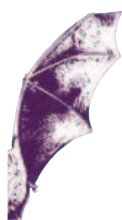
## 14 Super Arduino

The chipKIT™ Max32 board offers 32-bit computing power and some 80 I/O pins while remaining compatible with the Arduino environment. It is the hardware component of the exciting design challenge brought to you by RS Components, Circuit Cellar and Elektor. In this article you'll find a number of tips that should give you a head start in working with the hardware and software that go into making your entry for the challenge.



## 18 Improved Radiation Meter

The basic instrument described in this article can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive 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.



## 24 Simple Bat Detector

Many species of bat produce sounds at frequencies in the 40 kHz range, which happens to be the operating range of most standard ultrasonic transducers. If you amplify the signals picked up by one of these transducers and feed them into a frequency divider, the output will be within your normal range of hearing.



## 66 Temperature Gradient Meter

Measuring temperature is a fairly simple task, but doing so precisely takes rather more skill, particularly when it is desired to obtain readings to a resolution of better than one tenth of a degree. The circuit presented here measures temperature to a resolution of one ten-thousandth of a degree, using just four active components and an optional display.

## 46 Here comes the Bus! (9)

This month HTML and Javascript are used to make a control station running on a Smartphone.

## 56 Economical Voltage Monitor

This circuit monitors the voltage output of a solar cell while consuming precious little power itself.

## 58 Time Domain Reflectometry

When it comes to finding partial opens and shorts in wires and cables TDR is the way to go.

## 62 Spice It Up!

An introduction to LT's renowned simulator — so easy to follow it will get you hooked on Spice!

## 66 Temperature Gradient Meter

This instrument reports and records the tiniest temperature changes with a resolution of 1/1000th of a degree centigrade.

## 70 Resistive Bolometer

An innovative use of two electric bulbs you were about to trash.

## 72 Retronics: Exotic Tubes Facebook

Old friends hail from this month's Retronics pages. Series Editor: Jan Buiting

## 74 Gerard's Columns: Speaking Out Loud

From our monthly columnist Gerard Fonte.

## 76 Hexadoku

Elektor's monthly puzzle with an electronics touch.

## 84 Coming Attractions

Next month in Elektor magazine.

Elektor International Media provides a multimedia and interactive platform for everyone interested in electronics. From professionals passionate about their work to enthusiasts with professional ambitions. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital; practical and theoretical; software and hardware.



The Complete Source for Electronics: PROJECTS, TECHNOLOGY and EMBEDDED

November 2011

**elektor**

+ Time Domain Reflectometry

Exposure levels in check? Cut-rate Radiation Meter for alpha, beta and gamma radiation

Super Arduino Getting started with the chipKIT™ Max32

OnCE/JTAG Interface Program and debug Freescale DSPs

Simple Bat Detector Inexpensive, sensitive and easy to build

DesignSpark chipKIT™ Challenge

US \$ 9.95 • Canada \$ 12.95

ANALOG • DIGITAL  
MICROCONTROLLERS & EMBEDDED  
AUDIO • TEST & MEASUREMENT

No. 35, NOVEMBER 2011

ISSN 1947-3753

Elektor aims at inspiring people to master electronics at any personal level by presenting construction projects and spotting developments in electronics and information technology.

Elektor (ISSN 1947-3753) is published monthly (except for one issue in July/August) at \$39.95 per year, Canada add \$11.00 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](http://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.

International Editor:  
Wisse Hettinga ([w.hettinga@elektor.com](mailto:w.hettinga@elektor.com))

Editor: Jan Buiting ([editor@elektor.com](mailto:editor@elektor.com))

International editorial staff: Harry Baggen, Thijs Beckers, Eduardo Corral, Ernst Krempelsauer, Jens Nickel, Clemens Valens

Design staff: Christian Vossen (Head), Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser

Graphic design / DTP: Giel Dols, Mart Schroijen

Publisher: Hugo Van haecke  
([h.vanhaecke@elektor.com](mailto:h.vanhaecke@elektor.com))

Marketing: Carlo van Nistelrooy

Customer Services: [sales@elektor.com](mailto:sales@elektor.com)

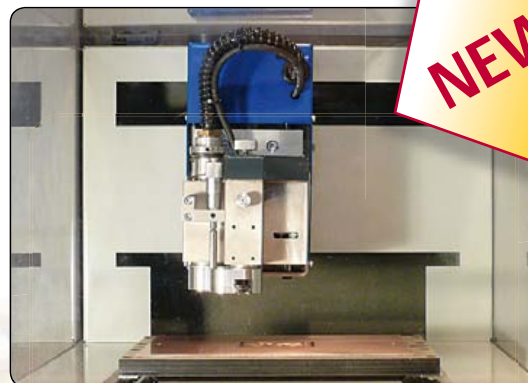
Subscriptions:  
Elektor US, 4 Park Street, Vernon, CT 06066, USA.  
Phone: 860-875-2199, Fax: 860-871-0411  
Internet: [www.elektor.com](http://www.elektor.com)  
E-mail: [subscriptionsus@elektor.com](mailto:subscriptionsus@elektor.com)



# Elektor PCB Prototyper

➔ A professional PCB router  
with optional extensions!

This compact, professional PCB router can produce complete PCBs quickly and very accurately. This makes the PCB Prototyper an ideal tool for independent developers, electronics labs and educational institutions that need to produce prototype circuits quickly. The PCB Prototyper puts an end to waiting for boards from a PCB fabricator – you can make your own PCB the same day and get on with the job. In addition, the PCB Prototyper is able to do much more than just making PCBs. A variety of extension options are available for other tasks, and a range of accessories is already available.

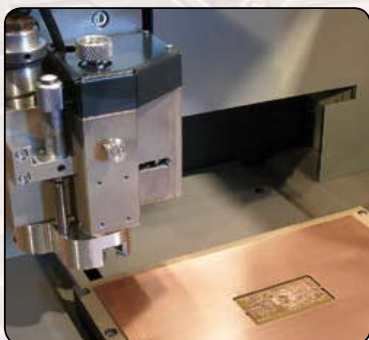


## Specifications

- Dimensions: 440 x 350 x 350 mm (W x D x H)
- Workspace: 220 x 150 x 40 mm (X x Y x Z)
- Weight: approx. 35 kg (78 lbs)
- Supply voltage: 110–240 V AC, 50/60 Hz
- Integrated high-speed spindle motor; maximum 40,000 rpm (adjustable)
- Integrated dust extraction (vacuum system not included)
- USB port for connection to PC
- Includes user-friendly Windows-based software with integrated PCB software module

## Ordering

The complete machine (including software) is priced at US \$4,900 plus VAT and shipping charges (please enquire at [sales@elektor.com](mailto:sales@elektor.com)).



**elektor**

Further information and ordering at  
[www.elektor.com/pcbprototyper](http://www.elektor.com/pcbprototyper)



## 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,  
1187 Washington St., Gloucester MA 01930 USA.  
Phone: 978-281-7708, Fax: 978-281-7706  
E-mail: [peter@smmarketing.us](mailto: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.

## Disclaimer

Prices and descriptions of publication-related items subject to change. Errors and omissions excluded.

© Elektor International Media b.v. 2011 Printed in the USA

## Compact, high-Efficiency RF Power amplifier for 5 GHz Wi-Fi® applications

Microchip Technology Inc., recently announced the new SST11CP15 RF power amplifier for 5 GHz IEEE 802.11a/n WLAN embedded applications. The device operates on the 4.9 to 5.9 GHz band, and offers a wide operating voltage of 3.3 V to 5 V. The SST11CP15 features a high linear output power of 18 dBm at 2.5 percent EVM, using 802.11a OFDM 54 Mbps at 3.3 V, and 20 dBm at 5.0 V, and offers an output power of 23 dBm at mask compliance of 6 Mbps, at 3.3V. The device is offered in a compact, 2 x 2 x .55 mm, 12-pin QFN package. It is ideal for 5 GHz WLAN applications where small size and high-efficiency operation are required, such as in wireless multimedia and MIMO applications for broadband gateway and consumer-electronics equipment. The SST11CP15 meets the needs of designers who must reduce DC current consumption in their portable multimedia and MIMO appli-



cations. With its high power-added efficiency, the device reduces battery current drain and extends battery operation. Its 4.9 to 5.9 GHz linear operation enables 802.11a/n operation and increases data rates, while its small size is ideal for space-constrained applications.

Developers can begin designing today with the SST11CP15 Evaluation Board (part # 11CP15-QUBE-K), which is available now, via any Microchip sales representative.

The SST11CP15 RF power amplifier is available in a 2 x 2 x 0.55 mm, 12-pin

QFN package. Samples are available today, at [www.microchip.com/get/TNQ6](http://www.microchip.com/get/TNQ6). Volume-production quantities can be ordered today at [www.microchip.com/get/9JD6](http://www.microchip.com/get/9JD6). For additional information, contact any Microchip sales representative or authorized worldwide distributor, or visit Microchip's website.

[www.microchip.com](http://www.microchip.com) (110604-VI)

## New multi-function ASB DAQ modules

ADLINK Technology, Inc. announces the release of its USB-1900 Series and USB-2401 USB DAQ modules. Equipped with built-in signal conditioning, the USB-powered Plug and Play USB DAQ modules deliver easy connection and accurate results for both portable measurement and machine automation applications. Featuring built-in signal conditioning, ADLINK USB DAQ modules enable direct measurement of most frequently applied signal sources, which reduces manpower requirements and associated development costs while increasing overall accuracy.



All of ADLINK's USB DAQ modules feature USB power and removable screw-down terminals for simplified device connectivity, and a multi-functional stand for fast and

easy desktop, rail, or wall mounting. Additionally, a lockable USB cable secures connectivity. The USB DAQ modules also provide device ID setting by a rotary control for convenient identification of the active module in multiple-connection configurations. ADLINK's USB DAQ collection offers the USB-1900 series, consisting of USB-1901 and USB-1902 models of 16-bit 250 kS/s DAQ modules. Also in the series is the USB-1903, with additional built-in precision current-to-voltage resistors allowing direct measurement of current signals from 0 to 20 mA. Rounding out the USB DAQ category is the USB-2401, a 24-bit four-channel simultaneous-sampling universal DAQ module supporting sampling rates up to 1.6 kS/s and a more flexible signal conditioning circuit such as voltage, current, strain, load cell, thermocouple, and RTD measurement. The USB DAQ line is ideally suited for easily accomplished deployment and superior accuracy when measuring temperature, stress, strain, and other factors in diverse applications.

The USB DAQ modules include ADLINK's free U-Test utility, allowing direct operation and testing of all functions with no requirement for coding or programming. Driver support is provided for Windows 7/Vista/XP, in both 32-bit and 64-bit versions, and 3rd party software support accommodates LabVIEW & MATLAB.

[www.adlinktech.com/USBDAQ](http://www.adlinktech.com/USBDAQ) (110675-II)

## Power controllers for next-generation server, desktop and computing applications

International Rectifier has introduced a new digital power platform that dramatically improves energy efficiency in a wide variety of applications, including high performance server, desktop and computing applications.

The new family of digital controllers is based on CHiL's proven digital platform offering full telemetry and programmability, providing system designers a chance to differentiate their products with custom features. The family offers a fully compliant high speed serial bus to meet new industrial requirements. These 5, 6 and 8-phase dual output PWM controllers, in which phases are flexibly assigned between loops 1 and 2, can be easily configured for Intel® VR12, AMD® SVI/PVI/G34, and feature switching frequency between 200 kHz to 1.2 MHz per phase.

The family of digital controllers offers efficiency improvements with features such as variable gate drive and dynamic phase control for best-in-class efficiency and programmable 1-phase or 2-phases for light loads. When used in conjunction with IR's PowIRstage® devices this family offers an optimized end-to-end solution delivering highest efficiency for next-generation servers.





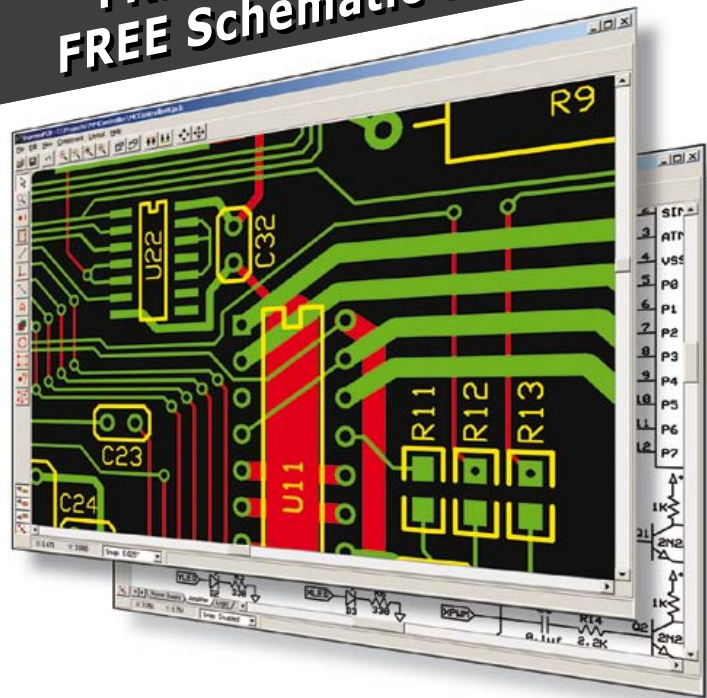
Other key features of the new digital controllers include adaptive transient algorithm (ATA) on both loops to minimize output bulk capacitors and system cost, auto-phase detection with auto-compensation and per-loop protection features. In addition, the digital ICs offer I<sup>2</sup>C, SMBus and PMBus system interface for full telemetry and programmability. Non-volatile memory for custom configuration, 3.3 V tri-state driver compatible, +3.3 V supply voltage and 0°C to 85°C ambient operation are also featured.

All CHiL digital power solutions are fully supported by IR's Digital Power Design Center (DPDC) GUI at [www.irf.com](http://www.irf.com) which facilitates design, development and deployment of the company's digital solutions. Additionally, hardware support using IR's proprietary SVID emulator system enables customers to emulate and monitor either Intel or AMD serial interface protocols, as well as high speed I<sup>2</sup>C communication.

[www.irf.com](http://www.irf.com) (110675-V)

# \$51<sup>For 3</sup> 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](http://expresspcb.com)

## New series of free schematic symbols and PCB footprints for DesignSpark PCB design tool

RS Components' (RS) and Accelerated Designs' new series of component libraries provides customers with schematic symbols and Printed Circuit Board (PCB) footprints for an extensive range of products from STMicroelectronics and Microchip Technology.

Thousands of PCB footprints and schematic symbols are available for free download in a vendor-neutral format from RS' online DesignSpark electronics design community and resource center, and can be exported to virtually any EDA and CAD/CAE system using Accelerated Designs' Ultra Librarian (UL) translator software, thus saving the design engineer valuable time and effort in the CAD design process. The UL Reader also supports RS' free DesignSpark PCB design tool that offers powerful schematic capture and PCB layout software.

The free Ultra Librarian software, including footprints and symbols, is available now for download from the Designspark website. Bill of Materials (BOM) reports can be generated and prices quoted using RS' free Online Quotes tool.



[www.designspark.com](http://www.designspark.com) (110604-VII)

## Digital differential pressure sensors with very low measurement ranges

Sensirion has recently launched two new versions of its differential pressure sensors with-in the SDP600 series. The new sensors SDP6x0-125Pa and SDP6x0-25Pa feature particularly low measurement ranges of  $-125$  to  $+125$  Pa and  $-25$  to  $+25$  Pa, respectively. Due to their better resolution, the new products have improved zero point accuracy of  $0.1$  Pa. In addition, zero point repeatability reaches a remarkable  $0.05$  Pa for the  $125$  Pa version and  $0.03$  Pa for the  $25$  Pa version. The new differential pressure sensors are excellent solutions for many applications in medical and HVAC markets where high accuracy measurements of particularly low differential pressure are required.



The two versions complement the comprehensive product range of Sensirion's digital differential pressure sensors in the SDP600 series. Along with the other products of this series, they offer a digital I<sup>2</sup>C output and are fully calibrated and temperature compensated. Thanks to the principle of calorimetric flow measurement, the CMO-Sens® differential pressure sensors achieve outstanding sensitivity and accuracy, especially around the zero point. Furthermore, the sensors exhibit very high long-term stability and freedom from zero-point drift.

[www.sensirion.com/sdp6x0](http://www.sensirion.com/sdp6x0)

[www.sensirion.com/differential-pressure-sensor-sdp6x0-datasheet](http://www.sensirion.com/differential-pressure-sensor-sdp6x0-datasheet)  
(110675-III)

## Wireless QWERTY keypad-equipped DUAL IR/RF remote control

Ultra low power (ULP) RF specialist Nordic Semiconductor ASA announced that

## iPad, iPod touch, and iPhone turned into spectrum analyzer or dynamic power meter...or both

Oscium's breakthrough product line for the iOS Test industry enables iPad, iPod touch, and iPhone to now become either a spectrum analyzer or a dynamic power meter...or both. Oscium's first-to-market product, iMSO-104, successfully merged a mixed signal oscilloscope and the iOS family of products using the 30-pin dock connector. The contribution was so significant that Cypress Semiconductor Corp heralded the product as 'revolutionary'. Oscium's new product line, called WiPry, is the next installment in modular test equipment. This new category of test equipment has the potential to change the benchtop dominated landscape by establishing the touchscreen-based iPad (or iPhone, iPod touch) as the new user interface. This platform presses the refresh button on the antiquated buttons and knobs of benchtop instruments while at the same time offering mobility that PC-based instruments cannot match.

Three distinct products fit under the WiPry brand: WiPry-Spectrum, WiPry-Power, and WiPry-Combo (which combines the functionality of both WiPry-Spectrum and WiPry-Power).

WiPry-Spectrum leverages the colorful potential of the OpenGL interface on the iOS platform for stunning real-time views of RF activity in the  $2.4$  GHz ISM band. WiPry-Power crosses the chasm of this new platform by not only graphically displaying RF data from  $100$  MHz –  $2.7$ GHz but also adding the ability to capture, trigger and record the actual power output of RF amplitude. An optional accessory kit is also available that boosts the products ability by giving the user the ability to make conducted measurements. The final product combines all the features of both WiPry-Spectrum and WiPry-Power into one product called WiPry-Combo.

By merging a spectrum analyzer and a dynamic power meter into the iOS Test industry, Oscium is opening the door for a more productive and useful mobile platform.

WiPry-Spectrum costs \$99.97, WiPry-Power \$149.97, and WiPry-Combo \$199.97. WiPry is compatible with all generations of iPod touch, iPhone, and iPad devices running iOS version 3.1.3 or higher. It is made for: iPod touch (1st, 2nd, 3rd, and 4th generation), iPhone 4, iPhone 3GS, iPhone 3G, iPhone, iPad 2, and iPad.

[www.oscium.com](http://www.oscium.com) (110675-I)



Philips Home Control, Singapore, is using Nordic wireless technology in its advanced QWERTY keypad-equipped DUAL infrared (IR)/RF remote control.

DUAL is designed for use by consumer electronics (CE) manufacturers of emerging 'connected' products such as Smart TV, Over The Top (Internet) boxes, and Hybrid set-top boxes (STB). The DUAL platform comprises everything required for CE manufacturers to develop a customized IR/RF remote control with minimum design overhead, and includes controller handset, com-

pact USB dongle, and demonstration software that runs on a PC.

DUAL is equipped with a full QWERTY-keypad on one side, and on the other, touchpad or optical sensor controls for alternate input methods allowing manufacturers to implement free cursor, gesturing, and moving highlight mechanisms. Philips Home Control says this makes browsing a better experience than traditional RF remote controls with directional keys.

Traditional remote controls struggle to meet the demands of modern consumers





due to limited bandwidth and 'one-button-one-operation' interfaces. RF provides sufficient bandwidth for advanced navigation interfaces — such as scroll wheels, touch screens, and track balls — and bidirectional communication required when negotiating the complex user interfaces and menus typical of modern media devices. In addition, the QWERTY keypad is useful for browsing the Internet on the latest generation of

Smart TVs and other connected products. RF eliminates the need for IR's line-of-sight access, allowing devices to be controlled in the presence of obstacles and even interior walls (up to a range of 15m and assuming wall construction materials do not excessively attenuate the RF signals). DUAL remote control also incorporates IR remote control functionality so that users can operate legacy entertainment devices. In operation, the DUAL remote handset utilizes a Nordic nRF24LE1 System-on-Chip (SoC) 2.4 GHz ULP transceiver running a modified version of Nordic Gazell RF protocol software. An nRF24LU1+ System-on-Chip (SoC) 2.4 GHz ULP transceiver and USB 2.0 compliant device controller, incorporated into a compact USB dongle, plugs into the host device (the product to be controlled) to form the other node of the wireless link. The Nordic RF technology enables a bidirectional communication link with sufficient bandwidth for rapid screen refresh and seamless navigation.

The nRF24LE1 integrates a proven nRF24L01+ transceiver core, enhanced 8051 microcontroller, 16 Kbytes of on-chip flash and 1 Kbytes of SRAM into a single-chip solution. The chip boasts a 2 Mbps on-air data rate combined with ultra low power (ULP) operation and advanced power management. The nRF24LU1+ integrates a USB 2.0 compliant device controller, 8-bit application microcontroller, and nRF24L01+ compatible 2.4 GHz RF transceiver. Gazell RF protocol software provides features for advanced navigation, remote data transfers, and advanced pairing schemes while handling up to five remote devices at the same time. In addition, Gazell is a frequency agile protocol that is highly immune to interference from other 2.4 GHz radio sources. DUAL from Philips Home Control is now ready for commercialization.

[www.nordicsemi.no](http://www.nordicsemi.no) (110675-VI)



pico<sup>®</sup>  
Technology



**PicoScope<sup>®</sup> 4262**

HIGH-RESOLUTION OSCILLOSCOPE

A Digital Oscilloscope for the Analog World

PICOSCOPE 4000 SERIES



# 16 bit

Low noise • Two channels  
16 MS buffer • 16-bit resolution  
10 MS/s sampling • 5 MHz bandwidth  
Advanced digital triggers  
Low-distortion signal generator  
Arbitrary waveform generator  
USB powered

## Analyze:

- Audio
  - Ultrasonic
  - Vibration
  - Distortion
  - Noise
  - Power Supplies
  - Anything Where
- High Accuracy is a Must



**Call Us At:**

**1-800-591-2796**

[www.picotech.com/pco462](http://www.picotech.com/pco462)



*The industry's brightest minds  
come here to shine.*

Every year one place brings together the innovators on technology's cutting edge.  
And for those four days, Vegas glows brighter.



Tuesday, January 10–Friday, January 13, 2012  
Las Vegas, Nevada | [CESweb.org](http://CESweb.org)



REGISTER NOW





DESIGNSPARK

Turn a **hot**  
**idea** into a  
**cool**  
**solution.**

## DesignSpark chipKIT™ **Challenge** Contest launches **November 28, 2011!**

Challenge your talent against other engineers worldwide to produce an energy-efficient design solution using the free DesignSpark PCB software and Microchip's chipKIT™ development board.

Achieve the most energy-efficient design and **you could win a share of \$10,000!**

Plus, keep the DesignSpark community regularly informed through posts on the DesignSpark Project Pages and your updates will make you eligible for Community Choice Awards and random prize drawings!



**FREE** chipKIT™ Max32™ development kit for qualified engineers.

Visit **[www.chipkitchallenge.com](http://www.chipkitchallenge.com)**

for complete rules and to see if you qualify for a

**FREE** ChipKIT™ Max32™ development kit.\*

\*Subject to availability.

IN ASSOCIATION WITH:



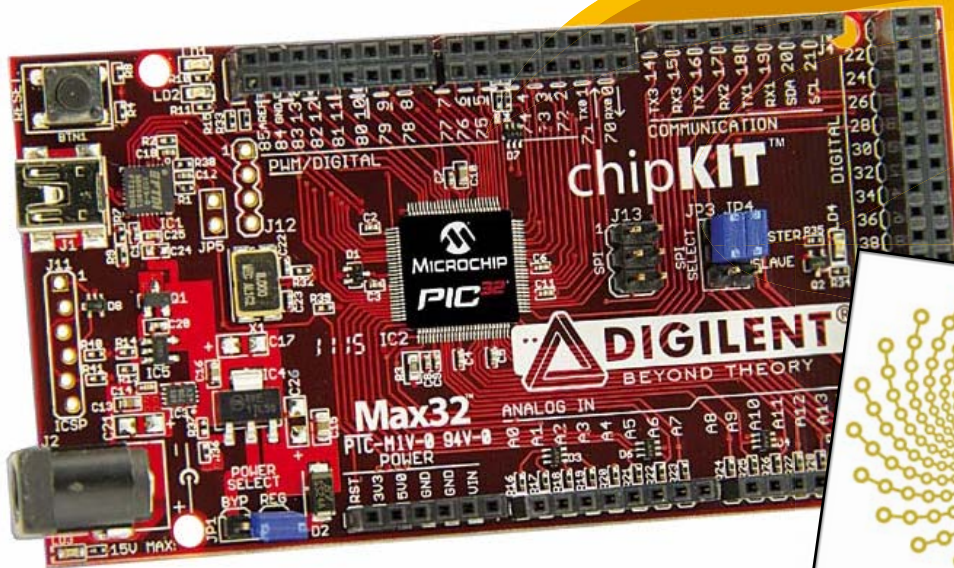
chipKIT™ is a registered trademark of Microchip Technology Inc.

Max32™ is a registered trademark of Digilent, Inc.



# Super Arduino

## Getting started with the chipKIT™ Max32



By Clemens Valens (Elektor France)

If you are interested in microcontrollers you probably have heard of Arduino, maybe you even worked with it. If you did, you may have run into the limits of this charming 8-bit platform and wished it had more computing power, memory or even I/O pins. With microcontroller boards galore that's within easy reach but at the cost of learning new tools all the time. Until recently, when Digilent introduced their solution for those wanting more power without changing tools. Their chipKIT™ Max32 board offers 32-bit computing power and some 80 I/O pins while remaining compatible with the Arduino environment.

For sure there have been previous attempts at making 32-bit Arduino compatible boards, however as far as I know these attempts only achieve Arduino form factor compatibility, not compatibility at the tools level. Some of these boards are supported by software libraries that offer Arduino-like functionality and syntax, but each one requires a different compiler and employs specific firmware loading methods. Digilent has taken Arduino compatibility a step further by integrating the compiler, linker and programmer for the PIC32 processor used on their chipKIT boards

into the real Arduino 0022 integrated development environment (IDE from now on). From an Arduino IDE point of view the chipKITs are just targets, cheerfully listed alongside the classic 8-bit Arduino boards. Digilent even went so far as to create a website with a URL ending in .cc [1], just like Arduino.

Also in the spirit of Arduino, the chipKIT comprises completely open source and hardware, meaning the hardware design files (Eagle schematic and PCB files) are available for downloading and the software is all open source. Unlike Arduino the

chipKIT PCB is a 4-layer board, so few people will be tempted to roll their own.

There are two flavors of chipKIT: the Uno32 and the Max32. Mechanically the Uno32 is compatible with the classic Arduino Uno and the Max32 is compatible with the Arduino Mega, which is the longer version of the standard Arduino. Note that Digilent made the boards rectangular by straightening the curiously shaped short edge of the Arduinos so they are slightly larger. I don't think this will be a problem for anyone.

The rest of this article will concentrate on the Max32, so here we go!



## The board

The Max32 comes as a red 4-layer PCB in a tiny mainly red and white box without much else, i.e. no USB cable, no documentation, just a URL [2]. For those who do not know the Arduino Mega dimensions by heart: the board measures 10.2 x 5.4 cm. Like on the Arduino Mega, three edges of the board are lined with connectors, except that the connectors for the digital pins 0 to 13 (in Arduino-speak) are double-row types on the Max32. You will find digital pins 70 to 85 on the extra contacts. On the USB and power connector side there is room for a Microchip ICSP programming connector. This connector has the special Sparkfun “out-of-line” (staggered) placement of the pins allowing a pinheader to make proper contact even without being soldered.

Powering the board can be done through the USB connector or using the power barrel jack which will allow input voltages up to 15 VDC. Jumper JP1 lets you bypass the 5 V regulator so be careful what you do here or you may blow up the odd chip. When you power a virgin board you will notice an annoyingly bright red LED next to the power

jack indicating that the 3V3 rail supplies a voltage while a green LED (LD4) flashes at about 3 Hz.

The Max32 is hardly more than a breakout board (BoB) for the 100-pin PIC32 processor mounted close to the centre of the board with a power supply and a USB serial port on the side. This PIC32 is a 32-bit microcontroller from Microchip that compares pretty well to ARM’s Cortex-M3 (see inset). The board sports the largest PIC32 device currently available, the PIC32MX795F512L. This chip sits in a 100-pin package and features 512 KB of flash memory with 128 KB of RAM and runs from an 80 MHz clock. It has USB-OTG (on-the-go), an Ethernet MAC and two CAN controllers.

After this short introduction of the hardware, let’s take a look at the software.

## The IDE

As already mentioned, programming the board is done using a modified Arduino IDE you can download as a free archive from [2]. Installing the 128 MB file is very simple; you only have to unpack it to a convenient

place on a convenient disk on your computer. Unpacked it takes up some 480 MB of disk space. To start the IDE launch the executable called **mpide.exe** found in the root of the IDE folder. The IDE is ‘cross-platform’ and will work happily on Windows, Linux and MacOS although you may have to install Java first.

By basing the Max32 on the Arduino IDE, the people at Digilent saved themselves a lot of documentation writing. Indeed, all you need to know to install and get started with Arduino is explained in detail on the Arduino website [3]. Also go there with your questions about the programming language syntax.

At the time of writing this article the version of the IDE is 0022 (mpide-0022-chip-kit-win-20110619 to be precise), the same as the current official Arduino IDE. According to Digilent this IDE is identical to the official Arduino IDE except that it has been extended with a PIC32 compiler/linker and libraries, hence can be used to program 8-bit Arduino boards too. Of course I tried this only to discover that my Arduino clone, a Seeeduno v1.1, was not rec-

## Introducing the PIC32

When asked about 32-bit microcontrollers most people will first mention ARM, then some ARM-core implementers like Atmel, ST or NXP and only a few will think of Microchip. Although it is true that many cellphones rely on ARM technology, lots of other consumer electronic devices like digital cameras and printers contain MIPS-based processors. Now I haven’t done any serious checking on this, but it is said that there are far more 32-bit MIPS processors out there than there are ARM processors. Having some MIPS experience is therefore a good thing for any serious microcontroller enthusiast and the PIC32 is an excellent platform to get started.

There are currently five families: 3xx, 4xx, 5xx, 6xx and 7xx. The 3xx and 4xx are considered general purpose, whereas the other three have more peripherals like CAN or Ethernet and can have more RAM. The devices are based on a 32-bit MIPS MK4 core with a 5-stage pipeline and support clock frequencies up to 80 MHz. A maximum performance of 1.56 DMIPS/MHz (Dhrystone 2.1) is claimed, which is slightly better than an ARM Cortex-M3 that can do 1.25 DMIPS/MHz, according to ARM.

All families have up to 512 KB flash memory plus 12 KB boot memory and up to 32 KB RAM for the 3xx/4xx devices or up to 128 KB

for the 5xx/6xx/7xx devices. They feature all the peripherals you would expect from this kind of microcontroller (serial ports, PWM, ADC, etc.) but they also have multiple direct memory access (DMA) channels.

The families come in two ‘sizes’, 64 pin (H suffix) or 100 pin (L suffix). Note that a 121 XBGA package contains a 100 pin device. The PIC32s are pin compatible with some PIC24 and dsPIC devices so they integrate nicely in Microchip’s vast microcontroller range and development tools (MPLAB). Lots of software libraries are available on the website and a dedicated website exists for sharing PIC32 projects ([www.mypic32.com](http://www.mypic32.com)). Datasheets and other documentation can be accessed through the URL [www.microchip.com/pic32](http://www.microchip.com/pic32).

Family	USB OTG	CAN	Ethernet	RAM
3xx	–	–	–	Up to 32 KB
4xx	1	–	–	Up to 32 KB
5xx	1	1	–	Up to 128 KB
6xx	1	–	1	Up to 128 KB
7xx	1	2 (1 for 764)	1	Up to 128 KB



Figure 1. The Max32 IDE showing a `#define` to remove AVR specific code.

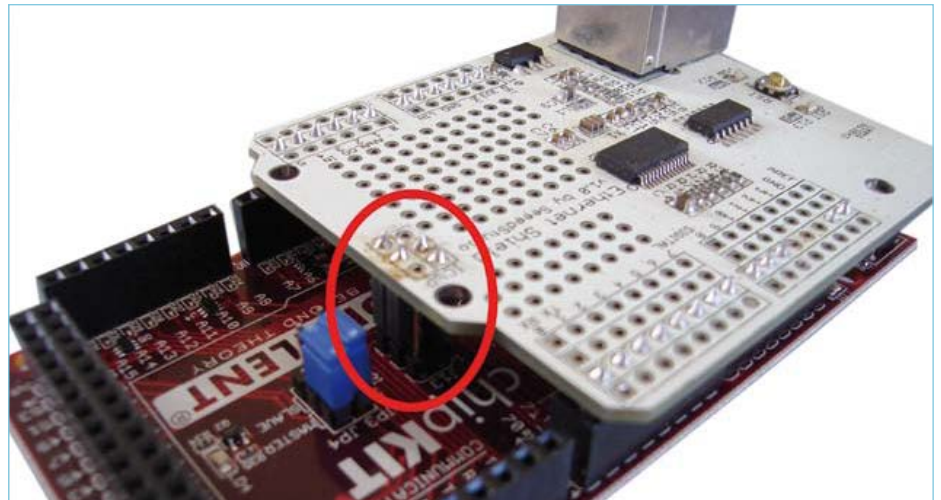


Figure 2. A close-up of the SPI connection between the etherShield and the Max32.

ognized: “Invalid device signature”. This board works perfectly fine with the official Arduino 0022 IDE.

After installing the IDE and hooking up the Max32 to the computer you can try out the tools by compiling one of the simple examples included with the IDE and uploading it to the board. Do not forget to select the Max32 board from the Tools -> Board menu and to set the serial port properly (Tools -> Serial Port). Once done the BlinkWithoutDelay example (File -> Examples -> Digital) should work without modification, just click the Upload button to see the green LED LD4 start to flash at a rate of 0.5 Hz.

If you got this far without problems you’re all set to develop real applications for the Max32. So read on, because there will be some hurdles that you may want to know of...

## Porting a shield

Flashing an LED is nice, but not very satisfying, that’s why I ventured on to try out my Arduino Ethernet shield on the Max32 (a shield is an extension card for an Arduino board). This shield is based on Microchip’s ENC28J60 stand-alone Ethernet controller with SPI interface. I know, the PIC32 has an Ethernet MAC inside, but I didn’t have a shield handy with an Ethernet PHY and RJ45 connector. Digilent proposes such a shield (that offers other things besides), but I didn’t have time to order one and wait for it to arrive. Besides, now I had a good opportunity to see just how Arduino compatible the Max32 is. As you will see below, not entirely...

My old Ethernet shield — I will call it etherShield from now on to distinguish it from the new official Arduino W5100-based Ethernet shield — is supported by a library and some examples. This shield and code works just fine on my Seeeduino. The first step was therefore to install this library in the Max32 IDE (in the folder `libraries\`) and see if it would compile. The answer was, as you may have expected: No(pe). The reason was not so much the code itself, but the fact that the compiler apparently was unable to find anything to compile. According to the Digilent website, where a few things about porting Arduino code are explained, libraries are supposed to be handled in the same way as Arduino does, but clearly not in my case. When I put the library in the folder `hardware\pic32\libraries\` (where you find the same items as in the folder `libraries\`) the compiler did find the code, but produced many errors and a warning saying that the code contained AVR specific code (Arduino boards are based on Atmel’s AVR processors). Great! Now I had clue.

The first thing you have to do when porting Arduino libraries is to get rid of the references to program memory. With an AVR some special compiler directives are needed for accessing constants (strings, tables) located in program space. This is unnecessary for the PIC32 and these directives have to be removed. To keep your code Arduino compatible it may be better to `#define` them away, you can use the macro `_BOARD_MEGA_` (see **Figure 1**) defined by the Max32 IDE to do this (confusing; you’d

have expected something like `_BOARD_MAX32_`). Do the same for any AVR-specific `#include` directives.

This may not be sufficient (in my case it wasn’t) because the library may also refer to AVR registers that the PIC32 doesn’t have. The SPI driver for the ENC28J60 Ethernet chip did this, probably because it is relatively old and the SPI library that’s now part of the Arduino IDE simply did not exist when at the time it was written (it did not appear before version 0019 from September 2010). I therefore modified the old etherShield library to use the new Arduino SPI library instead and tested this first in the real Arduino IDE before trying it in the Max32 IDE.

This introduced new problems because the SPI library references pin functions that the Max32 compiler did not like. As it turned out, the hitch was caused by my library being written in C and compiled as such, whereas the pin functions and the SPI library are written in C++. Files with the extension `.c` are compiled as C, files with the extension `.cpp` are compiled as C++. So, back to the Arduino IDE it was for porting the complete etherShield library to C++ and testing it. This was not particularly difficult to do, but you may have to watch out for compiler directives like `#extern "C" { ... }` hidden in unexpected places. After this last modification my etherShield library compiled without errors in the Max32 IDE and for the Max32 board. But did it work?

No — of course not! This did not really come



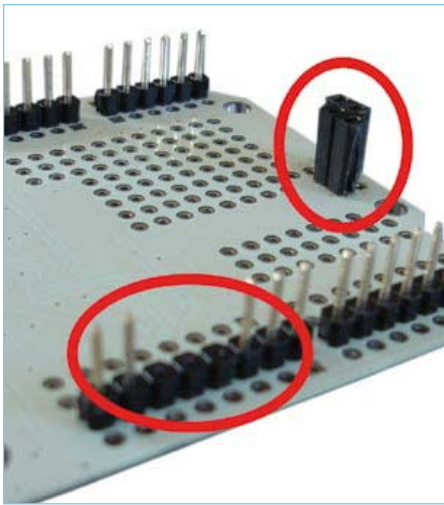


Figure 3. The modifications I made to the etherShield.

as a surprise as I had already spotted SPI problem posts on Digilent's website, but the optimist in me kept hoping.

The main problem is incompatibility of Max32 I/O pins with AVR I/O pins. Arduino digital pins 10 through 13 have SPI capability and pins 10 and 11 can do PWM also. The reason for this combination is simply the way the AVR I/O pins were laid out by Atmel. The PIC32 combines functions in a different way on its I/O pins and there are no exact equivalents for these AVR pins. Digilent has chosen to give priority to the PWM functions as they are used by Arduino's analogWrite functions, meaning that they could only connect part of SPI port 2 to these pins. However they did find a way to connect SPI port 1 in an Arduino compatible manner by using the Arduino ICSP connector (**Figure 2**) that's connected to the same signals as pins 10 through 13. I had never considered this connector as part of a compatible shield, and I am in good company, but Seeedstudio, the maker of my etherShield, had put an ICSP pin header on the shield in the right place. Replacing it with a female (socket style) connector on the solder side only took a few minutes and restored SPI compatibility with the Max32. To prevent conflicts on pins 11, 12 and 13 (MOSI, MISO & SCK) I simply removed these pins from my shield (**Figure 3**). Now it should work, right? Wrong. At this point I fired up the oscilloscope because I suspected compatibility issues between the SPI protocol as produced by the PIC32 and the one accepted by the ENC28J60. Notably the clock speed was worrying me slightly. The scope proved me

right, because where the Arduino managed to reach a clock speed of about 610 kHz, the PIC32 was blasting away at 20 MHz. According to the datasheet of the ENC28J60 this should be OK, but experimenting later when all was finally working showed me that 2.5 MHz was a more realistic value. For now I just reduced the PIC32's clock speed to an Arduino-like value: 625 kHz. With this change the shield still didn't work, but I felt I was getting close.

Nowadays even the lowest budget digital oscilloscopes can record traces, including my 240-euros (shipping included) 25 MHz Atten ADS1022C, and this very useful function showed me that there was a polarity/phase issue between the SPI clock and data lines. By carefully comparing transitions I discovered that the shield needed SPI mode 1 on the Max32 whereas it used mode 0 on the Arduino. Did that mean that the shield was now finally working? It did (**Figure 4**). Phew.

### Final remarks

Digilent have made a decent job of porting the PIC32 to the Arduino IDE. Although they did not achieve 100% compatibility, they definitely tried hard and managed to get pretty close. It's safe to assume that simple Arduino shields with simple libraries respecting Arduino coding rules and style will be easily portable, although you may run into some of the problems I encountered. The more complex shields exploiting AVR subtleties will definitely be more difficult and may require solid PIC32 knowl-

edge. To keep your life easy, try to use the functions available in Arduino libraries as much as possible, and let Digilent do the hard work for you.

Digilent has started keeping a list of known-to-work shields, so first look there before starting a project yourself. Updates to the Max32 IDE correcting some of the issues mentioned in this article may be expected, so make sure you always use the latest version.

The only thing that I haven't completely cleared up yet is the issue of where to put your own libraries. After thinking hard and experimenting a bit I settled on the assumption that all files that contain PIC32 specific code, like low-level drivers, must be placed in the folder `hardware\pic32\libraries\` including the files that need these files. All other files including the examples that use the library must be placed in the folder `libraries\` to ensure they are recognized by the IDE as examples.

The source code files for the tests and experiments described in this article can be downloaded from [4].

(110661)

### Internet Links and Resources

- [1] <http://chipkit.cc>
- [2] [www.digilentinc.com](http://www.digilentinc.com)
- [3] <http://arduino.cc>
- [4] [www.elektor.com/110661](http://www.elektor.com/110661)

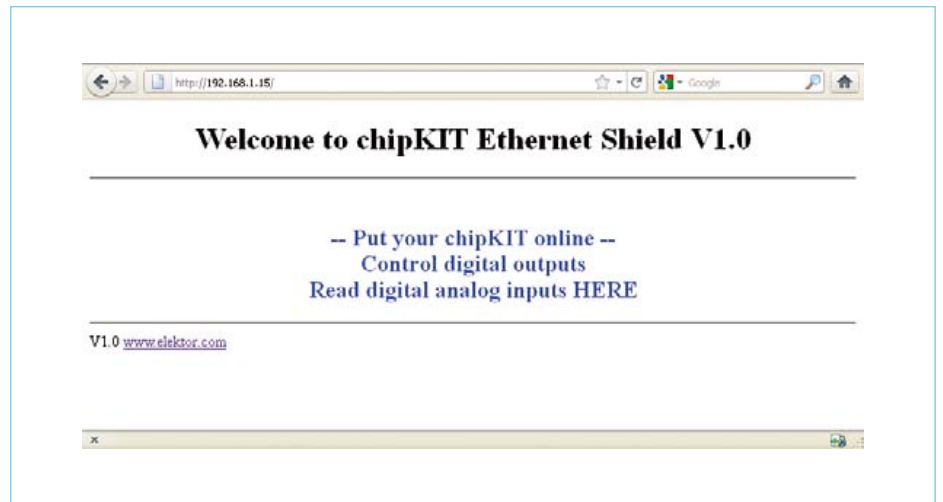


Figure 4. The result of a successful port: I can now connect to a tiny web server running on the Max32.

# Improved Radiation Meter Counter for alpha, beta and gamma radiation

By Burkhard Kainka (Germany)

All that's required to measure radiation is a simple PIN photodiode and a suitable preamplifier circuit. We present here an optimized preamplifier and a microcontroller-based counter. The microcontroller takes care of measuring time and pulse rate, displaying the result in counts per minute.



The device we describe can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive 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, allowing

a more detailed investigation of the characteristics of a sample. The optional PC-based software displays the energy spectrum, permitting a very detailed analysis to be carried out.

## Preamplifier

In the June 2011 issue of Elektor we described experiments using a BPW34 photodiode as a detector for gamma radiation [1]. Only simple experiments could be carried out as the pulses from the detector are very short. Here we look at an improved preamplifier design, which avoids the use of

a comparator, and which generates output pulses that can be heard directly or processed further. The amplifier uses a BF245B JFET at its input followed by an opamp circuit, providing an overall voltage gain of 1 000. At the output it delivers pulses with an amplitude of up to 200 mV with a pulse width of around 0.5 ms, which can be rendered audible without further processing or which can be used to drive a counter.

The circuit (see **Figure 1**) can use several photodiodes in parallel at its input. On the one hand, this increases the pulse rate; unfortunately, this advantage is on

## Elektor products and services

- Printed circuit board: # 110538-1
- Kit (components and circuit board): # 110538-71
- FT232R USB/Serial Bridge: # 110553-91
- PDF layout: free download at [2]
- Software and firmware: free download at [2] (file # 110538-11)



## Features

- Measures alpha, beta and gamma radiation
- Simple construction using standard components
- PC connection using Elektor FT232R USB/Serial bridge
- Threshold configurable in software from the PC
- Two different types of sensor can be used

the other hand offset by the reduced amplitude of the sensor's output due to its increased total capacitance.

The JFET input offers good signal-to-noise ratio and a high input impedance. A DC level of about 2 V or 3 V appears across the

BF245B's source resistor, more or less independent of the supply voltage. If a BF245C is used, this voltage will be higher; in the case of a BF245A, lower. This voltage level is a suitable operating point for the opamp. The capacitance of the photodiode drops as the voltage across it increases, and so it is operated at the full supply voltage, with

the gate of the JFET being pulled to ground through a 20 M $\Omega$  resistance.

### The counter

The pulse counter is constructed using an ATmega88 and a two-line LCD. The power supply voltage of 9 V to 12 V is taken via D1 (for reverse polarity protection) to voltage

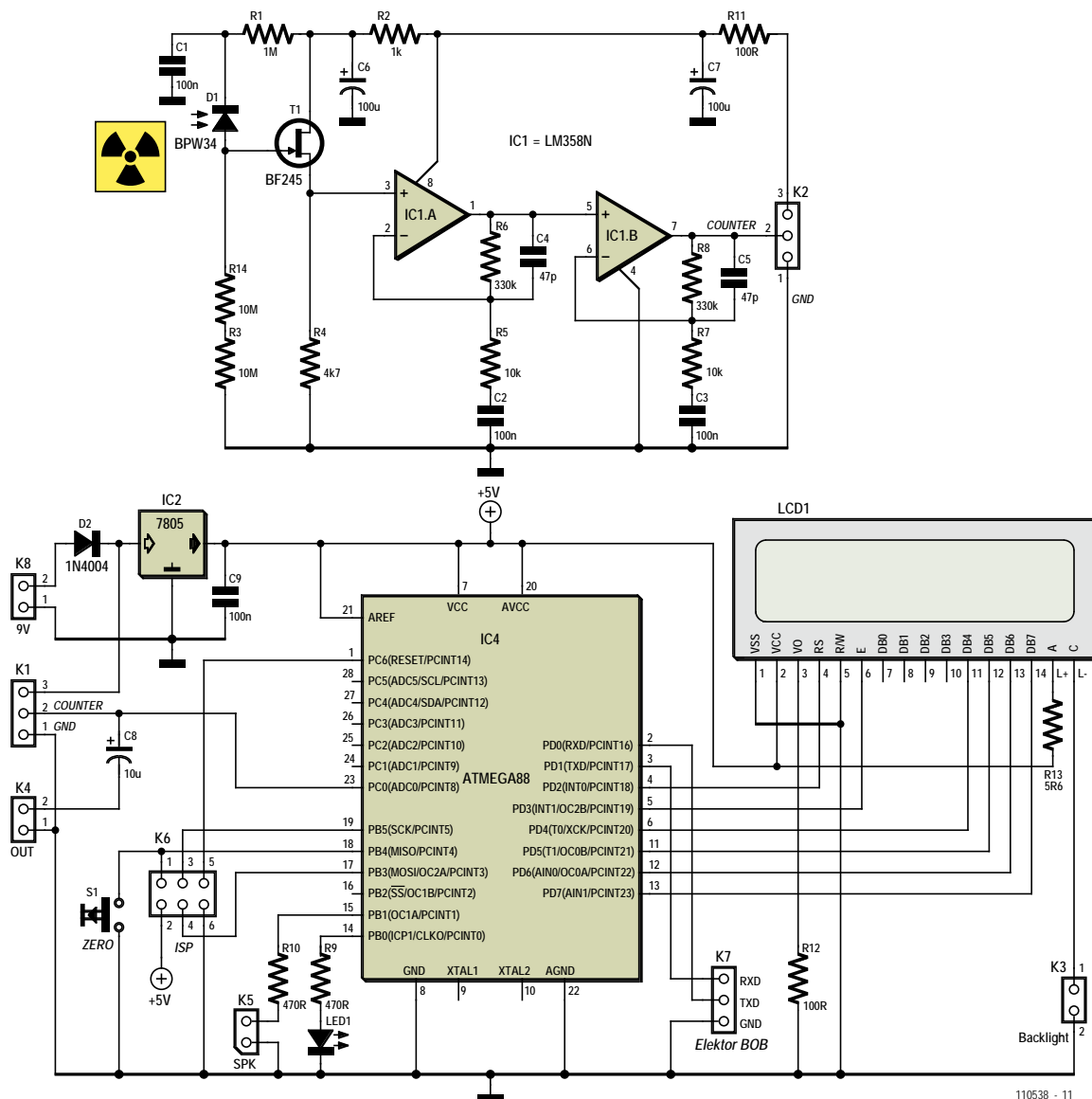


Figure 1. Circuit diagram of the preamplifier and microcontroller board.

## COMPONENT LIST

### Resistors

R1 = 1M $\Omega$   
 R2 = 1k $\Omega$   
 R3, R14 = 10M $\Omega$   
 R4 = 4.7k $\Omega$   
 R5, R7 = 10k $\Omega$   
 R6, R8 = 330k $\Omega$   
 R9, R10 = 470 $\Omega$   
 R11, R12 = 100 $\Omega$   
 R13 = 5.6 $\Omega$

### Capacitors

C1, C2, C3, C9 = 100nF  
 C4, C5 = 47pF  
 C6, C7 = 100 $\mu$ F 16V  
 C8 = 10 $\mu$ F 16V

### Semiconductors

D1 = BPW34  
 D2 = 1N4001  
 D3 = LED, 5mm, green  
 IC1 = ATmega88PA-PU (Atmel), programmed  
 IC2 = LM358N  
 IC3 = 78L05  
 T1 = BF245B

### Miscellaneous

S1 = pushbutton, 1 make contact  
 K1–K8 = pinheader, e.g. TE-Connectivity type 3-826926-6  
 LCD1 = DEM16217, Elektor Shop # 030451-72

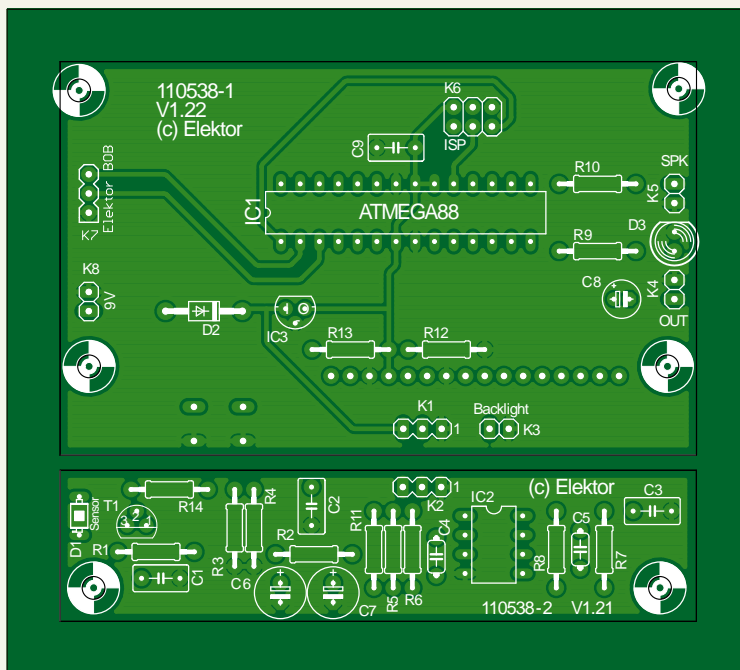


Figure 2. Printed circuit boards for the radiation meter.

regulator IC2, a 78L05, which generates the 5 V supply for the microcontroller and the sensor circuit. There is an ISP connector on the board to make software updates easy and there is a further connector to which a serial interface can be attached to allow transfer of data and settings to and from the PC (K7). The RXD and TXD signals on this connector are at TTL levels and compatible with the FT232R USB serial bridge breakout board ('BOB') published in the September 2011 issue of *Elektor*.

The sensor amplifier is attached at connector K1. Its output signal, which is superimposed on a quiescent DC level, is taken directly to the ADC0 analog input of the microcontroller. At start-up the microcontroller determines the quiescent DC

level and any incoming pulses that exceed this level by at least a preset amount are counted. Button S1 starts a new measurement without resetting the stored quiescent level: this lets you first set the quiescent level with the sensor not pointing at the sample of interest, and then start the measurement proper with the sensor in position.

Any pulse that is counted also results in a visible indication on the LED and on output K5, where a small loudspeaker can be connected for an audible indication. Any loudspeaker with an impedance of between 8  $\Omega$  and 32  $\Omega$  can be used, although it is a good idea to wire it in series with a volume control (a 1 k $\Omega$  potentiometer with a logarithmic taper) in case after a while the ticking

sound starts to get annoying.

The unprocessed signal is also taken to connector K4 via C8. This can be wired to a BNC socket for to allow, for example, connection of an oscilloscope. Alternatively, an audio amplifier can be connected, in which case particles of different energies will be distinguishable by their sounds.

### Printed circuit board

The printed circuit board for the project (Figure 2) comes in two sections. The sensor board can be separated from the main part and connected using a three-core cable. This allows the sensor to be mounted more easily in a light-tight enclosure.

The LCD and pushbutton are mounted on the reverse of the board; all other components are mounted on the top side. Sensor D1 can be mounted on whichever side of the board is more convenient, bearing in mind how the unit will subsequently be mounted in its enclosure.

The circuit is best tested first without the sensor diode fitted. The output of the opamp circuit should be a mid-level DC voltage, and the input should be so sensitive that even moving a finger near it will trigger the counter.

The sensor can now be mounted: Figure 3 shows it mounted on the reverse of the board. The sensitive area of the device is then under the lower right side of the board. It is important to ensure that the BPW34 photodiode is fully shielded from incident light and that the circuit area around the diode is carefully screened. During construction fit a small piece of black insulating tape under the photodiode: this will block light that might otherwise pass through the not-quite-opaque printed circuit board. Figure 4 shows the component side of the populated prototype board.

Aluminium foil can be used to screen the area around the photodiode, on both sides of the board. The foil should be connected to ground. This will give good shielding against both light and interference that could otherwise lead to false counts. Further pieces of insulating tape can be used under the aluminum foil to prevent shorts to the rest of the circuit. The foil should be fitted tightly over the photodiode, as otherwise it is possible accidentally to construct



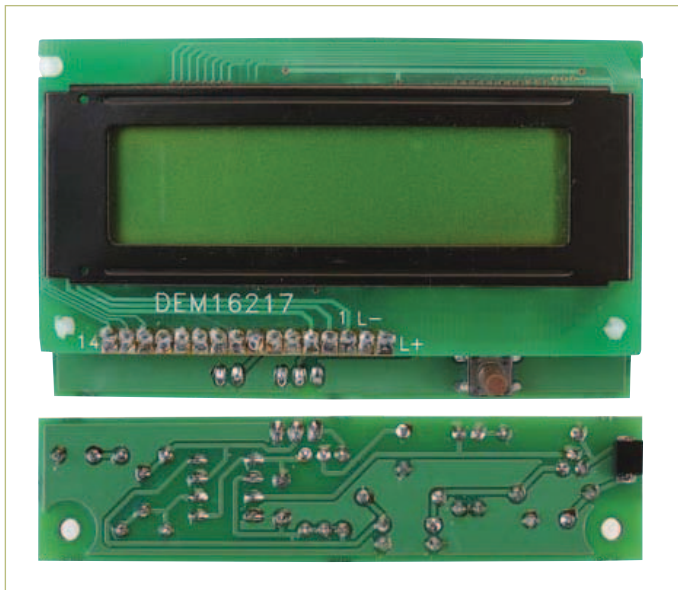


Figure 3. Components mounted on the reverse of the circuit boards.

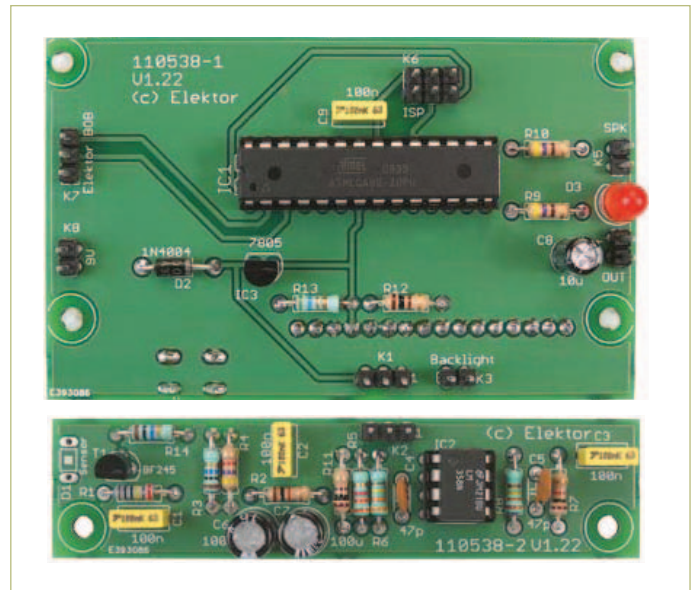


Figure 4. Component side of the boards.

a condenser microphone, with the counter responding to loud noises! The connection to the foil can be made, for example, using a bolt with two washers.

We can now test the operation of the board with the photodiode in its light-tight enclosure. A DC level of about 2 V to 3 V should appear at its output. This serves to check

active sample. Place a sample, for example of a suitable mineral, directly in front of the sensor. Any gamma radiation emitted by the sample will give rise to pulses clearly visible above the background noise level. Each pulse whose amplitude exceeds a certain level will be counted. The threshold level can be adjusted later in the software. If you

through. However, we can replace the BPW34 with a BPX61, with the glass window removed. The photodiode itself is now completely exposed and alpha particles can now be detected, resulting in signals with an amplitude some ten times greater than those produced by gamma rays. The same sensor amplifier can be used, but it must

## Measuring radiation using a dead standard photodiode

that there is indeed no light falling on the photodiode: if the operating point of the circuit is too high, this means that some light is probably getting in. When everything appears to be working satisfactorily, it should be possible to use an oscilloscope to observe background noise with an amplitude of around 5 mV<sub>pp</sub>. The unit can now be tested with a radio-

do not have a suitable sample to hand, all you need to do is wait: after a few minutes at the most a cosmic ray will impinge on the sensor and be counted (see **Figure 5**, **Figure 6** and **Figure 7**).

### Alpha particles

The BPW34 comes in a plastic package which is too thick to allow alpha particles

now be separated from the counter board and mounted in its own light-tight screened box. The sample to be examined must be placed inside the box next to the sensor, as even one layer of aluminum foil is thick enough to impede alpha particles.

A small abrasive disc mounted in a hobby drill can be used to remove the glass window from the BPX61. The job has to be done

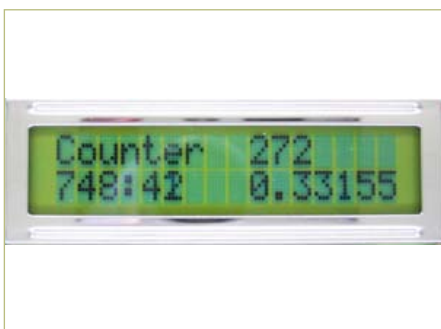


Figure. Measuring the background count rate: 0.33 pulses per minute is normal.



Figure 6. An individual pulse.

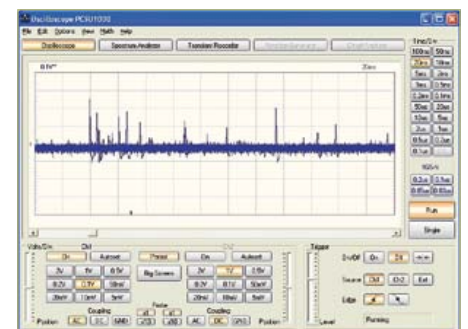


Figure 7. Background noise and the desired signal.

### Listing 1: Averaging readings and setting the trigger threshold

```

Readeeprom L , 1
If L = 255 Then L = 10
U = 0
For N = 1 To 1000
    D = Getadc(0)
    U = U + D
Next N
U = U / 1000
Um = U
U0 = Um + L
N = 0
M = M + 1
Count = N
Count = Count / M
Locate 2 , 10
Lcd Count
Lcd "      "
End If
Locate 1 , 10
Lcd N
Locate 2 , 1
Lcd M
Lcd ":"
Lcd S
Lcd "  "

```

### Listing 2: Detecting a pulse

```

Do
    Max = U0
    Do
        D = Getadc(0)
    Loop Until D > U0
    Portb.0 = 1
    Portb.1 = 1
    If D > Max Then Max = D
    Do
        D = Getadc(0)
        If D > Max Then Max = D
    Loop Until D < U0
    N = N + 1
    Max = Max - Um
    If Max > 255 Then Max =
255
    Print Chr(max);
    Portb.0 = 0
    Portb.1 = 0
Loop

```

### Listing 3: Timer processing and LCD output

```

Tim1_isr:
    Timer1 = -7812
    S = S + 1
    If S = 60 Then
        S = 0

```

### Listing 4: Processing the received energy values

```

Private Sub Timer1_Timer()
    While INBUFFER() > 0
        d = READBYTE()
        bin(d) = bin(d) + 1
    Wend
    For n = 1 To 255
        x1 = 2 * n
        x2 = 2 * n + 2
        y1 = 200 - bin(n)
        y2 = 200 - bin(n + 1)
        If y1 > 255 Then y1 = 255
        If y2 > 255 Then y2 = 255
        Picture1.Line (x1, y1)-
(x2, y2)
    Next n
End Sub

```

### Listing 5: Setting the trigger threshold

```

Private Sub Command2_Click()
    l = HScroll11.Value
    SENDBYTE l
End Sub

Private Sub Command4_Click()
    l = 100 + HScroll11.Value
    SENDBYTE l
End Sub

```

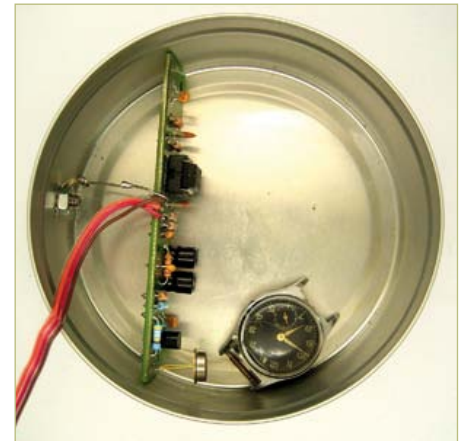


Figure 8. Screening.

extremely carefully as any damage to the diode itself or to its bonding wires spells doom for the device. A lidded tin makes a good screened enclosure (**Figure 8**). The metal must be connected to signal ground so that it provides electrical shielding as well as screening from ambient light. The lid must of course be fitted before any readings can be taken. The oscilloscope shows high-amplitude

peaks, up to 2 V, corresponding to alpha particles. Smaller peaks can also be seen: the BPX61 is, like the BPW34, sensitive to gamma rays too. Hence we can discriminate between the two types of radiation on the basis of the pulse amplitude.

### Firmware

The firmware is available for free download at [2]. It is written in BASCOM-AVR and is

reasonably easy to understand. The output voltage from the preamplifier in the quiescent state is about 2 V, on top of which we see the sensor pulses. In order to count these pulses we need to use some kind of comparator, and the ATmega is fast enough to do this job in software. At start-up the unit computes an average background signal level by averaging 1000 readings (see **Listing 1**). The averaged value U is then increased by a trigger threshold increment L to produce a threshold U<sub>0</sub> which gives adequate margin over background noise for reliable counting.

During the measurement itself (**Listing 2**) the output of the software comparator is also used to drive digital output ports PortB.0 and PortB.1. An LED is connected to pin B.0 to give a brief flash of light for every pulse detected. A small loudspeaker can be connected to B.1 with a resistor or a volume control potentiometer in series.

The maximum level achieved by each pulse is also measured and output as a single byte over the serial interface, to which a PC can be connected. This single-byte format is chosen so that no extra time is wasted in data transmission. As a consequence the maximum pulse height that can be reported is 1.25 V, which corresponds to a byte value of 255. Pulses of greater amplitude than this are clipped to the maximum value.

The display continuously shows the current counter value, updated once per second (**Listing 3**). Accompanying this on the lower



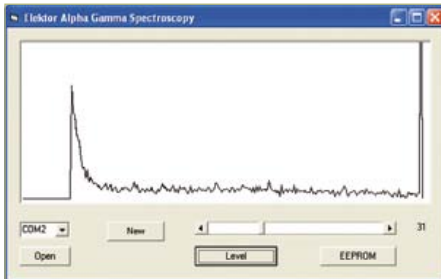


Figure 9. The alpha spectrum of a sample of pitchblende.

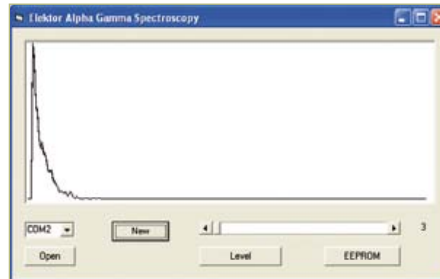


Figure 10. Gamma radiation: pitchblende behind a piece of aluminum foil.

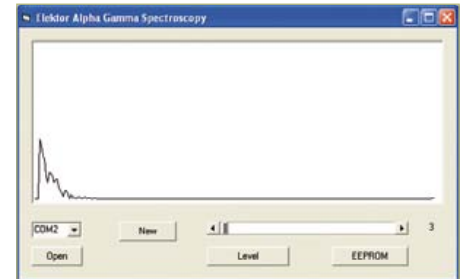


Figure 11. Beta particle measurements from a potassium chloride sample.

line of the display is the total time elapsed since the beginning of the measurement, shown in minutes and seconds. At the end of each full minute the program calculates the count rate in pulses per minute.

The sensitivity of the unit depends chiefly on the threshold setting for the comparator. This is initially set at ten A/D conversion steps above the quiescent level, but the value can be modified if desired using the

ure 9). The higher energy values are due to alpha particles: because of the relatively thick sample used these tend to lose a lot of their energy on their way to the sensor, and so no sharp energy lines are seen with this sample. When looking only at gamma rays using the BPW34 the higher energy levels are empty; when alpha particles are detected they usually have an energy beyond the range of the display, which

readings over a long period using the BPX61 and a small sample of potassium chloride. Ordinary potassium chloride contains a small percentage of radioactive potassium-40. Some 90 % of the disintegrations produce beta particles with energies of up to 1.3 MeV; the other 10 % produce gamma quanta with energies of 1.5 MeV. The beta spectrum shows a characteristic falling-off in energy up to a clearly-visible maximum

## Measuring background radiation

serial interface. A threshold of three steps has proved a good compromise, giving reliable pulse counting while still being above the background noise level. To change the threshold simply send a single byte to the unit over the serial port. Values of up to 100 are used immediately as the new threshold value. If you wish to store the new threshold as the default value in the microcontroller's EEPROM, add 100 to the value before sending it. In this case the new setting does not take immediate effect. So, for example, sending the byte value 103 will set the threshold after the next reset to three.

### PC software

The Visual Basic program AlphaGamma (available for free download at [2]) receives incoming bytes over the serial port and counts them in 255 bins. After a while collecting these values it becomes apparent which energy levels are particularly common in the incident radiation. The energy spectrum is plotted in a simple graph (Fig-

ure 9). The higher energy values are due to alpha particles: because of the relatively thick sample used these tend to lose a lot of their energy on their way to the sensor, and so no sharp energy lines are seen with this sample. When looking only at gamma rays using the BPW34 the higher energy levels are empty; when alpha particles are detected they usually have an energy beyond the range of the display, which

Listing 4 shows the main timer routine. In this routine all the bytes found in the input buffer are read in and processed. For each byte one element of the array 'bin' is incremented and then the results are displayed graphically.

The program also allows the trigger threshold to be set to any value from 2 to 100. The value can be used immediately or stored in EEPROM (Listing 5).

Alpha particles can easily be filtered out, for example by placing a piece of aluminum foil in front of the 'naked' BPX61. Doing this considerably reduces the energy spectrum obtained from a sample of pitchblende (Figure 10).

Beta particles also produce measurable signals. The pulse amplitude they produce is generally similar to that which results from gamma radiation, and it is therefore difficult to distinguish between the two. To test for beta particle sensitivity it is possible to take

value, while the gamma spectrum takes the form of a sharp line. The result is the full spectrum shown in Figure 11, demonstrating that the photodiode is capable of detecting alpha, beta and gamma radiation. (110538)

### Internet Links

[1] [www.elektor.com/110372](http://www.elektor.com/110372)

[2] [www.elektor.com/110538](http://www.elektor.com/110538)

## Kit of parts

A kit of parts is available for this project, containing the printed circuit board and all components including a pre-programmed microcontroller. A suitable display can also be purchased. Prices and further information can be obtained from the web pages accompanying this article at [2] and in the 'Shop' pages at the back of this issue.

# Simple Bat Detector

## Inexpensive, sensitive and easy to build

By Jan van Eck (The Netherlands) (j.vaneck@fontys.nl)

Various animal welfare associations everywhere in Europe have designated 2011 as the Year of the Bat. Their aim is to enhance public awareness of these flying mammals, which are unfamiliar and surrounded by mystery for many people. Bats hibernate at this time of year, but this gives you time to build this simple bat detector so that you will be prepared to go on safari for bat sounds in the spring.

In order to hear the sounds made by bats, we have to make use of a few electronic techniques. This is because bats use sounds in the ultrasonic range to detect objects while they are flying, and these sounds are well outside the range of human hearing. Many species of bats produce sounds at frequencies in the 40 kHz range, which happens to be the operating range of most standard ultrasonic transducers. If you amplify the signals picked up by one of these transducers and feed them into a frequency divider, the output will be within your normal range of hearing.

### The circuit

A type 400SR160 ultrasonic receiver was selected for the sensor, since most bats in our part of the world produce sounds in a range around 40 kHz. The author opted for the inexpensive plastic version of this transducer, which can easily be protected with a piece of aluminum foil or metallized adhesive tape.

The signal picked up by the transducer is amplified by a factor of approximately 200 by IC1, an LM386 (Figure 1). Although the LM386 is designed as a power amplifier IC, its low price makes it perfectly suitable for use as a 'normal' gain stage. The main advantage here is that aside from a few decoupling components, no external components are necessary. The gain can be boosted from the minimum value of 26 dB to 46 dB by connecting a capacitor across the internal feedback path of the LM386 (between pins 1 and 8). A 10- $\mu$ F electro-

lytic capacitor is used for this purpose in the standard application circuit, but this is only necessary if you want to amplify virtually the entire audio band with the same gain. That is not necessary for this application, so the value of capacitor C1 can be

reduced to 220 nF, which places the corner frequency at approximately 4 kHz. This means that in theory the capacitance could be made even smaller, which would reduce the risk of problems with low-fre-

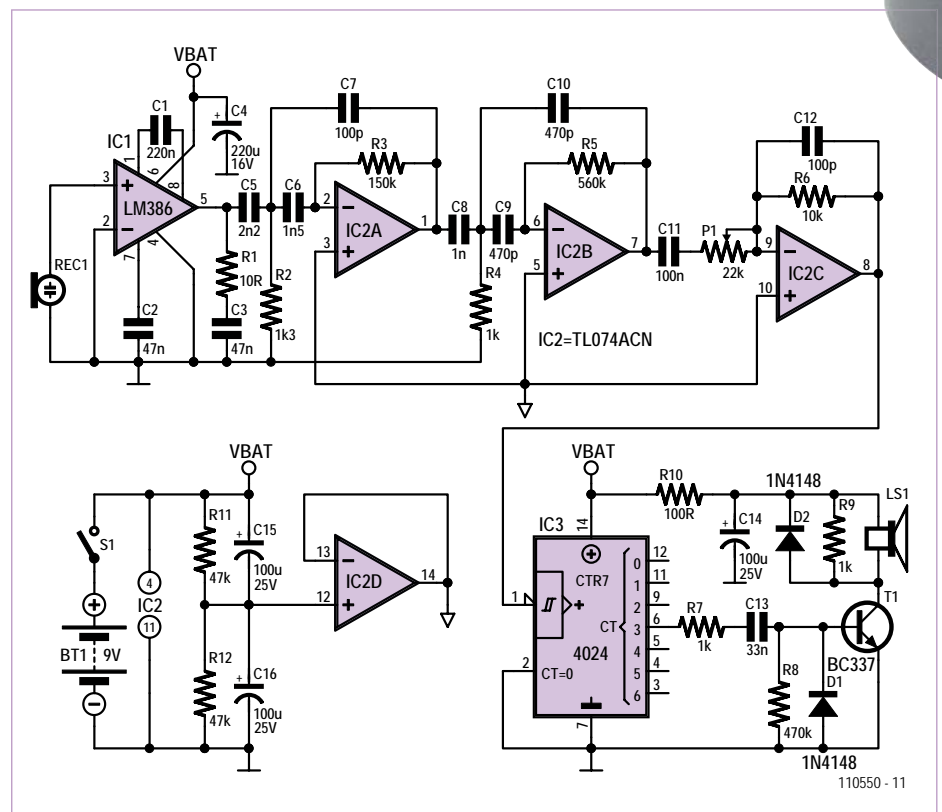


Figure 1. Schematic diagram of the simple bat detector.



quency noise.

The output signal from the amplifier passes through a fairly steep high-pass filter built around IC2a and IC2b. This is a Chebychev filter with pass bandwidth of approximately 15 kHz and a gain of around 50 (see the blue curve in **Figure 2**). It strongly suppresses undesirable signals, such as (mechanically coupled) feedback from the loudspeaker. The filter was dimensioned using the free program FilterPro Desktop from Texas Instruments [1]. The gain of the filter stage is approximately 35 dB, as can easily be seen from Figure 2 (0 dB, is relative to the output), where the bottom curve represents the output signal from IC1.

The bandwidth is further limited by adjustable-gain amplifier IC2c after the filter to suppress high-frequency noise. Capacitor C12 limits the bandwidth to 160 kHz. If

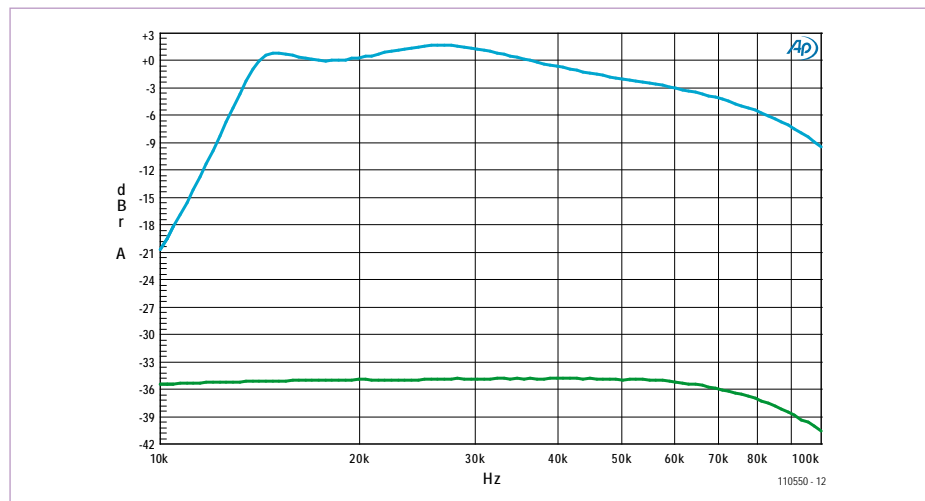


Figure 2. Frequency response at the output of IC2b (top) and the output of IC1 (bottom).

necessary, you can reduce the bandwidth a bit more.

The final opamp in the circuit (TL074) is used in conjunction with a voltage divider

(R11/R12, each 47 k $\Omega$ ) to generate an artificial ground reference for the other opamps at half the supply voltage level.

The output signal from IC2c is fed to the

## COMPONENT LIST

### Resistors

R1 = 10 $\Omega$   
R2 = 1.3k $\Omega$   
R3 = 150k $\Omega$   
R4,R7,R9 = 1k $\Omega$   
R5 = 560k $\Omega$   
R6 = 10k $\Omega$   
R8 = 470k $\Omega$   
R10 = 100 $\Omega$   
R11,R12 = 47k $\Omega$   
P1 = 22k $\Omega$  preset, vertical mounting

### Capacitors

C1 = 220nF MKT, lead pitch 5mm  
C2,C3 = 47nF MKT, lead pitch 5mm  
C4 = 220 $\mu$ F 16V radial, lead pitch 2.5mm

C5 = 2.2nF MKT, lead pitch 5mm  
C6 = 1.5nF MKT, lead pitch 5mm  
C7,C12 = 100pF ceramic, lead pitch 5mm  
C8 = 1nF MKT, lead pitch 5mm  
C9,C10 = 470pF ceramic, lead pitch 5mm  
C11 = 100nF MKT, lead pitch 5mm  
C13 = 33nF MKT, lead pitch 5mm  
C14,C15,C16 = 100 $\mu$ F 25V radial, lead pitch 2.5mm

### Semiconductors

D1,D2 = 1N4148  
T1 = BC337-40  
IC1 = LM386N-3  
IC2 = TL074CN  
IC3 = 4024

### Miscellaneous

REC1 = 40kHz ultrasonic sensor (receiver) (e.g. Prowave 400SR16P, diam. 16 mm)  
LS1,S1,BT1 = 2-pin pinheader, lead pitch 0.1 inch (2.54mm)  
3 pcs 2-pin connector (for loudspeakers, switch and battery)  
LS1 = loudspeaker 8 $\Omega$ /0.3W, diam. 20mm (e.g. Kingstate KDMG20008)  
S1 = slide switch, 1 make contact  
BT1 = 9V battery with clip-on lead  
PCB # 110550-1 (see [www.elektor.com/110550](http://www.elektor.com/110550))

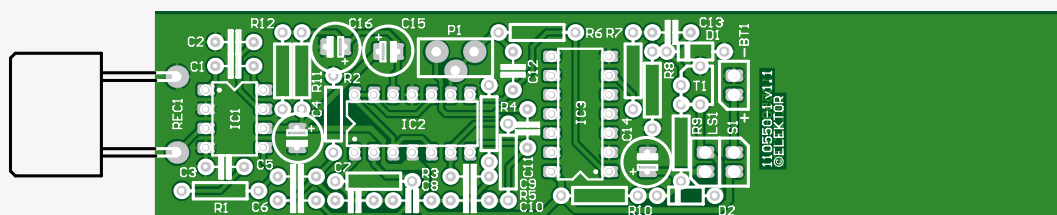
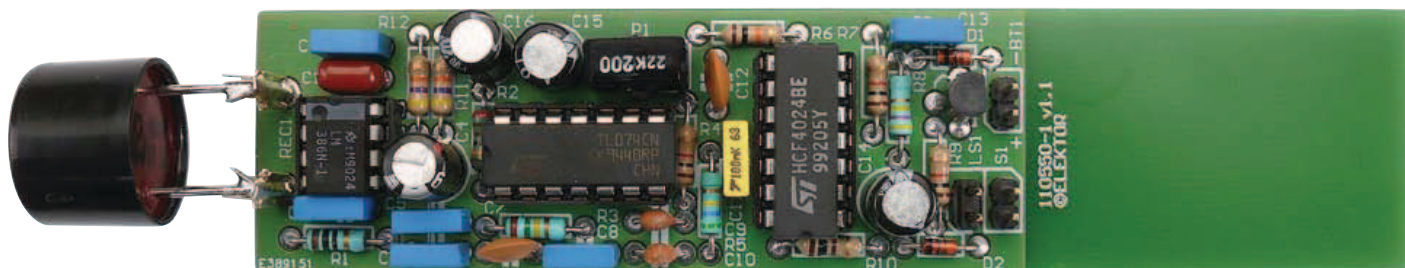


Figure 3. The PCB for the circuit is long and thin to enable it to fit in a length of PVC pipe.



clock input of a type 4024 seven-stage binary counter. The signal on the CT3 output (pin 16) is the clock input divided by 16. It consists of pulses in the audible range (2 to 3 kHz), which are fed via T1 to the loudspeaker to generate an acoustic output. Diode D1 is included for protection against negative voltages on base of the transistor. Resistor R7 limits the drive current that IC3 supplies to T1. A free-wheeling diode is connected in parallel with the loudspeaker (LS1) to protect transistor T1 from voltage spikes generated by the inductance of the speaker coil. You can also use a small piezoelectric buzzer for LS1. Resistor R9 is included to ensure proper operation of a piezoelectric buzzer, as otherwise the output would be mute because piezoelectric buzzers are highly capacitive. Decoupling network (R10/C14) in the speaker supply line suppresses noise on the supply voltage rail, which is necessary because the input stage is very sensitive.

The quiescent current consumption of the entire circuit is around 14 mA, rising to a maximum of 90 mA when it receives ultrasonic signals. The circuit works well with a supply voltage as low as approximately 4.5 V.

### Mini PCB

The PCB layout designed for the bat detector is shown in **Figure 3**. The 9 V battery can be fitted on the bottom of the board. The PCB has a ground plane on the top side to provide screening, which is necessary due to the relatively high input sensitivity. The board is dimensioned to fit in a length of standard PVC pipe with an inside diameter of 33 to 35 mm, which can be purchased in any home improvement shop. Matching end caps are also available, and a small loud-

speaker and switch can be fitted on an end cap as illustrated in **Figure 4**. If you fit three self-adhesive rubber feet on the bottom, you can stand the pipe upright on a garden table, and whenever any bats fly overhead you will hear them right away.

The sensitivity can be adjusted with the trimpot. It should be set to minimize or eliminate any ticking sounds from the speaker. When making the adjustment, take care to avoid undesirable sources of ultrasonic energy in the immediate vicinity, such as fluorescent lamps, television sets, computer monitors, switch-mode power supplies and so on. They can be detected at distances of 10 to 20 feet (3 to 6 meters).

Simply rubbing your thumb and forefinger together, zipping a plastic bag (such as a sandwich bag) or shaking a bunch of keys will generate enough ultrasonic energy to be detected by the circuit at a distance of several meters.

Bats are rather loud and can be detected at distances of 100 feet (30 meters) or more with this circuit. The various sounds produced by bats can also be clearly recognized with this detector.

(110550)

### Internet Link

[1] [www.ti.com/tool/filterpro](http://www.ti.com/tool/filterpro)



Figure 4. The on/off switch and a speaker are fitted in the end cap.





**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**



## STC Microcontroller

The World's Largest 8051 MCU Family



- 1000+ parts
- Free ISP software
- Keil IDE supporting
- Instant order online
- World wide free shipping

- Enhanced 80C51 CPU, 1T per machine cycle
- Binary level compatible with conventional 8051
- 0.5-62K on-chip flash ROM
- Up to 2048KB SRAM
- In-System program or In-Application program
- Internal oscillator, reset, WDT
- On-chip ADC, PWM, EEPROM, SPI
- Addressing up to 64KB of external RAM
- Dual data pointer
- 6 vector address, 2 level priority interrupt
- Up to 3 UART with baud-rate generator
- Low power consumption
- 8 to 48Pins, DIP/PLCC/QFN/SOP/DIP/SSOP
- Ultra safe code protection for flash ROM
- Highly EDS protecting

### Feature Product

### Free USB ISP Tool



**STC11F04E**  
Unit Price:  
**\$0.86@1ku**

Compatible with AT89C4051  
6-8 times faster in average  
2K EEPROM  
16, 18, 20pins DIP/SOP/LSSOP

Reduce the cost dramatically while  
considerably improving the performance

Tel: +8610 8256 2708 Email: sales@stc-51.com  
Website: <http://www.stc-51.com>



# Meet Eddie.

## Expandable Development Discs for Innovation and Experimentation

Eddie is a new type of robot from Parallax Inc. designed to foster creativity, innovation and experimentation. Compatible with Microsoft's Robotics Developer Studio, Eddie can roam autonomously, see in 3D using the Microsoft Kinect™, and be driven remotely using a wireless controller.

The included control board uses the 8-core Propeller microcontroller to directly control two 12 V motors and collect data from several sensors around the robot.

- **Eddie - Assembled (#28990; \$1199.00)**
- **Eddie - Unassembled (#28992; \$999.00)**

Note: Laptop, Microsoft Kinect, and wireless remote not included.

**PARALLAX**

[www.parallax.com](http://www.parallax.com)

"ParallaxInc" on Twitter, Facebook, and YouTube



Learn more about **Eddie** at [www.parallax.com/Eddie](http://www.parallax.com/Eddie) or call our Sales Department toll-free 888-512-1024 (Mon-Fri, 7AM-5PM, PT).

*Friendly microcontrollers, legendary resources.™*

Prices subject to change without notice.

Propeller, Parallax, and the Parallax logo are trademarks of Parallax Inc.

# Audio Guide

## First steps with Platino

Audio guides can be borrowed or rented in many museums: devices containing spoken information about the objects in the exhibition. In this article we describe such an audio guide, but this is a very special one: it detects the object using RFID. The wearer only needs to come in the vicinity of the object and will then automatically hear the correct details and explanation regarding the work (of art).



By Clemens Valens and Grégory Ester (France)

Here, we use two ready-made boards: a Platino [3], which forms the brains of this circuit, and an rMP3-module [2], which takes care of the sound. With the latter you can playback audio files that you have recorded beforehand on a memory card. An rMP3 supports SD-, SDHC- and MMC-cards from 8 MB to 32 GB. You can see the design of the whole unit in the block diagram of **Figure 1**. The Platino and the rMP3 communicate with each other via the serial TTL-connection. The method has some similarities to when sounds are produced from a mobile phone or a PC: when a certain event occurs we play back a certain sound (file). The event, in this case, is when the antenna [6] of the RFID-reader [4] inside the Audio Guide comes in the vicinity of a certain object — a painting, or whatever. Inside or nearby that object is a transponder [5], also known as RFID tag. This tag contains a unique identification code which is picked

up wirelessly by the RFID reader. Based on this code the software retrieves the corresponding audio file from the database and automatically plays it back.

### Reading MP3s

The rMP3-module is a so-called *Arduino-shield*, a board that you select as a 'shield' in the headers of an Arduino and is therefore also completely compatible with it. The rMP3 has a holder for an SD-card. You can play back MP3s stored on this card, but a few common WAV-formats are also supported. The module obeys commands which are sent to it via an asynchronous serial connection. With these commands you can read MP3-files, queuing the next file, fast forward, pause playback, adjust the volume, etcetera. You can of course also use the memory card as a storage unit. The possibilities are endless! We also (heartily) recommend that you read the online

documentation provided by the manufacturer [7]. The rMP3 is fitted with a 3.5-mm socket for headphones or a set of speakers of at least 16  $\Omega$ .

Get cracking! While the rMP3 has been designed to be plugged straight into an Arduino (or the Platino, which is compatible with that), you nevertheless have to make a small modification before you do that. That's because we are driving the sound module from USART0, one of the two serial ports on board of the ATmega1284P on the Platino. That's why RXD0 and TXD0 have to be rerouted to TX and RX respectively of the rMP3. In **Figure 2** you can see the two pins which have been cut off; the orange wire connects leg 0/R (RXD0) to pin 6 (TX on the rMP3); the yellow wire connects 1/T (TXD0) to pin 7 (Rx-MP3).

If necessary, begin by formatting the SD-card as FAT32. In the root directory of the





SD-card you have to make a directory *audioguide*. Choose a nice MP3 from your collection, store it in this directory and rename it 'music.mp3'. Now plug the SD card into the appropriate socket on the rMP3-module and

connect a 4 x 20 LCD to K9 of the Platino. You can now fit the rMP3 in place on the Platino.

It is now time to load the software. The hex code to be loaded is called 110544-l\_audioguide\_test\_rmp3\_lcd.hex and can be downloaded from [1]. The test consists of reading the music file, and on the display appears something resembling that shown in **Figure 3**. On the top line will be the heading 'rMP3 and

Platino', with the version of the firmware below that, followed by a serial number. The third line shows that the left and right volume are equal to 16, which corresponds to  $-0.5 \times 16 = -8$  dB. On the bottom line, from left to right, we can see the number of seconds the music has been playing (here 2), the sample frequency is 44.1 kHz and the bit rate is 128 kbits/s. The parameter 'J' indicates that the MP3-file is encoded in a mixed (stereo/mono) format. At this stage we know for sure that the rMP3 module is working as it should.

### Identification with RFID-i

We have already discussed the RFID reader for this project in an earlier issue of Elektor [8]. It comprises a motherboard with an EM4095 chip and an antenna. The EM4095 is designed for frequencies ranging from 100 kHz to 150 kHz. This is the reason we selected a family of passive RFID transponders [5] that operate at 125 kHz. The maximum reading range is about 12 cm when using an antenna with a diameter of 53 mm. If we use an antenna of 25 mm then the reading range becomes 6 cm. Connecting the RFID module is a piece of cake: the card has five lines called +5 V, SHD, DEMOD OUT, MOD and GND; these we con-

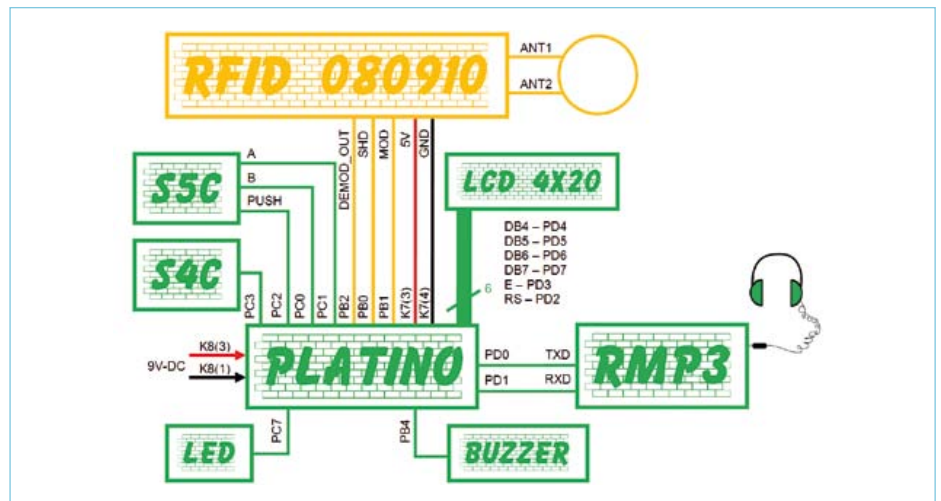


Figure 1. Block diagram of the Audio Guide.

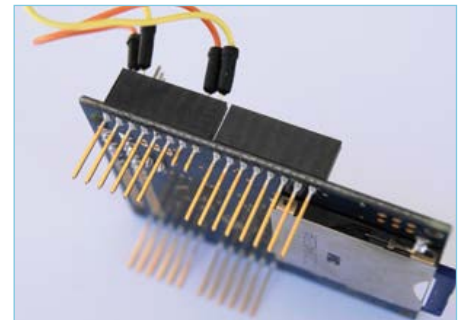
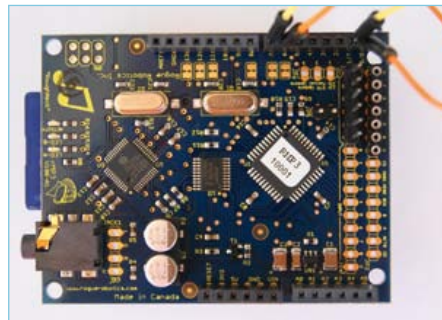


Figure 2. Platino and rMP3: a successful partnership.

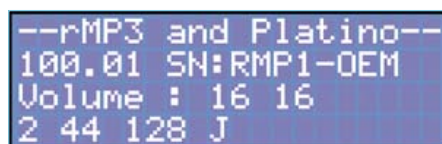


Figure 3. Fortunately there is still something to do to the rMP3.

nect to +5 V, 8, 10, 9 and GND respectively of the female connectors on the rMP3, which we've just plugged into the Platino.

### Preparing the Platino

The Audio Guide is provided with two operating devices: a pushbutton and a rotary button, or more accurately, a *rotary encoder*. These are both soldered onto the copper side of the Platino circuit board. The pushbutton is connected to S4C and the encoder to S5C.

This requires that you make a few links on the Platino as is shown in **Figures 4 and 5**. For an overview we again refer you to the

block diagram of **Figure 1**. The links are solder bridges: no more than a drop of solder across both pads. At JP7, S4 (S4C in the block diagram) has to be connected to PC3; signal B of the encoder (S5C) goes to PC0 via S1 of JP4; signal A from S5C arrives at PC1 of the microcontroller via S2 of JP5. In addition to turning, the user can also push the rotary button, to confirm a selection, for example. Translated into hardware this is a drop of solder on JP6 across S3 and PC2, which allows the state of the pushbutton to be read by PC2 of the microcontroller. We've used a green LED as an indicator, connected via JP14 to PC7 of the controller. You can limit the current through the LED with a resistor of 470  $\Omega$  in the location of R8.

The backlight for the LCD is not used. We do, however, need a piezo buzzer. That's BUZ1 on the Platino, connected to PB4 via JP1, again with a solder bridge.

The controller that we use here is an ATmega1284P, in a 40-pin DIP package. It can be programmed in-circuit (ISP, for *In System Programming*) via a synchronous

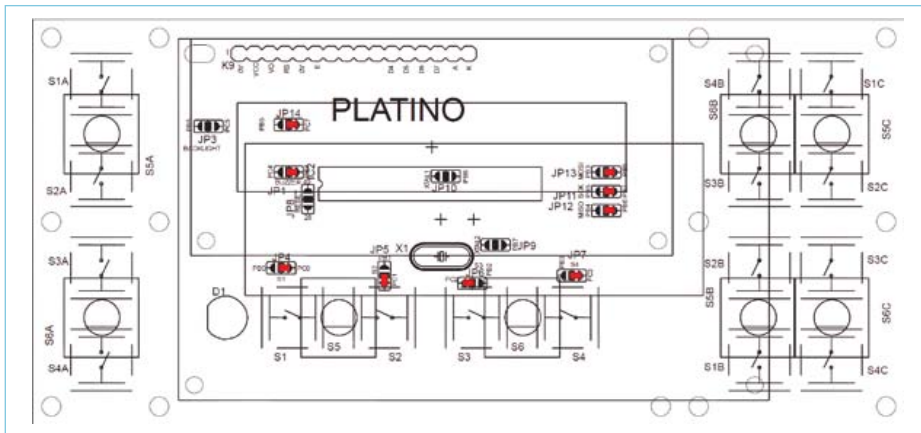


Figure 4. Solder bridges on the copper side of the Platino..

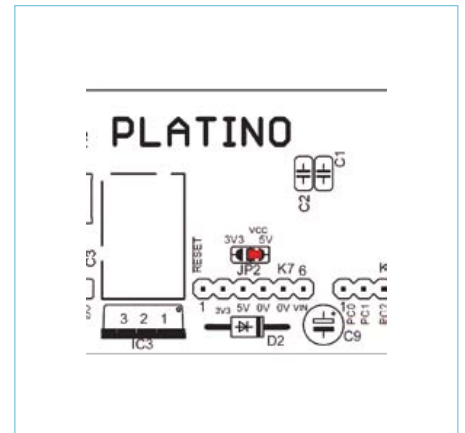


Figure 5. ...and one more on the other side.

serial connection. An external programmer [9] is connected via K3, a 6-pin HE10 connector. The lines MOSI, MISO and SCK that arrive at that connector have to be routed to PB5, PB6 and PB7 respectively via jumpers JP13, JP12 and JP11.

The power supply voltage for the entire assembly is connected with the ground to K9-1 and the positive to K8-3. The rMP3 gets +5 volts, which is configured with a solder bridge at JP14 for the positive.

### Everything working together

Every individual object that needs to be recognized has to be fitted with a unique transponder. When the Audio Guide comes in the vicinity the transponder, the corresponding audio file will be played back through the headphones or speakers. The

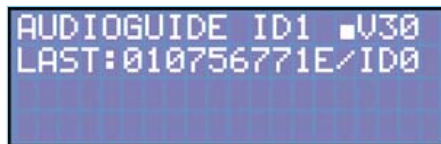


Figure 6. "ID=0" means that this transponder has not yet been registered.

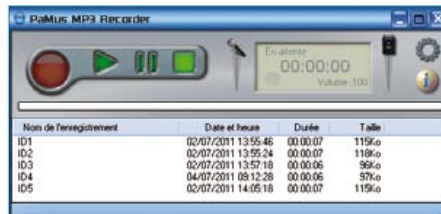


Figure 7. A handy little aid: PaMus MP3 Recorder.

green LED lights up and the buzzer sounds the moment the RFID-card has been recognized. With the program '110544-I\_audioguide\_firmware\_v1.2.bas' belongs an ASCII-file called 'tags.txt'. This contains the identification number, the title and the creator of the object as it will appear on the LCD, see Listing 1. These codes are included during compilation thanks to the instruction \$include tags.txt. The corresponding MP3 sound files on the SD-card have to be named IDx.mp3, where x is the index number (the number in brackets) in tags.txt. For example, the file ID4.mp3 belongs with the painting 'The first steps' by Vincent van Gogh. Note that the file names are not case sensitive.

The sound volume is adjustable from 0 to 100 in steps of 5 with the use of the rotary button. The rotary button does not only

### Listing 1.

The file tags.txt contains all the information that has to appear on the LCD. The titles (Title(x)) and the name of the creators (Artist(x)) are not allowed to be longer than twenty characters.

```
const maxi = 6
Dim Rfid(maxi) As String * 10
Dim Title(maxi) As String * 20
Dim Artist(maxi) As String * 20

Rfid(1) = «0107567708»
Title(1) = «VOLUME SETTING»
Artist(1) = «MARCUS MILLER/POWER»
Rfid(2) = «0107567790»
```

```
Title(2) = «LA GIOCONDA»
Artist(2) = «LEONARDO DA VINCI»
Rfid(3) = «01075677B4»
Title(3) = «STARRY NIGHT»
Artist(3) = «VINCENT VAN GOGH»
Rfid(4) = «01075677AE»
Title(4) = «FIRST STEPS»
Artist(4) = «VINCENT VAN GOGH»
Rfid(5) = «010756785E»
Title(5) = «FRONT COVER»
Artist(5) = «ELEKTOR 1978»
Rfid(6) = «01075676E6»
Title(6) = «THE LAST SUPPER»
Artist(6) = «LEONARD DE VINCI»
```



operate in the two rotary directions, clockwise (CW) and counter clockwise (CCW), but you can also push it, which is equivalent to the closing of S4C. This is used to stop the playback. Pushing it again will start the playback. The file that will play, is the one that corresponds to the ID number that is shown on the first line at that time. As soon as the playback starts, the title and creator appear on the display. In this way you can also listen to the narration without having to be in the vicinity of the corresponding object, or you can repeat certain parts of the text.

The transponder that is stored in the first position (under RFID(1)) starts the file ID1.mp3. You can use this one to adjust the sound volume with the aid of the rotary button. You can use any music file of your liking.

We assume that by now you have already loaded the SD-card with MP3s with ID 1 though 6. When you now load the firmware (110544-I\_audioguide\_firmware\_v1.2.hex) for the first time then the Audio Guide only operates under manual control. Even though the system can pick up the transponder codes, it does not know which index number belongs with which transponder — from the software's perspective all transponders still have ID0.

Now it is very handy that, as a preparation step, you give each transponder a sequence number and arrange them side by side and in order. Now open the file tags.txt in a text editor, for example Notepad. Now hold the RFID antenna near the first transponder and copy the hex ID on the display into tags.txt. Repeat this for all the transponders. Save tags.txt and then compile and download the firmware again. The Audio Guide is now ready for use, which you can check by passing the antenna over each transponder in turn. The firmware has space for up to 96 transponders, in the code, the constant maxi has to indicate the correct number. You can see the const maxi = 6 for the six tags in our example.

### MP3 Recorder

The author uses the program PaMus MP3 recorder [10] to record spoken text directly from a microphone into an MP3-file. This is very easy to use, just like a tape recorder (**Figure 7**). You can choose from three levels of sound quality, files are automatically generated and you can rename or erase them from within the program. If you rather not work with French-language software then you can choose something else yourself from the countless other options available on the internet. Search for 'mp3 recorder open source' for free software.

(110544)

### Internet Links

- [1] [www.elektor.com/110544](http://www.elektor.com/110544)
- [2] [www.lextronic.fr/P18805-module-de-restitution-rmp3-pour-arduino.html](http://www.lextronic.fr/P18805-module-de-restitution-rmp3-pour-arduino.html)
- [3] [www.elektor.com/100892](http://www.elektor.com/100892)
- [4] [www.elektor.com/products/kits-modules/modules/080910-91-rfid-savvy.925192.lynkx](http://www.elektor.com/products/kits-modules/modules/080910-91-rfid-savvy.925192.lynkx)
- [5] [www.lextronic.fr/P1503-carte-transpondeur-rfid-card1.html](http://www.lextronic.fr/P1503-carte-transpondeur-rfid-card1.html)
- [6] [www.lextronic.fr/P1478-antenne-rfid-125-khz-ant-rfid2.html](http://www.lextronic.fr/P1478-antenne-rfid-125-khz-ant-rfid2.html)
- [7] [www.roguerobotics.com/wikidocs/rmp3/start](http://www.roguerobotics.com/wikidocs/rmp3/start)
- [8] [www.elektor.com/080910](http://www.elektor.com/080910)
- [9] [www.elektor.com/080083](http://www.elektor.com/080083)
- [10] <http://papiermusique.fr/dossier03.php>



Back-to-school discounts  
on basic kits at [paia.com](http://paia.com)

## Electronic Kits for Enthusiasts

# BUILD YOUR OWN!

### PAiA 9700s Modular Synthesizer



### FatMan Analog MIDI Synth



### Theremax The Ultimate Theremin



Check out these  
and other exciting  
D.I.Y. projects!



[paia.com](http://paia.com)

Music/Tutorials/Kits  
Guitar Effects  
Tube Electronics  
Studio Gear



# Audio DSP Course (5)

## DSP program structure

In this article we describe the basic structure used for the DSP application programs developed in this course. We have decided to divide application programs into two parts: the first, the same for all applications, provides the high-level structure, while the second, unique to each application, carries out the actual processing. This division makes it easier for beginners to get to grips with the applications we describe and get started with DSP programming. The article concludes with some handy hints and tips on using the assembler and on programming the DSP.

By Alexander Potchinkov (Germany)

The first of the two parts of the software we shall call the 'framework', and is the same for all applications. The second part, the 'audio loop', is concerned with the audio signal processing proper, and has to be developed anew for each project within this course and for any application you might design. Both these terms were introduced in the second article in this series. We will now look at how they are applied in the context of the DSP board and of the applications developed in this course.

### Framework tasks

The first job of the framework code is to initialize the DSP itself. This involves configuring the processor clock frequency and the interrupt system, setting up a software stack and various other tasks. The next job is to establish connections to the peripheral devices on the DSP board so that audio and control data can be passed to and from. This requires writing to various registers in the DSP to configure the processor's interface hardware to suit the peripherals. In effect the task of the framework is to initialize all the hardware on the board and put it into a state where it can be used without further ado. In our case we have to take account of the needs of five peripheral components: ADC, DAC, SRC, SEEPROM and SPI port.

**Figure 1** illustrates this array of peripherals connected to the central DSP. The solid lines show paths where data transfer takes place and the dotted lines show control paths. The control paths are used by the DSP to configure the peripherals and to request status information from them.

- A clock signal is made available to the **ADC**, from which the sample rate and data transfer timings are derived. The ADC is capable of determining the ratio between the master clock frequency and sample rate by itself and this value therefore does not need to be configured. In our case the ratio is 512.
- The **DAC** is also provided with a clock signal, and, as with the ADC, it is able to determine the master clock to sample rate ratio autonomously and configure itself accordingly.
- The **SRC** is not capable of such independent operation as the ADC or DAC, and requires some configuration. We gave the byte sequence necessary to configure the I<sup>2</sup>S and digital audio interfaces in the SRC in the third article in this series. The DSP configures the SRC over its SPI port as part of the framework code, and in order to do this the DSP must first configure its SPI port to be compatible with the communications

protocol used by the SRC. Then the DSP can reset the SRC and send the necessary 54-byte configuration sequence.

The **SEEPROM** can be used for general-purpose non-volatile storage. If the DSP board is to be used in a stand-alone application (that is, without the use of the debugger) with fixed application code then it is possible to arrange matters so that the DSP boots from the SEEPROM, taking advantage of the bootloader code stored in the DSP's on-board ROM.

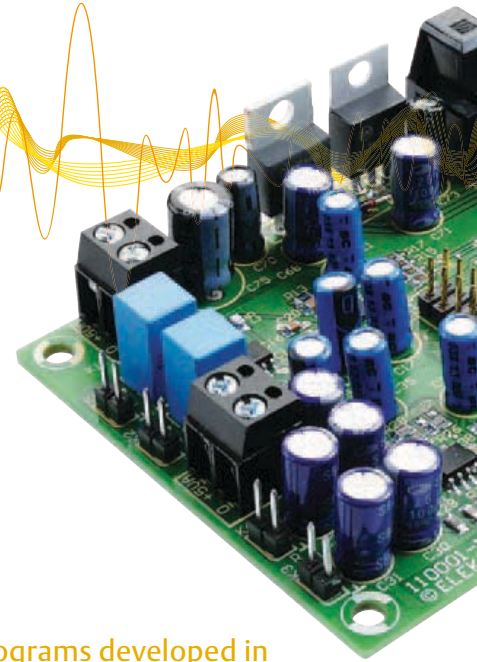
The **SPI port** can be used to transfer data to and from devices external to the DSP board. For the digital signal level meter application in this course we use the SPI port to drive an LED bargraph display. We could also use the SPI port to connect to an external microcontroller with keypad, digital potentiometer (rotary encoder) and LCD to allow the parameters of a program running on the DSP to be adjusted interactively, for example altering the frequency of the sine wave output in our signal generator application.

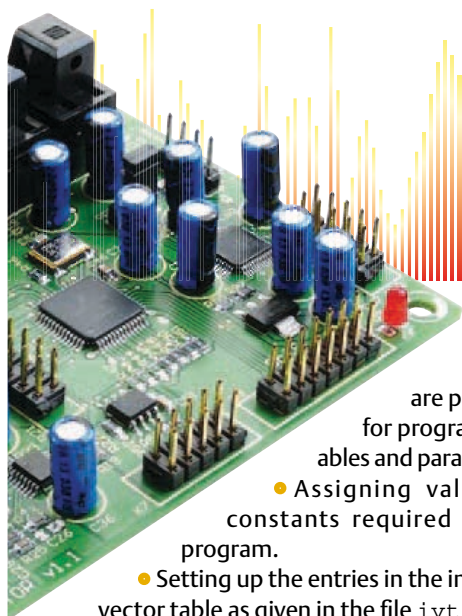
### Structure of the framework code

The framework can be divided into four parts.

#### 1. General declarations

- Defining the memory map: fixed areas





are provided for program variables and parameters.

- Assigning values to constants required by the program.
- Setting up the entries in the interrupt vector table as given in the file `ivt.asm`.

## 2. DSP setup

Setting up the DSP itself, which includes configuring the processor core and the peripheral interfaces, is done by programming the 24-bit registers that lie at the top of the processor's address space from \$FFFF80 to \$FFFFFF.

This range covers a total of 128 locations of which we use up to 18. The processor manufacturer refers to this address range as 'internal I/O memory'. In the 52-pin DSP56374 device that we use only the X-RAM locations are used. The locations are written to using the special instruction `movep` (move peripheral data) which allows values to be written directly to the register without the overhead incurred by the normal `move` instruction of transferring the value via a CPU register. For example, the instruction `movep #D17D00,x:RCR` writes the specified value directly to RCR (Receive Control Register), which controls the characteristics of the receiver part of the audio interface.

Here RCR is a handy and easy-to-remember abbreviation (much like the mnemonics used for assembler instructions) for the address \$FFFFBF. To make the job of the programmer easier all our programs make use of the auxiliary file `mioequ.asm`, which contains (among other things) similar abbreviations for all the I/O register addresses. A suitable 'include' directive to the assembler makes the abbreviations in the file available to our program.

- The program starts at address \$000000, which is the value of the DSP's program counter immediately after reset. The first instruction is a branch to program address \$000100, thus skipping over the interrupt vector table and other paraphernalia associated with the DSP core. Then the program proper begins.

- All interrupts are disabled while the processor configuration is changed. This avoids the possibility of the processor crashing or other unexpected operations if an interrupt should be generated before the core and peripheral registers are properly initialized.

- The hardware stack pointer (`sp`) is reset and the software stack is set up in ascending locations in X-RAM starting at address \$40.

Setting up the DSP and its peripheral interfaces is a tedious task, which is a reflection of the flexibility of the device and its range

using the PLL multiplier and divider values it is important to bear in mind that the VCO only operates reliably between 300 MHz and 600 MHz; we operate the VCO at a frequency of 589.824 MHz. The manual provides tables that give advice on suitable settings for common audio sample rates. If non-standard frequencies are to be used the calculations must be done manually. Based on the fact that the board is populated with a 24.576 MHz crystal oscillator module the correct PLL configuration is obtained by setting `PCTL=#01E006` which results in an overall clock multiplication factor of 6.

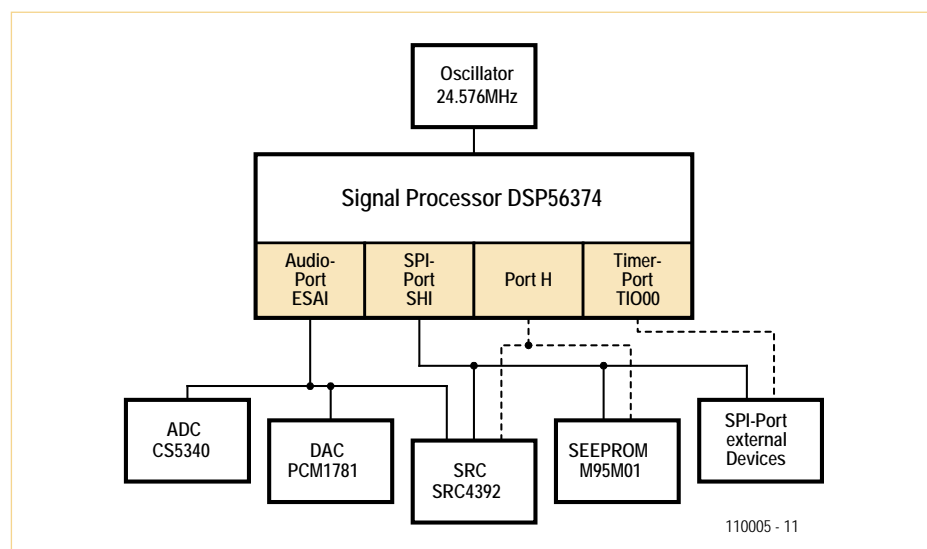


Figure 1. An array of peripherals connected to the DSP.

of possible applications. There is no denying that setting up the processor properly demands extensive study of the manuals and can be a significant hurdle to the beginner. Fortunately, however, the processor is logically designed and Freescale's manuals are well written. A complete step-by-step description of what is required would far exceed the space we have available, and so we discuss below only the most important aspects of the set-up process.

### • Processor clock PLL configuration.

The processor clock frequency is set to 147.456 MHz, which is six times the audio master clock frequency and 3072 times the audio sample rate of 48 kHz. When config-

- The instruction sequence `movec #0,sp`, `move #$40,r6` and `move #-1,m6` resets the **hardware stack pointer** and initializes the **software stack**.

- Setting `I PRP=#000003` allows the **interrupt system** to respond to interrupts from the audio interface at priority level 2. We do not use the core interrupts in the programs in this course.

- The **audio interface** is configured for operation in master mode with a master clock frequency of 24.576 MHz, in I<sup>2</sup>S network mode with two channels carrying 24 bit left-aligned data in 32 bit frames. Interrupts are set up for recep-

tion and transmission, error handling and 'last slot' (right channel). The receive and transmit control registers are set to the same values: `RCCR=TCR=$FDD302` and `RCR=TCR=$D17D00`. The registers `RSMA`, `TSMB`, `RSMA` and `TSMB` allow the network-mode channels (up to 32 of them) to be individually enabled or disabled for transmission or reception. By setting all these registers to the value `$00FFFF` we enable all possible channels, of which we use just two. Setting `PCRC=PRRC=$000FFF` configures Port C as an audio port. It is also possible to configure individual pins on port C as GPIOs that can be accessed via the audio connector.

- The **SHI interface** is configured in master SPI mode with a clock frequency of 0.9216 MHz. Interrupts are not used, `CPOL` and `CPHA` are both set to zero, and the narrow spike filter and the FIFO are disabled. For programming the SRC: 8-bit data, with `HCKR=$002048` and `HCSR=$000040`. For controlling the LED bargraph display: 16-bit data, with `HCKR=$002048` and `HCSR=$000044`.

- **Port H** is used for the SRC and the SEE-PROM and for setting the boot mode. PH4: GPIO input, lock signal from SRC4392. PH3: GPIO output, reset signal to SRC4392. PH2: used as MODC to select the boot mode of the DSP. PH1: GPIO output, chip select to the SRC4392. PH0: GPIO output, chip select to the M95M01 SEEPROM. This configuration is obtained by setting `PCRH=$000014` and `PRRH=$00000F`.

### 3. Final set-up before entering the audio loop

- Interrupts (and in particular the audio interrupts) are enabled so that the audio buffer is read and written and the flag synchronization system for received audio data is initialized.

- The SRC is configured over the SHI in SPI mode.

### 4. Interrupt service routine

- The interrupt service routine code forms

the final part of the framework program. The code is in the source file `esai4R2T.asm`.

### The audio loop

The audio loop contains the digital signal processing code and is embedded within the framework code. We will adopt the principle of reflecting the block structure of the processing to be done through the use of subroutines, where each processing block is embodied in one subroutine. The representation of a processing chain as a series of blocks will be familiar to analog designers.

Each of our subroutines therefore has 'input signals' and 'output signals'. In the example below we shall deal with four signals, called `SignalInL/R` and `SignalOutL/R`. Note that not all subroutines will have input signals: for example a signal generator will only have outputs. In our projects signals are represented by samples with 24 bits of precision, each occupying one location in memory within a dedicated area in the DSP's RAM. A subroutine may also have one or more parameters or settings that affect its operation. An example of such a parameter might be the time constant of a filter. **Figure 2** shows an example of how subroutines can be connected together.

There are four subroutines in the example, labeled from 'A' to 'D', each representing a signal processing block. The small rectangular boxes in the middle of the figure are the subroutines, connected together via signal paths. The boxes with rounded corners symbolise parameters of the subroutines. Some subroutines do not require any parameters: this is the case for subroutine C in the illustration. The pointed boxes contain the signals passed between the subroutines. These signals have a dedicated storage area in the DSP's memory which makes it possible to observe their values at any point in time. We have constructed the block diagram so that it is possible to examine the function of the system in more detail by observing the signal at these monitoring points. For example, we can copy any one of these signals to the audio buffer at the

end of the audio loop and then observe it as if we were using a digital audio oscilloscope. This is a great help when it comes to tracking down bugs in the signal processing code. To capture the output the author uses a simple low-cost USB sound card (available for considerably less than £100) in conjunction with a waveform editor program. Commercial waveform editors are available, or there are several free editors and analyzers available for download on the internet. The software allows one or more signals to be inspected in the time domain or converted to the frequency domain to allow a spectrum to be viewed, and thus can be used as a powerful replacement for an ordinary oscilloscope. Spectral analysis is particularly useful for analyzing distortion, for example in the output of a sine wave generator. It is also possible to use a waveform editor to create sound files and carry out various kinds of processing on them.

These files can then be loaded into a numerical computing package such as MATLAB after which the sky is the limit! Such a set-up could be used to test the characteristics of a dynamics processor using signal bursts of varying duration, frequency and amplitude. Since the DSP board includes a DAC, we could equally observe the signals using an ordinary analog oscilloscope, although this approach is better suited to giving an overview of a circuit's performance than a precise and detailed analysis.

The ADC is connected to SDI1 (SDO4) and the SRC to SDI2 (SDO3). The receive buffer, which occupies four locations in memory, is laid out as follows.

```
x:RxBuffBase
                                ADC, left channel
x:RxBuffBase+1
                                SRC.RX, left channel
x:RxBuffBase+2
                                ADC, right channel
x:RxBuffBase+3
                                SRC.RX, right channel
```

The DAC and the AES3 encoder, whose transmitter circuit is in the SRC, are connected to SDO0. The transmit buffer occu-



pies two memory locations with the following layout.

```
x:TxBuffBase
    DAC and SRC.TX, left channel
x:TxBuffBase+1
    DAC and SRC.TX, right channel
```

At the beginning and at the end of the audio loop there are two small sections of code. The first synchronizes with the audio clock and reads from the audio receive buffers; the second writes to the audio transmit buffers.

```
AudioLoop
    jclr    #RightRx,x:LRFlag,*
    bclr    #RightRx,x:LRFlag
    move    x:RxBuffBase,a
            ; ADC CS5340, left
    move    x:RxBuffBase+2,b
            ; ADC CS5340, right
    brset
#Lock_SRC4392,x:PDRH,NoSRC
    ; use ADC if SRC does not lock
    move    x:RxBuffBase+1,a
            ; SRC.RX SRC4392, left
    move    x:RxBuffBase+3,b
            ; SRC.RX SRC4392, right
NoSRC move    a,y:InL
    move    b,y:InR
```

The first two lines are responsible for synchronization, as described in the second installment in this series. The third and fourth lines read the values received from the ADC and store them in accumulator registers *a* and *b*. The fifth line checks the lock flag of the SRC. If this indicates that a valid audio signal is available, then the values in accumulators *a* and *b* are overwritten with the values read from the SRC; if no valid signal is available the overwriting is skipped. In lines 8 and 9 the accumulator contents are written to memory locations *InL* and *InR* from where they can be read by the first signal processing subroutine. This approach guarantees that we will always have a valid audio signal to use: if a digital signal is connected to the board then it will take priority, but if no valid digital signal is available then the system will fall back to using the signal at the ADC input.

The second section of code, at the end of

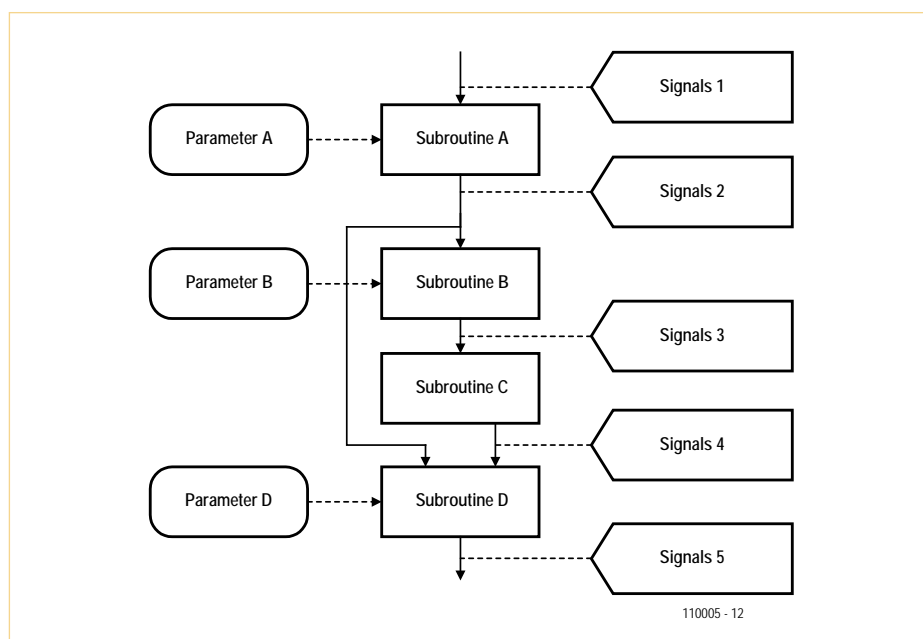


Figure 2. Example subroutines in an audio loop.

the audio loop, is responsible for writing the contents of accumulators *a* and *b* to the transmit buffer, the samples to be transmitted having been fetched from memory locations *OutL* and *OutR*. Finally the program jumps back to the start of the audio loop.

```
move    y:OutL,a
move    y:OutR,b
move    a,x:TxBuffBase
        ; -> DAC and SRC.TX, left
move    b,x:TxBuffBase+1
        ; -> DAC and SRC.TX, right
jmp     AudioLoop
```

## Memory map

**Table 1** shows the memory map used by the programs in this course. We have not expended much effort in ensuring that there are no gaps in the layout, preferring to adopt a common layout for all the programs to make them easier to compare with one another. Not all memory locations indicated here are used by every program.

## Address registers, software stack and sample counter

Two of the eight address registers *R0* to *R7* are used in the framework code for dedicated purposes and should not therefore be used as general-purpose registers by an application program. Register *R6* is used to point to the software stack in X-RAM. The software stack is an extension to the hardware stack provided by the processor and occupies RAM addresses from \$40

upwards. We make a total of 32 locations, from \$40 to \$5F, available to programs. The framework code uses the software stack in the audio interrupt service routine (ISR) to store registers *R0* and *M0* for the duration of the routine and restore them afterwards: it is important that the ISR does not alter any of the registers used by the main program. The software stack can also be used by application programs for temporary storage of processor registers, both those in the datapath and, more significantly, those in the address generation unit (AGU). Address register *R7*, which we refer to as the ‘sample counter’, is used to count modulo 192 at the audio sample rate. It is used in two of our example applications. In the test programs it is used to index into the two 192-entry sinewave look-up tables

```
Sinewave, f=1 kHz,
                x:$800 to x:8BF
Sinewave, f=2 kHz,
                y:$800 to y:8BF
```

from which sample values are fetched and fed to the DAC or to the digital output. The second use is in the digital signal level meter where it is used to trigger the regular writing of data to the LED bargraph. The display is updated at a frequency of  $48 \text{ kHz}/192 = 250 \text{ Hz}$ : in other words, every 4 ms. Register *R7* can be used for other purposes in an application program as long as it is saved on the stack and restored after use.

## Subroutines

A signal processing subroutine has the following structure.

```
NameOfSubroutine
    move    y:SignalInL, x0
    move    y:SignalInR, y0
    <signal processing code,
possibly dependent on parameters>
    move    x0, y:SignalOutL
    move    y0, y:SignalOutR
    rts
```

The subroutine is called using the instruction `jsr NameOfSubroutine`. The `jsr` (jump to subroutine) instruction tells the DSP to push the program counter and processor status register onto the hardware stack and branch to the subroutine. The subroutine terminates with an `rts` (return from subroutine) instruction which causes the processor to restore the program counter and status register from the hardware stack, thus allowing execution of the calling code to continue. Parameters may be passed to the subroutine using the software stack.

## Macros

The DSP assembler has a text macro facility. The contents of a macro are inserted into the program code at each point where it is called. A macro is declared as follows.

```
NameOfMacro macro    param1 param2
... paramN
...
endm
```

A macro is called by simply giving its name followed by any parameters. The use of macros can make programs easier to read, especially when parameters are used. However, they do not result in a reduction in code size. This is in contrast to subroutines, which can help both to make code more readable and to make it smaller. On the other hand, subroutines consume processor clock cycles in the stack operations involved in calling them and returning from them.

## Initialization of the signal processing code

Two subroutines are called immediately before the audio loop is entered. The first

is the routine `ZeroState`, which sets the memory storing the state of the signal processing routines to zero. The second is called `SetDefaultParams` and sets up the default values of the signal processing parameters. The values used in this routine can be changed to adapt the system for a particular application; alternatively, parameter values can be modified using an input device connected to the SPI port.

## Loading and executing the program

Once a DSP program has been written it must be assembled and checked for errors. This can be done in the command window using the following command.

```
asm56300 -a -b -l myprogram.asm
```

The `-l` option tells the assembler to generate a listing file, which is often useful when trying to locate errors. Assuming there are no errors, the assembled code in the file `myprogram.cld` can be loaded into the debugger to which the DSP board is connected. From there, using an adapter, the code can be transferred to the DSP itself and executed.

## Handy hints

### Using the assembler

- The assembler interprets text starting in the first column of a line as a label. A label must begin with a letter, and reserved words such as `move` or `a0` are not allowed as labels.

- The assembler is of the ‘two pass’ variety. A two-pass assembler first looks through the code building a ‘symbol table’, which is a list of the locations of all labels used. In the second pass it assembles the code itself, using the values stored in the symbol table to resolve ‘forward references’, where a label is defined after it is first referred to in the assembler source file. Without the symbol table from the first pass it would not be possible to resolve these labels as their values would not be known.

- Labels starting with an underscore (`‘_’`) are useful when writing macros: their scope

is local to the macro itself.

- Labels, unlike keywords, are case sensitive.
- DSP instructions can appear from the second text column onwards.

- The file `mioequ.asm` can be included in a DSP program using an `include` directive. This file defines a large number of abbreviated names for the interface registers in the DSP.

- The first usable address for program code is `p:$100`: below this address are the interrupt vector table and other reserved areas.

- The `org` directive can be used to specify the use of X, Y, P or L memory.

- The processor has an instruction pipeline which greatly speeds up code execution as long as it is not disturbed by program branches or long interrupts. Because of this pipeline, the assembler is sometimes obliged to introduce additional `nop` instructions into the code. When this happens it reports a warning to the programmer that there may be an opportunity to improve the efficiency of the code by reordering it so that the `nop` instructions can be replaced by instructions that do useful work.

At the end of the processor manual there is a number of so-called ‘programming sheets’ which greatly simplify programming the processor’s registers. We would recommend any programmer to take advantage of these sheets: they can easily be printed off from the PDF manual, used during the development of a program, and then finally attached to the other program documentation.

## Programming the DSP

- It is not possible to move immediate data directly to memory: instructions like `move #$123456,x:$000100` do not exist. Instead, a register must be used as a staging-post, for example using `move #$123456,x0` followed by `move x0,x:$000100`. There are, however, special instructions for moving values directly to addresses in the peripheral register area

**Table 1. Memory map used by the software in the course**

X-RAM use	Y-RAM use	Region	P-RAM use
Audio RX buffer	Signals	\$00 to \$0F	Interrupt vector table and reserved areas
Audio TX buffer	Signals	\$10 to \$1F	
Audio flags and pointers	Pointers, coefficients, status storage	\$20 to \$2F	
Program parameters		\$30 to \$3F	
Software stack	Free for application use	\$40 to \$4F	
Software stack	Free for application use	\$50 to \$5F	
Free for application use		\$60 to \$9F	
Auxiliary variables		\$A0 to \$BF	
SRC boot sequence	Free for application use	\$C0 to \$CF	
Free for application use		\$D0 to \$FF	Application program
Filter and polynomial coefficients		\$100 to \$5FF	
Circular buffer, left	Circular buffer, right	\$600 to \$7FF	
Sine wave, 1 kHz	Sine wave, 2 kHz	\$800 to \$8BF	
Free for application use		\$8C0 to \$17FF	

at the top of the memory map, such as `movep $123456, x: $FFFFE0`.

- A register must again be used for intermediate storage when moving data from one memory location to another, for example `move x: $000010, x0` followed by `move x0, y: $000010`. However, memory-to-memory transfers within the peripheral register area are possible.
- The instruction `move # $F, x0` does not result in the perhaps expected outcome `x0 = $00000F`, but rather in `x0 = $0F0000`. The instruction `move # > $F, x0` gives the right-aligned result. The DSP is designed to work with fractional values, which means that numbers are in general left-aligned.
- Nested ‘do’ loops must use distinct labels. Furthermore, there must be at least a `nop` instruction between the two labels if otherwise no code would appear there.
- P-RAM should not normally be used for data storage as accessing it incurs an extra penalty in processor cycles.
- When integer values are multiplied, for example in address calculations, it is necessary to halve the result by arithmetic-shifting it right one place using an `asr` instruction. This again is because the DSP’s multiplier is designed for left-aligned fractions, not right-aligned integer values.
- The three-operand instructions `mac` and `mpy` do not allow all sixteen possible combinations of source operand registers to be

used. This is because there are only three bits available in the instruction encoding to specify them rather than the four that would be required. So for example `mpy x0, x0, a` and `mac x0, y1, b` both exist, but neither `mpy x1, x1, a` nor `mpy y1, x0, b` is possible.

- Fast interrupts should not be terminated by an `rti` instruction.
- We have reserved address register R6 for use as a pointer to the software stack, which starts at address X: \$40. The software stack is particularly useful when manipulating the AGU registers in a subroutine. It is a common schoolboy error to set up an address register for modulo addressing inside a subroutine and forget to reset its mode before leaving the routine, giving rise to unexpected operation elsewhere in the code.
- The hardware stack pointer `sp` should be reset to zero during initialization.
- In particularly time-critical applications it is best to try to avoid the use of instruction extensions. An instruction extension is a second word added to an instruction, for example containing an address or an immediate value. The instruction `mpyi #0.3, x1, a` requires an extension word to store the immediate value 0.3 and so the instruction occupies a total of two words in program memory. In contrast, the instruction `mpy x0, x1, a` only occupies one word of program memory. It is normally preferable to store constants in the DSP’s RAM during program initialization and load

them into a processor register (in this case the operand register `x0`) shortly before they are required: this can often be done using a parallel move instruction.

- Looking at a program for the DSP it is clear that a very large number of `move` instructions are used. One of the reasons for this is that the CPU is register-based. This in turn means that it is important to take full advantage of the DSP’s capability to do parallel moves. In general this means that a value is loaded into a register in advance of (sometimes considerably in advance of) its use, whenever there is a suitable arithmetic operation available to which the move can be attached. Unfortunately this kind of optimization makes program code rather hard to follow, since different parts of what might be considered a single operation are separated from one another in the source file.
- A few years of experience of programming DSPs has taught us that the commonest programming error is inadvertently using the same memory location for more than one purpose. We therefore recommend that programmers draw up a detailed and exhaustive memory map for each program they write. Also, our preferred approach of allocating dedicated memory areas to groups of variables, while not exactly economical in memory use, nevertheless can help avoid many simple errors. For later programs it will prove useful to write two routines to push and pop the ten datapath registers (`x0, y0, x1, y1, a0, b0, a1, b1, a2, and b2`) to and from the software stack. These routines can then be used in



Table 2. Two's complement integer and fractional representations

Integer	Decimal	Integer	Decimal	Fractional	Decimal	Fractional	Decimal
0000	0	1000	-8	0.000	0	1.000	-1.0
0001	1	1001	-7	0.001	0.125	1.001	-0.875
0010	2	1010	-6	0.010	0.25	1.010	-0.75
0011	3	1011	-5	0.011	0.375	1.011	-0.625
0100	4	1100	-4	0.100	0.5	1.100	-0.5
0101	5	1101	-3	0.101	0.625	1.101	-0.375
0110	6	1110	-2	0.110	0.75	1.110	-0.25
0111	7	1111	-1	0.111	0.875	1.111	-0.125

subroutines to avoid unwanted side effects on these registers should they be used by the calling program.

### How samples are represented in the DSP

Signal samples in the DSP are represented using a special numeric format. The binary point is immediately after the top bit, which is used as the sign bit. The only difference between this format and the usual representation of integers is that with normal integers there is no binary point (or you can imagine a binary point immediately to the right of the integer value). The DSP's format is called 'two's complement fractional', or just 'fractional', representation. The difference between two's complement representation of integers and the fractional format can be illustrated simply using four-bit values. There are sixteen different four-bit values, of which seven represent positive numbers and eight represent negative numbers, leaving one representation of zero: see **Table 2**.

For clarity we have marked the sign bit in the table and included the binary point in the fractional representations. The values represented differ only in the significance of each bit. From left to right, the bits after the sign bit in the integer representation have significances of 4, 2 and 1 while in the fractional representation the bits after the binary point have significances of  $1/2$ ,  $1/4$  and  $1/8$ . In the integer format values are naturally right-aligned while in the fractional format they are naturally left-aligned. This means that we can extend an integer without changing its value by adding zero

bits to the left (if it is positive) or one bits to the left (if it is negative); we can extend a fraction without changing its value by adding zero bits to the right, whether the value is positive or negative. Observant readers will have noticed that the extreme positive and negative representable values have different magnitudes. This sometimes causes difficulties as the clipping level depends on the sign of the signal. In the case of our 24-bit DSP the difference in magnitude is just  $2^{-23} = 1.1921 \times 10^{-7}$ , which is in many cases small enough to be ignored.

Now let us look at how the fractional representation is used. We will imagine a DSP-lite working with four-bit values (single precision) and eight-bit values (double precision). We will interpret the values in exactly the same way as the full-fat 24-bit DSP: the only difference will be in the resolution (that is, the difference between consecutive representable values). One common job the DSP has to do is 'requantization'. In our example this means converting an eight-bit value to a four-bit value. Suppose the eight-bit value is 0.100 0100, representing a value of  $1/2 + 1/32 = 0.53125$ . One way to requantize this value is simply to drop the last four digits, leaving the value 0.100, which represents  $1/2 = 0.5$ . That might not seem too impressive, but readers are invited to consider how a similar task would be done with integer values.

Back to the real DSP: in single precision it calculates to 24 bits of accuracy and in double precision to 48 bits. We know that the preferred basic word length for digital audio processing is 24 bits, corresponding to sin-

gle precision on the DSP. But what happens when we multiply together two signal sample values or if we want to implement a simple 20 dB attenuator? In both cases the multiplication produces a result in double precision with a word length of 48 bits. Compare with multiplying two numbers on the four-bit DSP:

```
0.010 * 0.001 representing
0.25 * 0.125 = 0.03125
0.000 0100 representing 0.03125
```

The result has eight digits.

With this in mind we can revisit one of the hints given earlier. If we want to multiply two integer values together, for example in a memory address calculation, the result from the DSP's multiplier has to be shifted to obtain the correct answer. Looking again at the example above, thinking of the values as integers:

```
0010 * 0001 representing 2 * 1 = 2
0000 0100 representing 4,
the result produced by the
fractional multiplier
0000 0010 representing 2,
the correct result obtained
by arithmetic-shifting the
multiplier's output right by
one place
```

### Coming soon...

Now that we have described the structure of DSP programs, we will move on in the next installment to make the DSP board into an audio signal generator.

(110005)

**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**

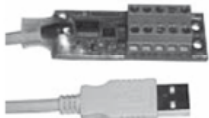
- Multipurpose: 7 interfaces
- Royalty-free, robust USB drivers
- No in-depth knowledge of USB required
- Standard 18-pin DIP interface; 0.6x1.26-inch footprint



Only \$28.95!

**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](http://www.dlpdesign.com)

**WIRELESS MADE SIMPLE®**  
QUICKLY AND COST-EFFECTIVELY MAKE YOUR PRODUCT WIRELESS

**RF MODULES**  
LOW COST • LONG RANGE



TRANSMITTERS • RECEIVERS  
• TRANSCEIVERS

**INTEGRATED CIRCUITS**  
ENCODERS • DECODERS • TRANSCODERS



COMMAND • CONTROL  
• KEYLESS ENTRY

**OEM PRODUCTS**  
EASILY CUSTOMIZED



**ANTENNAS**  
FROM WHIPS TO CHIPS



**Featured Products:**

**LT Series Transceiver**

• 315 MHz  
• 418 MHz  
• 433 MHz

- Data Rates to 10K bps
- Direct Serial Interface
- Low Power Consumption
- Low Cost
- Long Range
- Ultra Compact

Ideal for the bi-directional wireless transfer of serial data, control, or command information in the favorable 260-470MHz band. Its advanced synthesized architecture delivers outstanding stability and frequency accuracy.

**Function Relay Module**

Provides four latched or momentary relay outputs that are capable of switching external AC or DC powered loads of up to 5 amps.

- 315 MHz
- 418 MHz
- 433 MHz
- Dual Power Inputs
- 256 Unique Addresses
- Wide Operational Voltage
- FCC and Industry Canada Pre-Tested



**LINX**  
TECHNOLOGIES

800-736-6677

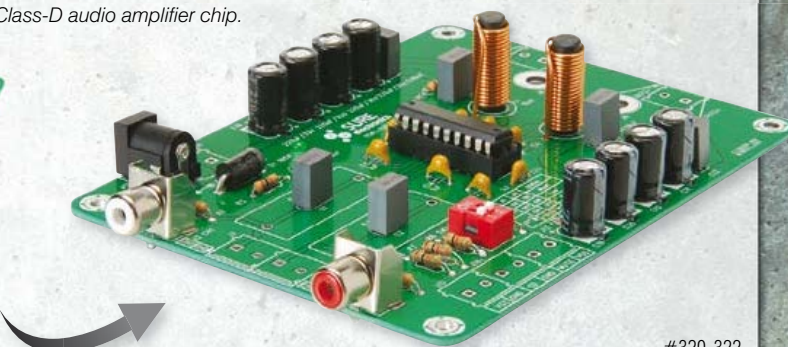
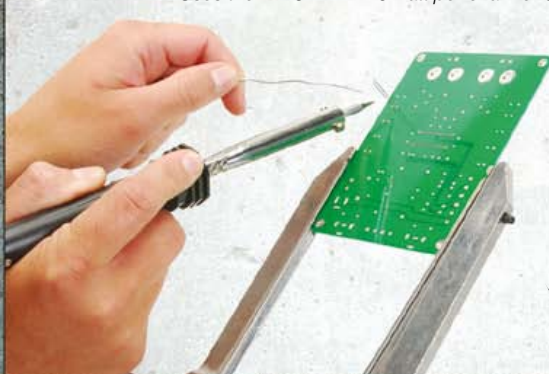
159 Ort Lane  
Merlin, OR 97532

[linxtechnologies.com](http://linxtechnologies.com)

**SURE** - you can do it yourself!

**NEW! Build this Sure Electronics 30 watt Class-D power amplifier for pristine, accurate audio reproduction!**

Uses the TPA3122D2 15 watt per channel Class-D audio amplifier chip.



#320-322

Includes board and all components. Assembly is required.

Exclusively distributed by:

**PARTS**  
**EXPRESS**  
YOUR ELECTRONICS CONNECTION

1-800-338-0531

725 Pleasant Valley Dr. Springboro, OH 45066

Sure Electronics designs amplifier boards for the audio enthusiast who would rather build it than buy it. For a complete line of Sure Electronics audio amplifier boards and power supplies, visit [parts-express.com/sure](http://parts-express.com/sure)

**SURE**  
electronics



# OnCE/JTAG Interface

## Program and debug Freescale DSPs

By Ton Giesberts (Elektor Labs)

To help with the programming and debugging of the DSP board used in our DSP course, Elektor Labs designed a small adapter that can be used to connect the board to a PC via a modern, fast USB connection. This adapter can also be used in combination with other Freescale DSPs from the DSP56K series.

For the DSP course we developed a circuit based around a DSP from the Symphony series from Freescale, the DSP56374. The programming and debugging of the DSP happens via a JTAG interface, for which Freescale uses their own 14-pin connector, known as OnCE (an abbreviation of On-Chip Emulation). To keep the DSP board as compact as possible, no direct interface for connection to a PC was added to the board. Freescale does have several programming adapters for sale, but they tend to be rather expensive. Those of you who have already used an evaluation kit from Freescale will probably have a suitable programming adapter as well. Furthermore, there are several simple (home-built) programmers for these Freescale DSPs to be found on the Internet. Most of these, however, work via a parallel port and there aren't that many (new) computers that still come provided with such a port. The OnCE/JTAG interface described here is an addition to the DSP board, and provides a fast and modern USB connection between the DSP board and a PC.

### The circuit

For the USB interface we've used a Hi-Speed Dual USB UART/FIFO IC from FTDI, the FT2232H (IC1, see **Figure 1**). This IC comes from the latest generation from FTDI (refer to the datasheet in [1]). With the help of this IC the interface can communicate with

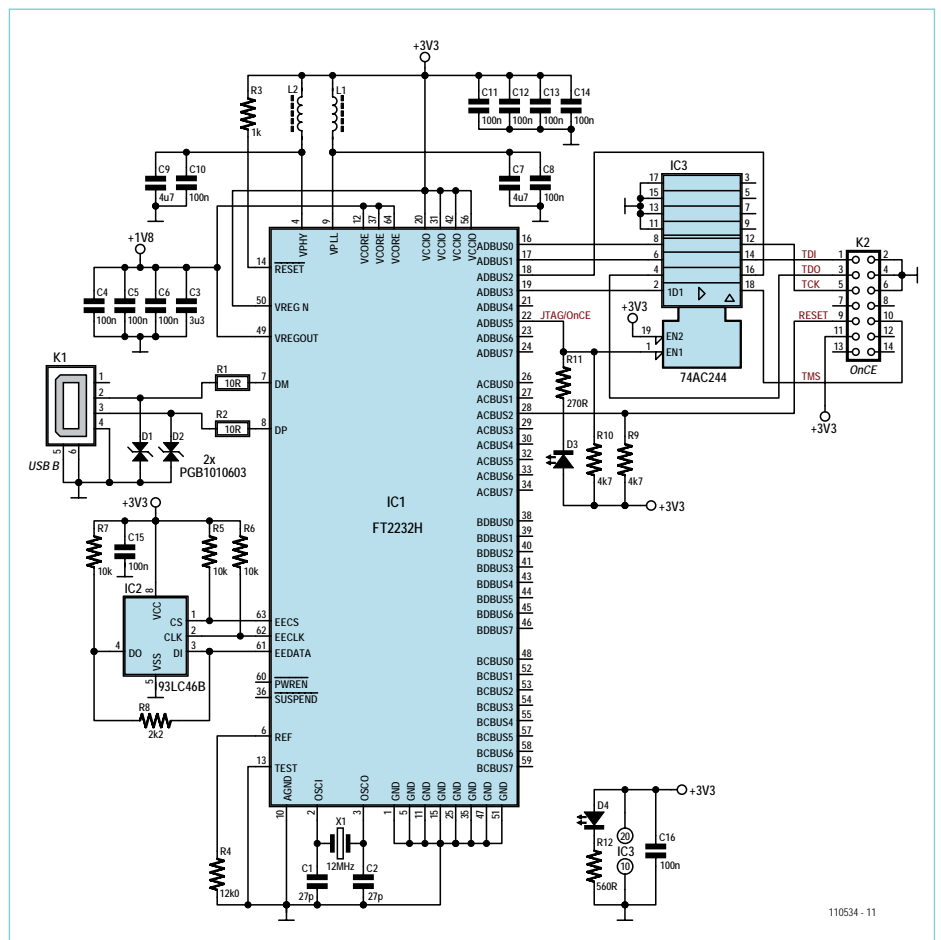
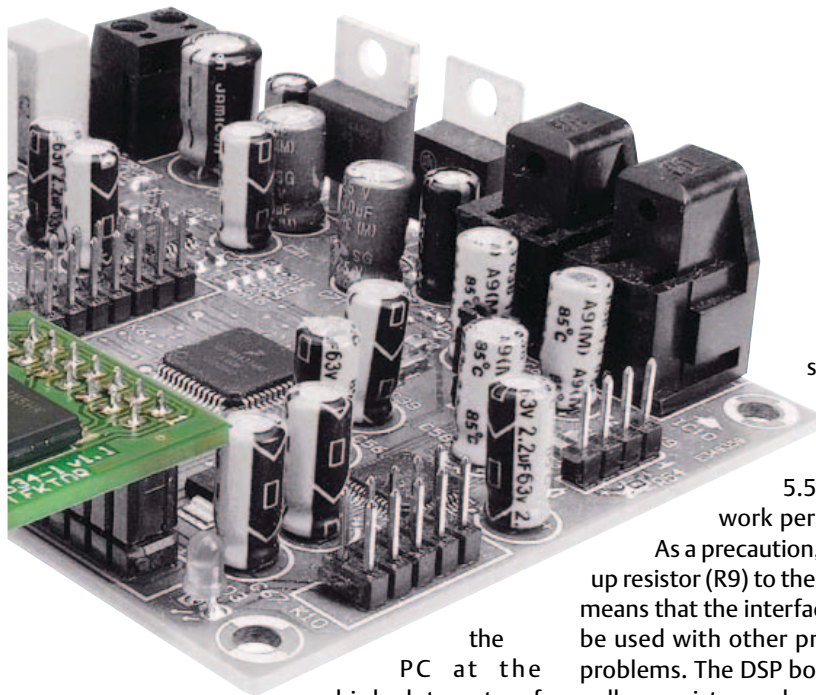


Figure 1. The circuit diagram of the programming adapter, with an FT2232H made by FTDI at its heart.





the PC at the high data rate of USB2.0. The supply for the circuit comes from the DSP board itself, which makes the interface Self Powered. This makes the circuit a bit simpler, since the supply voltage for the I/O interface is 3.3 V and this is already present on the DSP board. The core of the FT232H operates at an even lower voltage, which is 1.8 V. This is obtained from an on-chip voltage regulator, which is a welcome feature. The connection to the USB bus requires nothing more than two resistors and a USB Type B connector (K1). Two ESD suppressors (D1 en D2) have been added next to the USB connector, which protect against static charges. The reaction speed of these diodes is less than 1 ns. Because of their extremely low capacitance (just 0.055 pF) they don't affect the USB signals. The active-low reset line of the IC isn't used and is pulled up via R3 to the supply voltage. The external EEPROM (IC2) is connected according to the standard application note (R5 to R8) to the FT232H. The EEPROM should be one with a word size of 16 bits and which operates from a supply voltage of 3.3 V. There is no need for it to have a selectable memory configuration input (usually called ORG). The 93LC46B made by Microchip is a suitable candidate for this application. The supply voltage for the FT232H has been decoupled extremely well, as can be seen from the 12 capacitors and two inductors used (C3–C14 & L1/L2).

We decided to buffer all the output signals, with the exception of the reset signal. When the adapter hasn't yet been activated in the development environment the buffered outputs will be in the off state and will have high-impedance outputs. The fast 74AC244 (IC3, an octal buffer/line-driver with tri-

state outputs) is suitable for use with supply voltages between 1.5 V to 5.5 V and will therefore work perfectly well at 3.3 V.

As a precaution, we've added a pull-up resistor (R9) to the reset output, which means that the interface should be able to be used with other projects without any problems. The DSP board already has this pull-up resistor on-board. If the adapter is going to be used exclusively with the DSP board then R9 may be left out.

In **Figure 2** you can see the PCB that was designed for the adapter. Its size has been kept small by using SMD components. The 2x7 pin header is mounted on the underside of the board, which makes it easy to plug the adapter into connector K8 of the

DSP board. Apart from the bare PCB, Elektor also supplies a fully populated and tested version of this adapter [2].

### Software for Symphony Studio

When the circuit was designed we assumed that the adaptor would be used in combination with the development environment from Freescale, Symphony Studio. With the help of a template from Freescale the FT232H can be programmed such that it will be recognized as a Symphony Sound-Bite. This will have been done for you in the fully populated board supplied by Elektor so that you can plug the adapter into the DSP board straight away. For those of you who want to program the FT232H themselves, FTDI provides a utility called Mprog that will let you do this. The template from Freescale is meant to be used with this program. (There is a more recent program from

## COMPONENT LIST

### Resistors (SMD 0805, 0.1W)

R1, R2 = 10Ω 5%  
R3 = 1kΩ 5%  
R4 = 12.0kΩ 1%  
R5, R6, R7 = 10kΩ 5%  
R8 = 2.2kΩ 5%  
R9, R10 = 4.7kΩ 5%  
R11 = 270Ω 5%  
R12 = 560Ω 5%

### Capacitors (SMD 0805)

C1, C2 = 27pF 50V 5%, NP0  
C3 = 3.3μF 10V 10%, X5R  
C4, C5, C6, C8, C10–C16 = 100nF 50V 10%, X7R  
C7, C9 = 4.7μF 6.3V 10%, X5R

### Spoelen (SMD 0805)

L1, L2 = 600Ω @ 100MHz,  
200mA/0.35Ω (e.g. Murata  
BLM21BD601SN1D)

### Semiconductors

D1, D2 = PGB1010603, V<sub>clamping</sub> = 150V  
(Littelfuse, SMD 0603)  
D3 = LED, green (Kingbright KPHCM-2012CGCK, SMD 0805)  
D4 = LED, red (Kingbright KPHCM-2012SURCK, SMD 0805)  
IC1 = FT232HL-R (FTDI, SMD 64-pin LQFP)  
IC2 = 93LC46B/SN (Microchip, SMD SO-8)  
IC3 = CD74AC244M (Texas Instruments, SMD SO-20)

IC3 = CD74AC244M (Texas Instruments, SMD SO-20)

### Miscellaneous

K1 = USB-B connector, right angled, PCB mount  
K2 = 14-pin (2x7) IDC socket, lead pitch 0.1 in.  
X1 = 12MHz, C<sub>load</sub> 18pF ±30 ppm, HC-49S  
PCB # 110534-1  
Assembled, programmed and tested board: # 110534-91

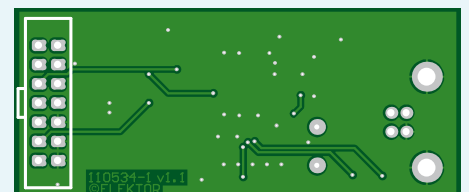
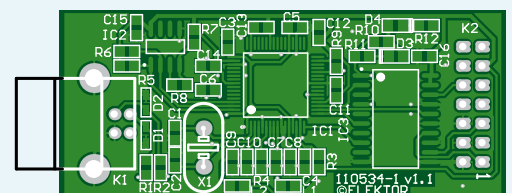


Figure 2. The PCB for the circuit has been designed such that it can be plugged in directly into connector K8 on the DSP board.

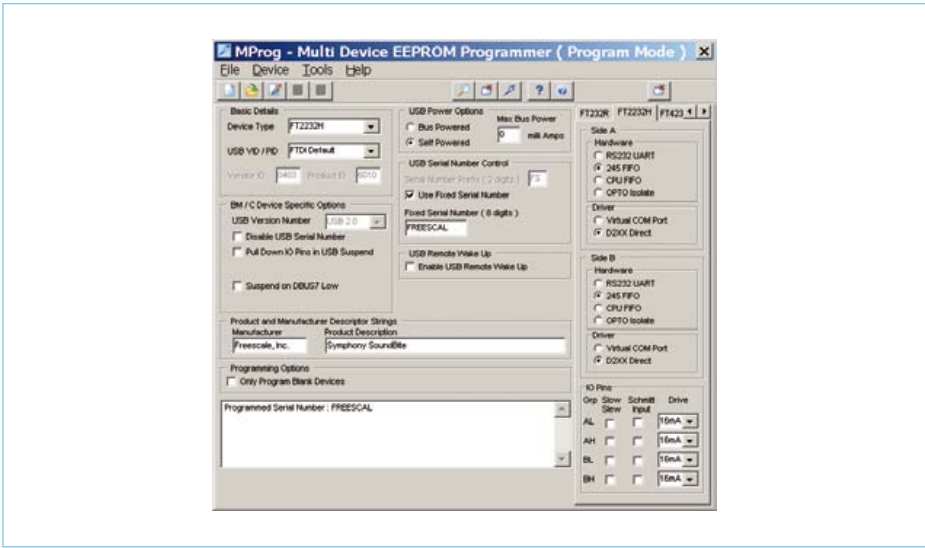


Figure 3. Screenshot of the program used to program the FT232H.

FTDI, FT\_Prog, but the template from Freescale can't be used with this.) The most recent version of Mprog (3.5) can still be downloaded from the FTDI website [3]. One problem is that this program doesn't support the Single Channel version of the USB interface, the FT232H. In order to make it easier to program the interface, and avoid problems with setting up a template, we decided to use the (unfortunately more expensive) dual-channel version. We have already adapted the original SoundBite template in a few places (see **Figure 3**) and this can be used directly. If you first carry out a *Scan* (under *Device*) you can check in the status window if the program recognizes

the interface. You can also verify that it has not yet been programmed and if any other devices have been found. If the interface is found it can be programmed, as long as it's still in the unprogrammed state, otherwise it needs an *Erase* first. When you want to build the interface yourself and/or program the FT232H you should do the following:

From the *File* menu select 'Open' and search for the template 110534-1.ept, which can be downloaded from our website (110534-11.zip, see [2]). If you want to make changes to the template you should select *Edit* from the *File* menu after the template has been opened. The following changes

have been made by us in the original template from Freescale: We used an FT232H instead of an FT232D as the Device Type. For the USB Power Options we selected Self Powered. When the FT232H has been selected a tab automatically appears on the right with a range of settings for the I/O pins. The choice for Hardware (Side A) should be '245 FIFO' instead of the standard 'RS232 UART'. The driver is already set correctly to D2XX Direct. Make sure that the driver has already been installed [4]. In the datasheet for the FT232H are two examples of applications that are Self Powered. In both of these a voltage divider is used to detect the 5 V bus voltage. In the text, it is mentioned that the option 'suspend on DBUS7 low' in MProg should be selected. However, this only works if this voltage divider is present. If you were to select this option with our circuit you'd find that the PC couldn't detect the circuit any more. To get round this it seems that connecting pin 46 (BDBUS7) to the 3.3 V supply is a solution, rather than pin 59 as mentioned in the datasheet. For the connection you could use a piece of 0.1 mm enamelled copper wire, if necessary. Make sure that you don't cause any shorts since the pins of the IC are only 0.5 mm apart. All I/O pins are set to work at their highest current rating, 16 mA. This

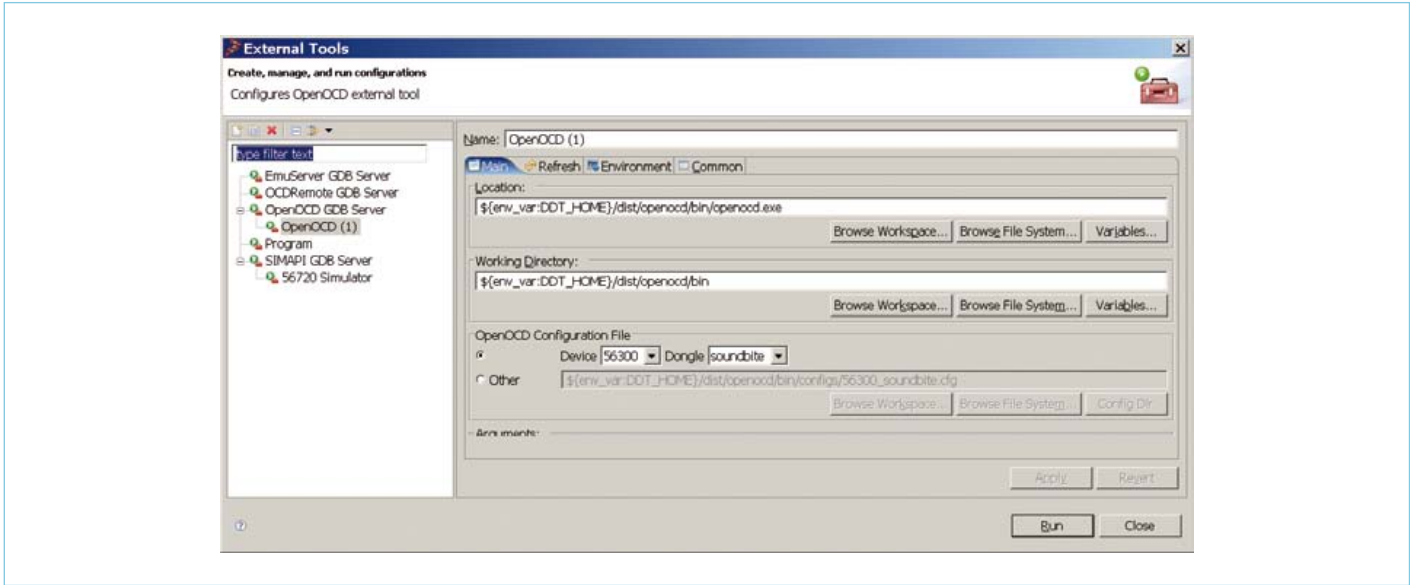


Figure 4. Here you can see how the adapter is selected in Symphony Studio.

## Current consumption

Several measurements were carried out on the prototype to determine its current consumption. When the DSP board without the interface is connected to 5 V (digital 5 V and analog 5 V on the DSP board are connected together via a small choke), the current consumption was found to be about 84 mA. With the programming interface connected this rose to about 87 mA when power was applied to the board. At that stage the FT2232H is still in Suspend mode and uses only a few hundred  $\mu$ A. Once the interface is connected to the PC, the total current consumption depends on the USB speed: 135 mA (Full Speed) or 155 mA (High Speed). Once the interface has been configured in Symphony Studio the latter value rises to about 160 mA. The extra 5 mA is caused by the green LED that indicates that the OnCE output is active. When, for example, one of the test programs is tried out, such as `tst_src2.asm`, the total current consumption rises to about 272 mA (with the SRC activated and optical digital audio signals).

way the green LED (D3) is driven better. The current through the LED is typically 4.7 mA. The current through the red LED (D4) is less, about 2.7 mA, which makes the brightness of both LEDs appear equal to the eye. If you want to program a modified template you should first save it. It's best if you never overwrite the original template, so choose a new name for it (select *Save As ...* under *File*). After opening a template it can be programmed immediately.

In the configuration for Symphony Studio we have to select 'soundbite' as our interface. From the C/C++ Perspective choose *Run*, *External Tools*, and *External Tools* again, *OpenOCD GDB Server* (double-click the very first selection). In the *Main* tab you have to select '56300' for the Device and 'soundbite' for the Dongle in the OpenOCD Configuration File section (see **Figure 4**).

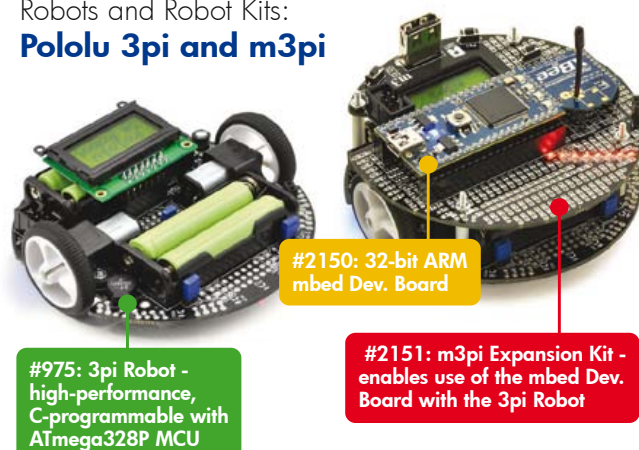
To make the use of the circuit as simple as possible it was decided to design the board as a plug-in module. The connector to the DSP board (K2) is a socket (female connector) and is mounted on the underside of the board. The USB connector is a standard Type B version for PCB mounting. For the connection it's best to use a USB2 cable; this will be specifically stated on the cable.

(110534)

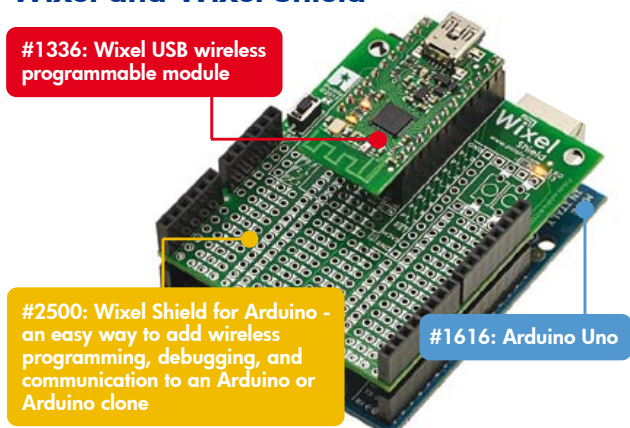
## Internet Links

- [1] [www.ftdichip.com/Products/ICs/FT2232H.htm](http://www.ftdichip.com/Products/ICs/FT2232H.htm)
- [2] [www.elektor.com/110534](http://www.elektor.com/110534)
- [3] [www.ftdichip.com/Support/Utilities/MProg3.5.zip](http://www.ftdichip.com/Support/Utilities/MProg3.5.zip)
- [4] [www.ftdichip.com/Drivers/CDM/CDM20814\\_Setup.exe](http://www.ftdichip.com/Drivers/CDM/CDM20814_Setup.exe)

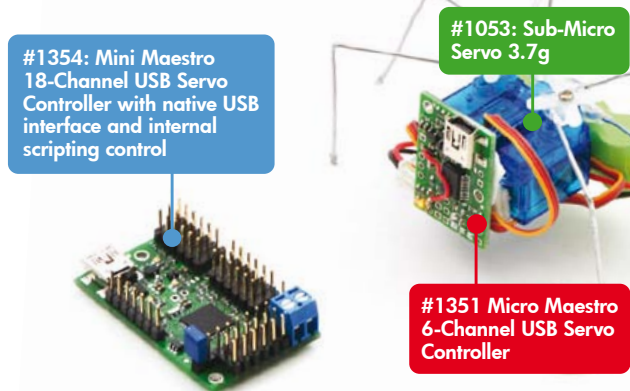
## Robots and Robot Kits: Pololu 3pi and m3pi



## Programmable Controllers: Wixel and Wixel Shield



## Hobby/RC Servo Controllers: Micro and Mini Maestros



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](http://www.pololu.com)





1



2



3

# Working with Stencils

By Thijs Beckers (Netherlands Editorial / Elektor Labs) and Antoine Authier (Elektor Labs)

Most electronics professionals know all the ins and outs of making PCBs, but we want to show you a technique that is probably not familiar to many electronics hobbyists: using stencils to apply solder paste.

When you order a PCB (or a panel) designed for predominantly SMD components, some manufacturers offer you the option of ordering a stencil at the same time. The purpose of the stencil is to considerably simplify, improve and accelerate the process of applying solder paste. The PCB panel for the Elektor BOB project provides an excellent example for demonstrating this.

## Figure 1

These are the components that you receive in a stencil kit: the PCB (front), the stencil (right) and the holder with the clamps (left).

## Figure 2

The PCB and the stencil are held in the proper position using these specially shaped pins.

## Figure 3

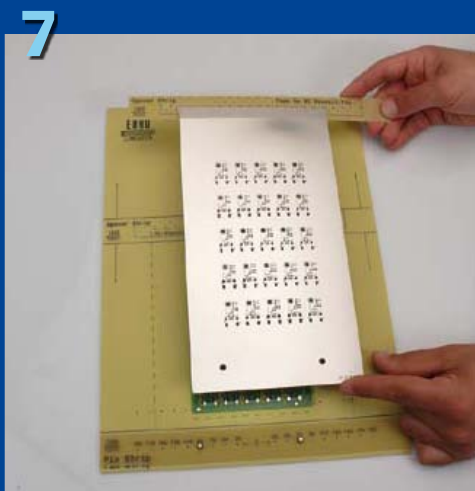
First we press the cone-shaped parts onto the strip of board material, and then place it on the holder PCB.

## Figure 4

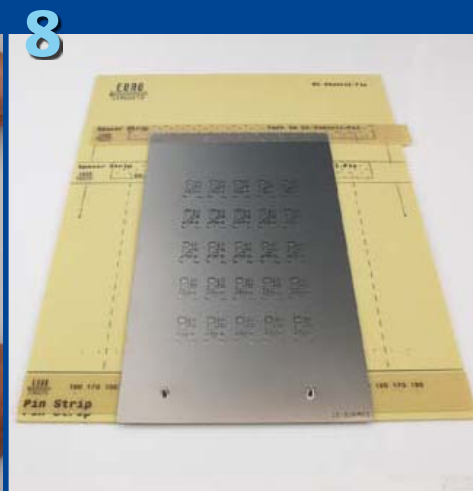
Next, we fit the board holder pins on the holder board.

## Figure 5

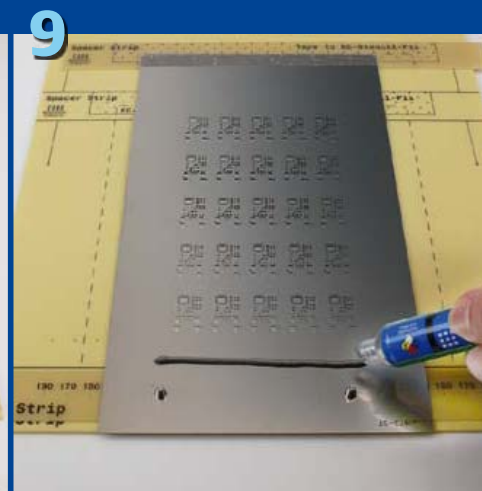
Now we can fit the PCB on the holder board. It's important to ensure that the board is nice and clean. Fingerprints or other dirt on the pads are taboo.



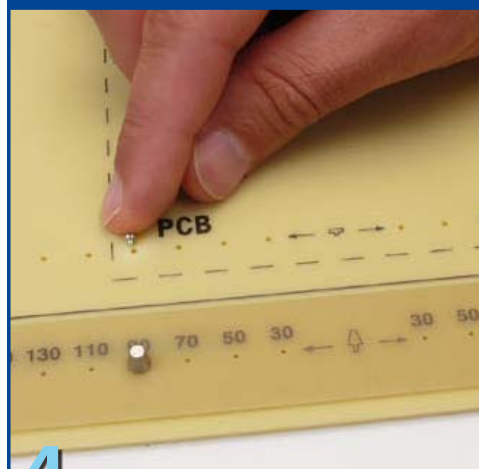
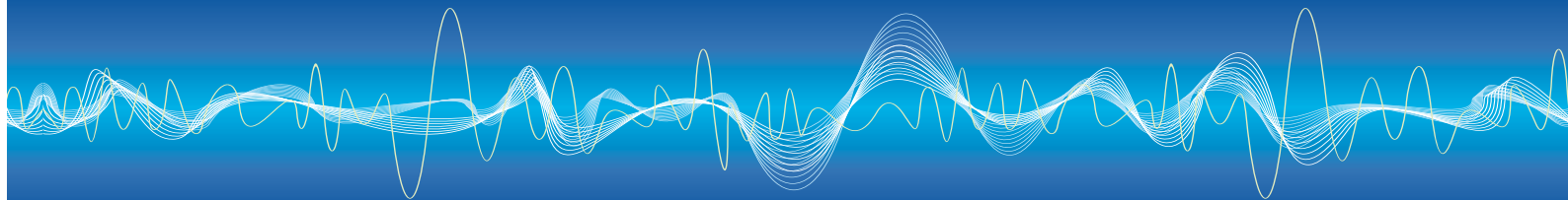
7



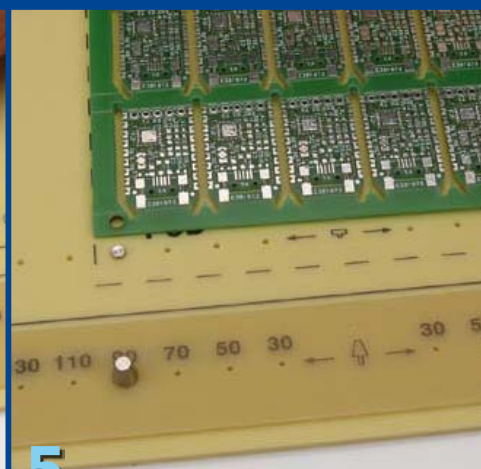
8



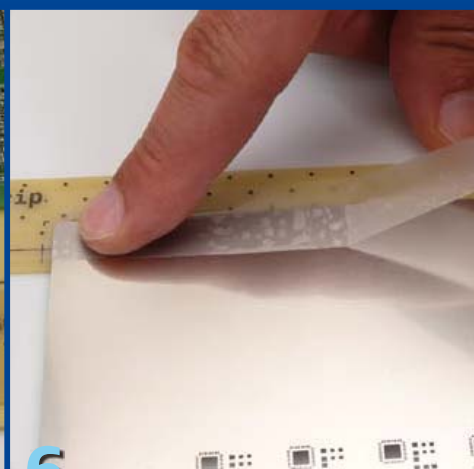
9



4



5



6

### Figure 6

We must ensure that the stencil can be laid on the PCB properly, without any curled corners. To achieve this, we tape the top edge of the stencil to a piece of board material with the same thickness as the PCB.

### Figure 7

Now we lay the stencil on top of the PCB.

### Figure 8

The cone-shaped pins hold the stencil exactly in place. As you can see, the solder pads are exposed.

### Figure 9

Now we can apply the solder paste. Here we only intend to apply paste to the first row of PCBs, so we don't want to put too much paste on the stencil.

### Figure 10

The next step is to use a squeegee to spread the solder paste over the stencil in a *single* motion. Pulling the squeegee towards you yields the best results.

### Figure 11

If everything went right, we now have a board with perfectly applied solder paste after the stencil is removed.

### Figure 12

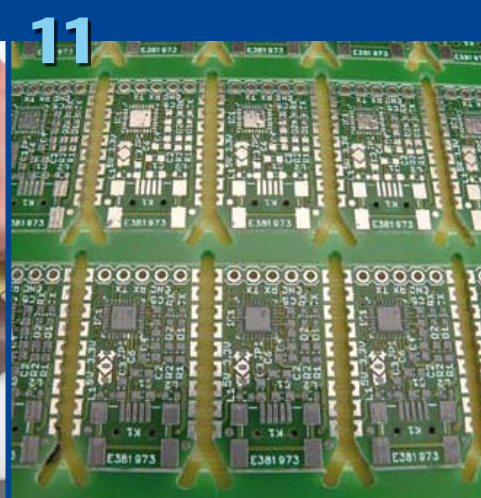
In this close-up you can see that even on the small pads for the IC, exactly the right amount of solder paste has been applied in the right place (note that the IC pin spacing here is only 0.5 mm).

Now the board is ready for the components. After they have been placed, we can put the assembly in the reflow oven — but that's a different story.

(110514)



10



11



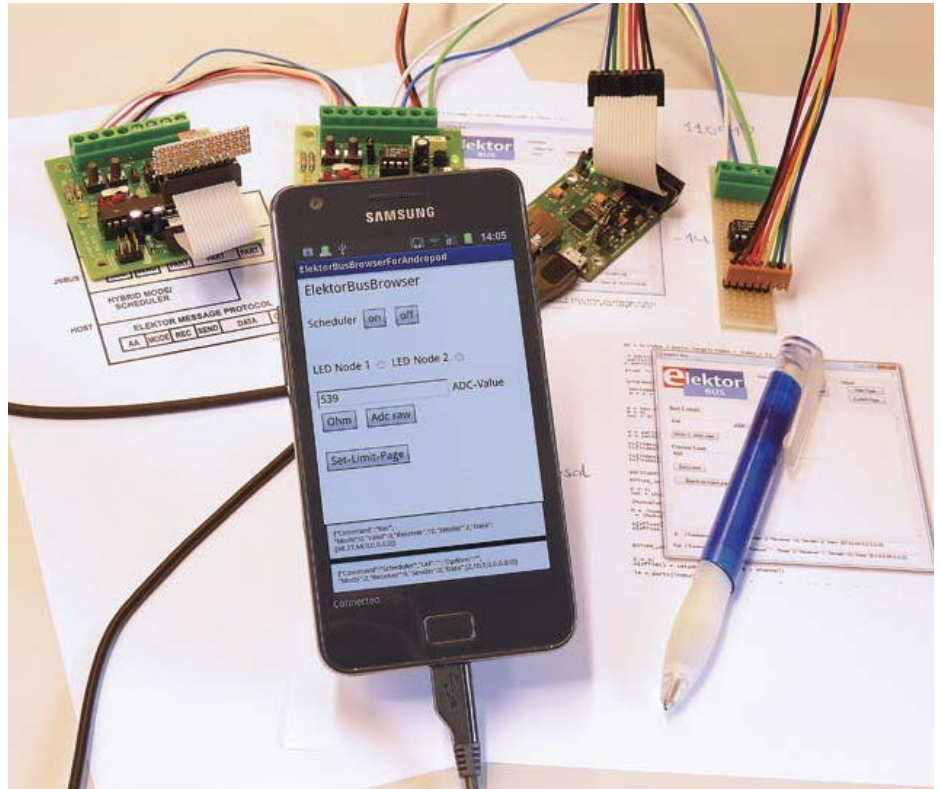
12



# Here comes the Bus! (9)

## Rapid application development

Any reader wanting to create their own application for the ElektorBus has so far had to be content either with making small modifications to the demonstration software we have given, or has had to start from scratch, guided by our examples. Here we change all that: we show how you can rapidly develop an application with a custom user interface, and modify it instantly as needed. The concept is based on HTML and Javascript, and so the central control station can run on a wide range of platforms, including PCs and smartphones.



The ElektorBus browser running on an Android smartphone.

By Jens Nickel (Elektor Germany Editorial)

Home automation and similar measurement and control applications need a central control unit with a display, allowing the user to view readings and adjust settings. The unit might take the form of a PC, programmed, for example, in Visual Basic. Alternatively a smartphone or tablet computer, running the Android open-source operating system, makes an ideal controller. In this case the application code has to be developed in the Java programming language within the powerful Android framework, which presents a steep learning curve to the beginner.

Of course, if you are already familiar with Java, the Android framework and the various development tools, there is nothing to stop you using them to implement the ElektorBus protocol and your own application. If the code for the protocol and the application proper are mixed within the program (as is the case in the demonstration software that we have given previously), then modifications and extensions can be hard to implement. And if you switch from one platform to another, the code must be rewritten from scratch.

Hence we would ideally like to have a library which

- implements the ElektorBus protocol, freeing the developer to concentrate on the application proper;
- provides a clear separation between the application code and protocol code;
- makes it easy for an electronics engineer to design and program a user interface; and
- is platform-independent, so that the same application can run equally well on a PC and on a smartphone.

Sounds like a tall order? Let's see...

### HTML spoken here

Before we look at how the library is used we will take a step back and describe how

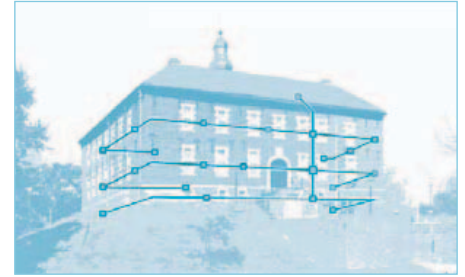
### Elektor Products and Support

- Experimental nodes: printed circuit board 110258-1 or set of three boards # 110258-1C3
- USB-to-RS485 converter (ready built and tested): # 110258-91

- Free software download (microcontroller firmware plus PC software)

All products and downloads are available via the web pages accompanying this article: <http://www.elektor.com/110517>





the whole set-up works. At first the concept might seem overcomplicated and elaborate. However, the advantages of our chosen approach over more conventional programming methods turn out in practice to be significant, and the ideas can be used in other Elektor projects that require control from a PC. Also, a similar approach is adopted with success in modern software development, for example of mobile ‘apps’. And so we recommend even beginners to work through the following description.

First, platform-independence: we achieve this by writing the application code itself, including the user interface, in HTML and Javascript. This combination is astonishingly versatile, allowing our user interfaces to run within a browser on Windows PCs, Macs, Linux machines and all kinds of mobile device. A further benefit is that HTML pages can be transmitted over the internet, which opens up a world of remote control possibilities. HTML’s star is decidedly in the ascendant: the new HTML5 standard brings in new features such as local database storage, 3D graphics and much more.

### A dedicated browser

The HTML front-end interface to our bus will run in a normal browser such as Firefox or Internet Explorer, but for security reasons these browsers are very limited (compared to fully-fledged applications) in what they allow an HTML page to do. For example, a normal browser cannot receive or send data over the serial port of the PC on which it is running. We therefore need a dedicated browser, still capable of displaying HTML pages, but with the extra features we need for ElektorBus applications. Since the ‘ElektorBus browser’ accesses USB and other device functions, it must be specially built for each platform. Fortunately this does not present any great obstacle, as normally the source code does not need to be changed for each new application. For simplicity we make the ElektorBus browser available as an executable file for various platforms: in the case of a PC, as an ‘exe’ file. When switching to a different machine, you simply need to install the appropriate version of the ElektorBus browser, put the HTML/Javascript files in the right directory, and off you go!

The screenshot in **Figure 1** shows the first version of the ElektorBus browser. As in a normal browser, the window that displays the application’s HTML occupies the majority of the display. The HTML and Javascript code form the core of the application, wrapped within the browser which itself is written in a more conventional programming language such as Visual Basic .NET or Java (see **Figure 2**). We can think of the ElektorBus browser the ‘host’ in our system.

### Protocol library

The host receives an ElektorBus message over the serial port of the device on which it is running. The message is then processed according to the protocol we have described previously, and the payload is passed on to the application program (the part written in HTML and Javascript). When the user presses a button in the HTML user interface, the Javascript code generates a corresponding payload for the message to be sent: an example of this is the setting of a threshold value in a sensor node as described in the previous installment. The message is then passed out to the ElektorBus browser, which is responsible for actually transmitting it.

In principle it would be possible to implement all three bus protocols (the ‘Elektor Message Protocol’, ‘Hybrid Mode’ (which is optional) and the ‘Application Protocol’) within the host. On the other hand, it would be possible to make the host transparent, passing the 16 raw bytes in a received message packet directly through to the Javascript code, where the details of the protocol could be implemented. We choose a middle road: the simple Elektor Message Protocol and the rather timing-sensitive Hybrid Mode and scheduler are implemented within the host, while the Application Protocol, which requires rather more code and which some readers will perhaps want to extend, is implemented with the help of a small Javascript library: see **Figure 3**.

In more detail, the process runs as follows. The host receives the sixteen bytes of the message sent over the bus using the start byte synchronization system described in [1]. The message is ‘unpacked’ into a



Figure 1. Screenshot of the first version of the ElektorBus browser with example application running.

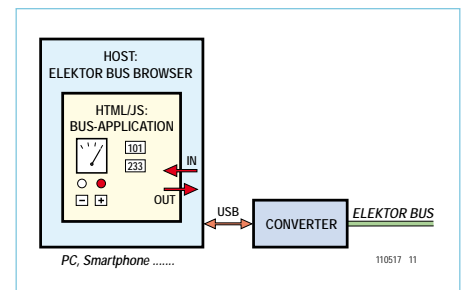


Figure 2. The platform-dependent host is only responsible for the lowest-level functions such as driving the serial port and synchronization. The application proper is written in HTML and Javascript.

data structure that contains (among other things) the transmitter address, the receiver address and the eight payload bytes. These parts are then encoded into a string (called ‘InCommand’) and passed in to the Javascript code (see **Figure 4**). The InCommand string is formatted as plain ASCII (see the text box) which ensures that it will be treated compatibly across different platforms.

The Javascript code accepts the InCommand string and converts it to a simple data structure called ‘Message’, which can

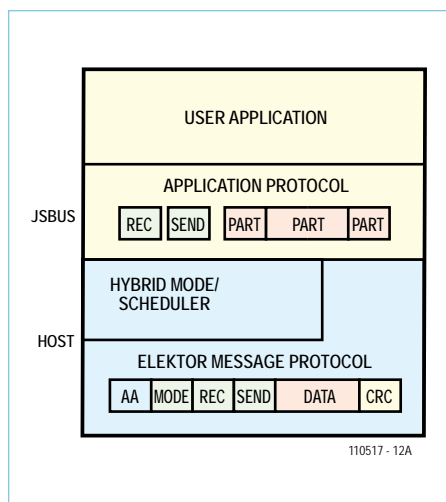


Figure 3. The standard ElektorBus protocol stack (Hybrid Mode and the Scheduler are optional). The host is responsible for the Message Protocol layer, the JSBus Javascript library for the Application Protocol layer.

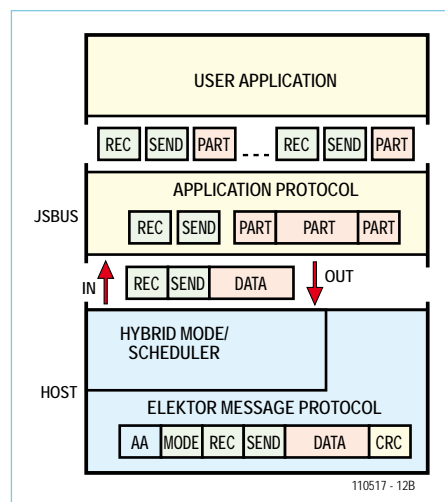


Figure 4. The host and JSBus library communicate with one another by exchanging, in plain text form, the contents of messages that are received or that are to be transmitted.

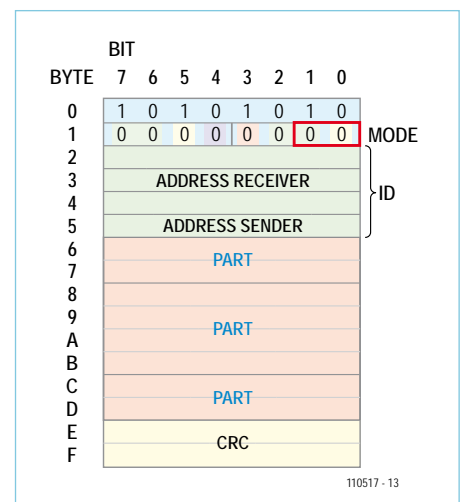


Figure 5. The Application Protocol allows several numeric values or other items of information such as alarm thresholds to be conveyed in one message. The diagram shows an example of three of these so-called 'parts': two consisting of two bytes and one consisting of four bytes.

then be further decoded. In accordance with the Application Protocol the message is divided into so-called 'Parts', which correspond to individual units of information: see the text box 'Messages and Parts'. One unit might for example comprise a single transmitted two-byte value (from -1023 to +1023), or be a message such as 'read-

ing below threshold on sensor 2'. Just as one message can convey several such morsels of information (see **Figure 5**), so the Javascript library can create an array of several parts from a message. The received parts are then passed to a Javascript function into which the developer can put code particular to the application. For example, this applica-

tion code could copy a numerical value contained within a part into an HTML text box.

## Command types

When the application wants to transmit data, the process operates in reverse. A click on an HTML button calls a Javascript function specially written for the application,

## InCommand and OutCommand

The ElektorBus application, written in Javascript, and the ElektorBus browser, written (for example) in Visual Basic .NET, communicate with one another using simple text strings. The JSON syntax is used to encode the necessary information in a data structure within the string to be passed outwards from the Javascript application to the host or inwards from host to Javascript application. The data structures for InCommand and OutCommand are very similar.

### OutCommand:

Command	command type ('Send', 'Url' or 'Scheduler')
add	Send
addurl	Url
addscheduler	Scheduler
cancel	Send
cancelurl	Url
cancelscheduler	Scheduler
clear	Send
clearurl	Url
clearscheduler	Scheduler
count	Send
counturl	Url
countscheduler	Scheduler
delete	Send
deleteurl	Url
deletescheduler	Scheduler
edit	Send
editurl	Url
editscheduler	Scheduler
find	Send
findurl	Url
findscheduler	Scheduler
get	Send
geturl	Url
getscheduler	Scheduler
list	Send
listurl	Url
listscheduler	Scheduler
move	Send
moveurl	Url
movescheduler	Scheduler
put	Send
puturl	Url
putscheduler	Scheduler
remove	Send
removeurl	Url
removescheduler	Scheduler
reset	Send
reseturl	Url
resetscheduler	Scheduler
set	Send
seturl	Url
setscheduler	Scheduler
test	Send
testurl	Url
testscheduler	Scheduler
update	Send
updateurl	Url
updatescheduler	Scheduler

Url      file name for HTML page to be loaded (only for 'Url' commands)

Options reserved for future use

Mode mode byte for the message to be sent (needed as part of the acknowledge mechanism)

Receiver receiver address

Sender    transmitter address

**Data** array of eight data bytes, or addresses of up to eight scheduled nodes

## InCommand:

<i>Command</i>	<i>command type ('Rec' or 'Status')</i>
----------------	---

*Mode* mode byte of the received message (status 2 = OK; -1 = error)

Valid checksum OK? (not yet implemented)

Receiver receiver address

Sender transmitter address

**Data**      array of eight data bytes

In JSON syntax an InCommand appears as in following example:

```
{ "Command": "Rec", "Mode": 0, "Valid": 0, "Sender": 2, "Receiver": 10, "Data": [0, 0, 64, 1, 0, 0, 0, 0] }
```

In the first version of the ElektorBus browser the In- and OutCommands are displayed to aid debugging; see the bottom of Figure 1.

which in turn generates one or more parts. Parts which are ultimately destined for the same receiver can be encoded into a single message. The message object is then transferred from the Javascript code out to the ElektorBus browser as an 'OutCommand', again in the form of plain text. The browser then generates a set of sixteen bytes and sends them out over the bus. Once this is done it returns a success message to the Javascript code. This message also takes the form of an InCommand, this time of type 'Status'.

There are also other types of OutCommand (see the text box) used in various ways to allow the HTML and Javascript to control the actions of the host. The OutCommand 'Url' causes the host to load a new HTML page. This makes it possible to construct an application over several different pages, selected, for example, using some kind of menu. The OutCommand 'Scheduler' switches the scheduler on and off. In this case the data array contains a list of addresses of up to eight nodes that are to be polled.

### Example application

The easiest way to see how the Javascript library and the ElektorBus browser fit together is through an example. We will use exactly the same hardware as we did in the previous installment in this series, with just minor changes to the firmware running on the two nodes: the BASCOM file is available at [2]. **Figure 6** shows once more the hardware involved: node 2, equipped with a light-dependent resistor, continuously sends readings to the master node. The master can instruct the sensor node to change its units of measurement and to set a lower threshold value. If the reading goes below this threshold value the sensor produces an alarm message. In response to this the master sends a message to node 1 to instruct it to pull in its attached relay, and acknowledges the alarm message from the sensor.

A first version of the ElektorBus browser to run on a PC can be downloaded from the web pages accompanying this article [2], along with the Javascript library 'JSBus.txt'

## Messages and Parts

The Javascript library works internally with two data structures to describe messages and parts (items of payload information such as two-byte values, alarm reports, quantity settings and so on) that are being transmitted and received.

The **Message** object basically consists of the familiar components of an ElektorBus message.

**Mode** mode byte

**Receiver** receiver address

**Sender** transmitter address

**Data** array of eight data bytes

**Valid** checksum OK? (not yet implemented)

Within the eight data bytes we can convey up to four parts in accordance with the Application Protocol. Each **Part** is characterised by the following properties.

**Valid** check sum OK? (not yet implemented)

**Sender** transmitter address

**Receiver** receiver address

**Channel** channel number

**Setflag** desired setting or current value?  
**Ackflag** acknowledge message or original message (application-level flag)

**Mode** message's mode byte (with message-level acknowledge flags)

**Parttype** type of part, with the following constant values defined: PARTTYPE\_VALUE2, PARTTYPE\_VALUE4, PARTTYPE\_VALUEFLOAT, PARTTYPE\_LIMIT, PARTTYPE\_SCALE

**Numvalue** numerical data value (for example from -1023 to 1023 in the case of PARTTYPE\_VALUE2)

**Limit** 0 = value between thresholds; 1 = below lower threshold; 2 = above upper threshold

**Quantity** physical quantity (from 0 to 127: see [3])

**Unit** unit of measurement (from 0 to 3: see [3])

**Scale** power of ten scaling (from -15 to +15)

**Interval** reserved

**Preset** reserved

**Options** reserved

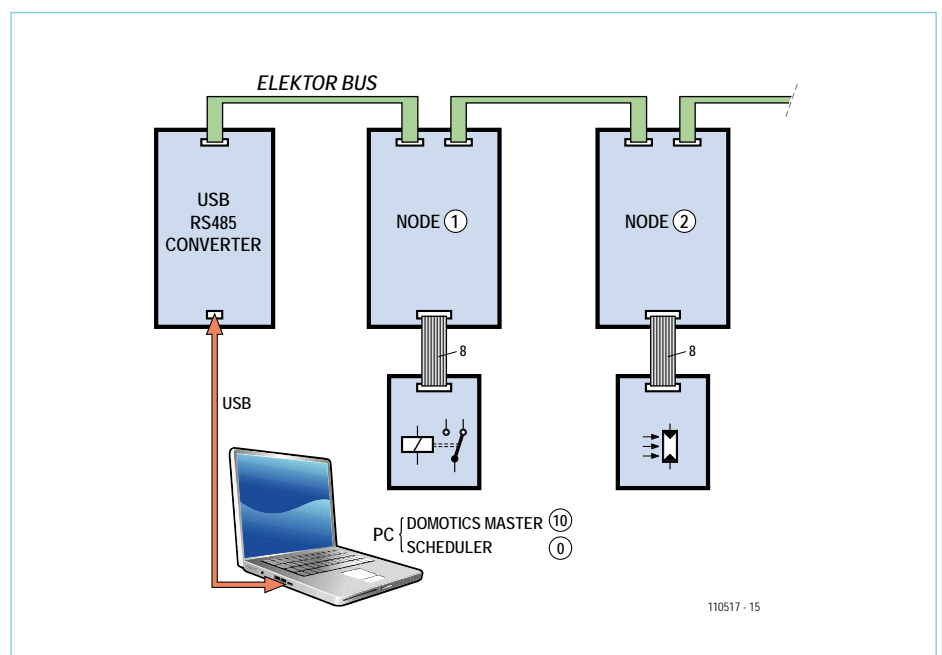


Figure 6. For our example application we re-use the hardware from the previous installment.



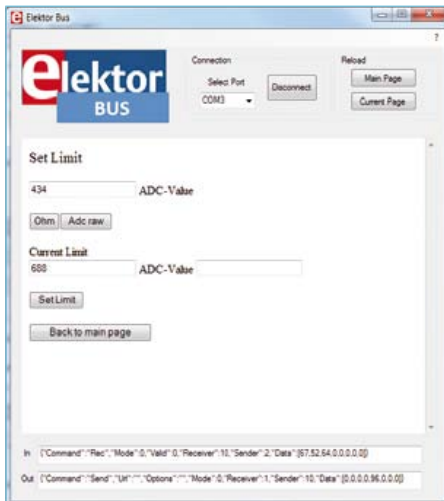


Figure 7. Threshold monitoring: everything is implemented in a platform-independent way in HTML and Javascript.

and the example application, consisting of two files called 'Index.htm' and 'Limit.htm'. You should drag the 'UIBus' directory, which contains these last three files, onto your desktop.

On running the program 'ElektorBus-Browser.exe' the first HTML page (which must be called 'Index.htm') will be displayed. Connect up the serial port and start the scheduler, as described for the PC demonstration software in the previous installment, and readings from the light-dependent resistor should appear in the text box. The buttons below the text box can be used to change the units between raw ADC value (from 0 to 1023) and ohms. The alarm part of the application is displayed separately to demonstrate the use of the 'Url' command: see Figure 7.

### Inward bound

Having completed this first test, we can now open the file 'Index.htm' in a text editor and take a look at its source code (see the Listing). At the top is the reference to the JSBus Javascript library. Below that, enclosed between a second pair of script tags, comes the application-specific Javascript code. The function `ProcessPart(part) { ... }` is mandatory: it is called by JSBus once for

each part received. Inside the function you can specify how received readings, alarm messages and so on should be handled. The properties of the part, such as the transmitting node, the channel and the data value, can be accessed using the syntax `part.property`. Even readers not familiar with the C-like syntax of Javascript will be able to recognize from the listing that the code checks from which transmitter node each part originates. Channel 1 carries the current LED status (0 for off and 1 for on). If a received part belongs to channel 1, the code will set the status of a radio button in the HTML form accordingly. This is done using the function `RadioButtonSetvalue()` which is included in the JSBus library. The most important functions in the library are described in the text box. This function expects as parameters the ID of the HTML radio button whose status is to be set and a numeric value of 0 or 1, corresponding to 'off' and 'on' respectively. Since in this case the value of `part.Numvalue` will itself be 0 or 1, we can include it directly in the arguments to the function. We have chosen to use the names 'LED1' and 'LED2' as IDs for the two radio buttons on the page. This has the lucky consequence that we can construct the first parameter to the function by simply concatenating the string 'LED' and the transmitter node address.

The next function,

`SetSensorScale(quantity)` is called when one of the HTML buttons is clicked on. In this case a part needs to be sent to the light sensor node to tell it to send subsequent readings either as raw ADC results or in ohms. The code has a standard layout: first the line `var parts = InitParts();` initializes an empty array of parts. The next line is the critical one: it is where a new part is added to the end of an existing parts array. The job is simplified by the library function `SetScale()`, which expects as parameters the current parts array to which the new part is to be appended, the transmitter and receiver addresses, the channel and mode byte, and finally the new values to be used for the quantity, units, and power-of-ten scaling. The return value is the extended parts array. Multiple calls to this and other Javascript library functions (see

the text box) can be used to add further parts to the array.

The mode byte comes in useful when we want to send a flagged acknowledge message (see the text box 'A new bit') to a node that transmits in the unregulated free bus phase. In this example node 1 reports the status of its own test LED when the test button is pressed. However, the code we have seen does not include the transmission of the corresponding acknowledge message back from the master. In fact, this job is carried out automatically by the Javascript library: see the function `ProcessReceivedParts(parts)` in `JSBus.txt`.

### Outward bound

The function `SendParts(parts, overrideQueue);` causes the assembled parts (in our case there is just one of them) to be encoded into a message and then transmitted. The library function can automatically assemble up to four parts stored in the part array with the same transmitter and receiver addresses into a single message. Parts that cannot be sent in a single message are encoded into a series of messages that are sent one after the other. All of this, however, is transparent to the application developer. The library stores the messages in a queue. When the Javascript code is informed by the host that the first message has been sent, then next one in sequence is brought to the head of the queue. Now we can see the point of the second parameter `overrideQueue`: a second call to `SendParts` can either overwrite any messages still awaiting transmission or can fail if it discovers that the queue is not empty. In the example file `Limit.htm` we use both of these alternatives when setting and resetting the threshold alarm, as in each case the master must send two messages.

The next two lines of the Javascript code change the text in our HTML form. For this to be possible the text must be given an ID, in this case 'unit'. Depending on the value of the variable `quantity` the text is modified to read either 'Ohm' or 'ADC value'. The library includes constant definitions for the commonest quantities: for example,

**Listing: The file 'Index.htm'**

```

<SCRIPT src='JSBus.txt' Language='javascript' ></SCRIPT>

<SCRIPT Language='javascript' >

function ProcessPart(part)
{
    if (((part.Sender == 1)|| (part.Sender == 2)) && (part.Parttype == PARTTYPE_VALUE2))
    {
        if (part.Channel == 1) {RadioButtonSetvalue('LED' + part.Sender, part.Numvalue);};
    }

    if ((part.Sender == 2) && (part.Parttype == PARTTYPE_VALUE2))
    {
        if (part.Channel == 0) {TextboxSetvalue('ADC', part.Numvalue);};
    }
}

function SetSensorScale(quantity)
{
    var parts = InitParts();
    parts = SetScale(parts, 10, 2, 0, 0, quantity, 0, 0);
    SendParts(parts, true);

    if (quantity==RESISTANCE) {TextSetvalue('unit','Ohm');};
    if (quantity==RAWVALUE) {TextSetvalue('unit','ADC-Value');};
}

</SCRIPT>

<FORM Name='Bus'>

<STYLE type='text/css'>#head {font-size:20}</STYLE>

<DIV ID='head' >ElektorBusBrowser </DIV> <br/>

Scheduler
<BUTTON Type='button' onclick='javascript:SetScheduler(SCHEDULER_ON,2,10,0,0,0,0,0,0)' >on</BUTTON>
<BUTTON Type='button' onclick='javascript:SetScheduler(SCHEDULER_OFF,2,10,0,0,0,0,0,0)' >off</BUTTON>
<br/><br/><br/>

LED Node 1
<INPUT Type='radio' ID='LED1' Name='LED1' Value='LED1' />

LED Node 2
<INPUT Type='radio' ID='LED2' Name='LED2' Value='LED2' /> <br/><br/>

<INPUT Type='text' ID='ADC' Value='' /> <SPAN ID='unit' >ADC-Value</SPAN> <br/>

<BUTTON Type='button' onclick='javascript:SetSensorScale(RESISTANCE)'>Ohm</BUTTON>
<BUTTON Type='button' onclick='javascript:SetSensorScale(RAWVALUE)'>Adc raw</BUTTON> <br/><br/>

<BUTTON Type='button' onclick='javascript:GotoUrl("Limit")'>Set-Limit-Page</BUTTON> <br/><br/>

</FORM>

```

## Main functions in the JSBus Javascript library

```
function InitParts()
```

Returns an empty array of parts. Called as follows: `var parts = Initparts();`

```
function SetLimit(parts, sender, receiver, channel, mode, limit, numvalue)
```

```
function SetScale(parts, sender, receiver, channel, mode, quantity, unit, scale)
```

```
function SetValue(parts, sender, receiver, channel, mode, setvalue)
```

These functions append a new part to an existing array `parts`, respectively representing a threshold, a quantity, unit and scaling value, and a set-point for a given sensor or actuator. The return value is the extended array.

```
function SendParts(parts, overrideQueue)
```

Encodes and sends all parts in the array in one or more messages: see the text for more details.

```
function PartText(part)
```

Returns a textual representation of a part, for example for debugging purposes.

```
function RadioButtonSetvalue(id, setvalue)
```

Sets or resets a radio button (`setvalue = 0 or 1`).

```
function TextboxSetvalue(id, setvalue)
```

```
function TextSetvalue(id, setvalue)
```

Sets the text in a text box or text element.

```
function GotoUrl(url)
```

Causes the host to load a new HTML page (`url = file name without trailing '.htm' extension`).

```
function SetScheduler(status, schedulednode1, ... , schedulednode8)
```

Switches the scheduler in the host on or off (`status = SCHEDULER_ON or SCHEDULER_OFF`) and provides the scheduler with a new list of nodes that should be regularly requested to send a message. A zero value terminates the list.

RESISTANCE is defined as the constant 18. BASIC aficionados should beware of the doubled equals sign in comparison operations and pay particular attention to consistency in the use of upper and lower case in Javascript code.

Now we turn to the HTML section. This consists of a series of elements, each introduced by a tag such as '`<DIV>`' or '`<INPUT>`'. The tags of particular interest to us are those enclosing '`<INPUT>`' and '`<BUTTON>`' elements. In the case of INPUT elements we specify the type (as '`radio[button]`' or '`text[box]`') and an ID string. The ID string allows the element to be identified and addressed subsequently in Javascript code. Text that is to be changed dynamically is best enclosed within '`<DIV>`' or '`<SPAN>`' tags. The first of these also puts the text in its own paragraph.

The extra distinguishing 'name' attributes in the radio button element ('LED1' and 'LED2') allow us to control the buttons independently of one another. If we had used the same value for the two 'name' attributes we would only be able to activate one of the buttons at a time, like the interlocking channel selector buttons on an old-style radio.

The 'onclick' attribute of the button links a click on the button to a call to a Javascript function. Clicking on the first of the two buttons switches the scheduler on or off via the function `SetScheduler()` in `JSBus.txt`. The two additional buttons defined towards the bottom of the listing call the application-specific function `SetSensorScale()` with the appropriate parameter (`RESISTANCE` or `RAWVALUE`); and finally the last button causes the browser to load the page `Limit.htm`, using the JSBus library function

`GotoUrl()`. The parameter to this function is the name of the destination page, without the trailing '.htm' extension. Note that in this case the name has to be given in double quotation marks, since single quotation marks have already been used to delimit the value of the 'onclick' attribute. The HTML files should be stored in the directory 'UIBus' on the desktop. Later versions of the Elektor-Bus browser will allow alternative locations to be used.

### Outlook

If you have already had a little experience with HTML and Javascript, the information in this article and in the text box describing the most important functions provided by the JSBus library will be enough to let you produce some impressive applications. Beginners should look first at the 'HTML and Javascript basics' download [2] and then try experimenting to see the effects of



simple changes to the example HTML code. A simple text editor can be used to create HTML files, and they can be opened in your favorite browser by simply double-clicking on them. This will let you see immediately what your application's user interface will look like. And since we will be using this system for the subsequent installments in this series, you will soon also have more example code that you can adapt for your own applications.

Getting to grips with the HTML/Javascript combination will come in handy when, in

a future installment in this series, we use the same idea to control our system from an Android smartphone (see **introductory photograph**). The bridge to the bus is based on a neat little board containing an FTDI Vinculum II chip, which we will describe in the January 2011 issue.

The board can be used for a wide range of applications beyond the ElektorBus, but in each case HTML and Javascript still provide an easy way to make a platform-independent user interface.

(110517)

**What do you think?**  
Feel free to write to us with your opinions and ideas.

### Internet Links

[1] [www.elektor.com/110258](http://www.elektor.com/110258)

[2] [www.elektor.com/110517](http://www.elektor.com/110517)

[3] [www.elektor.com/110428](http://www.elektor.com/110428)

[4] [www.elektor.com/110382](http://www.elektor.com/110382)

[5] [www.w3schools.com](http://www.w3schools.com)

## A new bit

While working on this article I received plenty of advice from Elektor readers who have already embarked upon their own applications and extensions. For example, Jan Dalheimer from Sweden, who at 15 must be one of our youngest fans, has already started on an AVR micro-controller version of an ElektorBus library. As of our copy deadline Jan's code already handles the Message Protocol and Hybrid Mode, but does not yet include the Application Protocol described in the last couple of installments. Jan found the following annoyance in the acknowledge mechanism: it is possible under the Message Protocol to flag a message so that an acknowledgement message is expected by the transmitter (using bit 0 of the mode byte: see [1]). However, at this level an acknowledgement message (which for good reasons contains a copy of the original data payload) cannot be distinguished from the original message, as we only introduced this flag as part of the Application Protocol. This means that this function of the bus system creates an interaction between the two layers of the protocol which makes it harder to implement cleanly in a library.

Jan's proposal was to use bit 1 of the mode byte to distinguish between an acknowledge message and the original message. Flagging acknowledge messages at the message level is a good idea that we are happy to adopt: unfortunately, bit 1 of the mode byte has already been appropriated for other purposes, and so we have to rearrange things slightly.

Bit	1	0
7	<i>no ID bytes, data from byte 2</i>	<i>ID bytes from byte 2</i>
6	<i>bytes 2 and 3 are ID bytes</i>	<i>bytes 2 to 5 are ID bytes</i>
5	<i>no CRC</i>	<i>bytes E and F form a 16-bit CRC</i>
4	<i>last ID byte is a fragment number</i>	<i>all ID bytes are used for addressing</i>
3	<i>next fragment follows immediately</i>	<i>no fragment follows immediately</i>
2	<i>top six address bits give bus segment</i>	<i>no segment address</i>
1	<i>acknowledge message</i>	<i>original message</i>
0	<i>acknowledge message expected</i>	<i>acknowledge message not expected</i>

In Figure 5 we have indicated bits 1 and 0 in red. We have already implemented the new system in the Javascript library: an acknowledgement message is automatically sent out with the mode byte set to 2 if the original message was received with its mode byte set to 1.

We should clearly distinguish the above from the acknowledge mechanism that is used at the level of the Application Protocol: an example of this is alarm reporting by the light sensor (see [3]). Since the node is regularly interrogated the mode byte in the messages it sends is normally zero: no acknowledgement of them is required since collisions cannot occur. Any acknowledgement message must therefore be implemented by the application developer. To distinguish it from the original message we use the Ack flag in the data payload, first introduced in [4]. An important 'part' within a 'message', such as an alarm report, should have its Ackflag set to true and re-encoded into a message to be sent back to the sensor. To this end the Javascript library includes the function `AddAckPart (...)`. An example of how to use this function can be found in the file 'Limit.htm'.

The "Must Have" reference for  
loudspeaker engineering professionals.

# Home, Car, or Home Theater!

Back and better than ever, this 7th edition provides everything you need to become a better speaker designer. If you still have a 3rd, 4th, 5th or even the 6th edition of the *Loudspeaker Design Cookbook*, you are missing out on a tremendous amount of new and important information!

Now including: Klippel analysis of drivers, a chapter on loudspeaker voicing, advice on testing and crossover changes, and so much more! Ships complete with bonus CD containing over 100 additional figures and a full set of loudspeaker design tools.

**A \$99 value!**

**Yours today for just \$39.95.**

Shop for this book, and many other Audio Amateur products, at **[www.cc-webshop.com](http://www.cc-webshop.com)**.

**Bonus CD offer inside includes LDC7 Design Suite software (a \$99 value)**

## Loudspeaker Design Cookbook



**BY Vance Dickason**

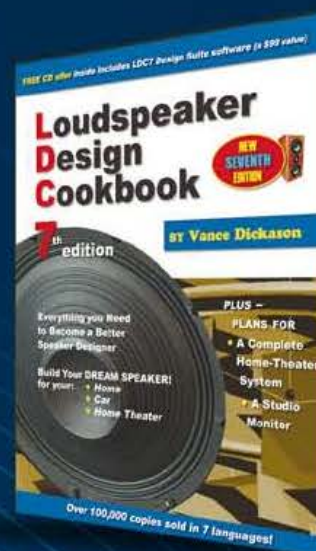
7th edition

Everything you Need to Become a Better Speaker Designer

**Build Your DREAM SPEAKER!**

- Home
- Car
- Home Theater

- PLUS —**
- PLANS FOR**
- A Complete Home-Theater System
  - A Studio Monitor



# Elektor Academy Webinars in partnership with element14

Elektor Academy and element14 have teamed up to bring you a series of exclusive webinars covering blockbuster projects from recent editions of Elektor magazine. Participation in these webinars is COMPLETELY FREE! All you need to do is register at [www.elektor.com/webinars](http://www.elektor.com/webinars).

## Webinar programme:

### **E-Blocks, Twitter and the Sailing Club**

Date: Thursday November 17, 2011

Time: 15:00 GMT (16:00 CET)

Presenters: Ben Rowland and John Dobson (Matrix Multimedia)

E-blocks are small circuit boards containing a block of electronics that you would typically find in an electronic or embedded system. In this webinar Ben and John demonstrate rapid prototyping of an E-Blocks configuration capable of automatically sending Twitter messages to members of a sailing club.

### **Let's Build a Chaos Generator**

Date: Thursday December 15, 2011

Time: 15:00 GMT (16:00 CET)

Presenters: Maarten Ambaum and R. Giles Harrison (Reading University)

Join us in this webinar to look at the making of the Chaos Generator project published in the September and October 2011 editions of Elektor. Get out your opamps, wipe your monitor and glasses and turn up the volume loud!

### **Here comes The Elektor Bus**

Date: Thursday January 19, 2012

Time: 15:00 GMT (16:00 CET)

Presenter: Jens Nickel (Elektor)

Many Elektor readers have actively participated in designing what's now known as the Elektor Bus. Elektor editor Jens not only tells the story of how it all came about, but also delve into protocols, bus conflicts and hardware considerations.

**Now available to view on demand at [www.element14.com](http://www.element14.com):**

### **Platino – an ultra-versatile platform for AVR microcontroller circuits**

Presenter: Clemens Valens (Elektor)

Many microcontroller applications share a common architecture: an LCD, a few pushbuttons and some interface circuitry to talk to the real world. Platino offers a flexible through-hole design for such systems based on the popular AVR microcontrollers from Atmel. Platino supports all 28 and 40 pin AVR devices, several types of LCD and has a flexible pushbutton and/or rotary encoder configuration.



**Places are limited. REGISTER NOW at [www.elektor.com/webinars](http://www.elektor.com/webinars)!**



# Economical Voltage Monitor

By Rolf Blijleven (Netherlands)

Powering stand-alone circuits from solar cells is perfectly possible as long as continuous operation is not required. You can also accumulate the relatively low energy output of solar cells until you have enough to power a circuit for a short while. How do you know when you have enough? You use a voltage monitor. The hitch is that the monitor itself consumes power, but it shouldn't drain the accumulated energy — or at least it should use as little as possible. Even a few microamperes make a difference.

We discussed the principle of energy harvesting in the January 2011 edition. The idea is to operate a circuit entirely from a small source of energy, without using an external power supply or batteries.

In the present case the energy source is a small solar cell (costing 5 pounds or so) with a maximum output of 1 V at 100 mA. This level is only possible on a cloudless day with the cell facing directly towards the sun. On a cloudy day the cell produces only a few hundred millivolts at a few dozen microamperes. Nevertheless, you can do something useful with this small amount of energy — not continuously, but for brief periods a few times a day.

## Design

**Figure 1** shows the block diagram. The output of the solar cell is connected to a voltage multiplier, which charges the storage capacitor  $C_s$ . A number of voltage multiplier designs were discussed in detail in this year's January issue of *Elektor*, so here the multiplier is shown as a functional block. The opamp in the block diagram is a comparator that is powered from the storage capacitor. In practice there are also a few resistors around the comparator. When the plus input reaches the threshold voltage, the output of the comparator goes high (to the level of supply voltage  $V_s$ ) and the transistor is driven into conduction. Current supplied by  $C_t$  and  $R_t$  keeps the transi-

tor conducting for a while after the comparator output changes back to the low level. As a result the load continues to operate as long as energy is available in  $C_s$ . This design is called a 'solar engine'. You can find information on solar engines on the Web (e.g. [1]).

On a cloudy morning the charging current is only some 20 to 30  $\mu\text{A}$ , so the storage capacitor charges very slowly. Nevertheless, we need to continuously check whether enough energy has been stored. We can't use a timer because we don't have a battery. Experiments using a Maximum MAX8282 voltage monitor [2, 3] as a comparator yielded results that were

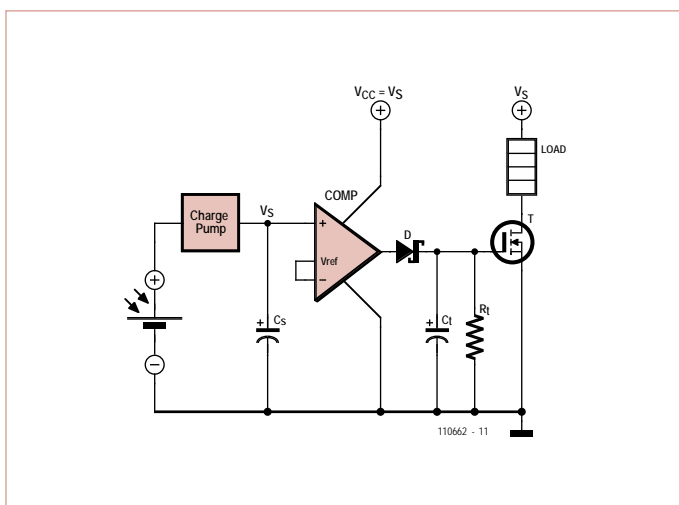


Figure 1. Basic block diagram of the energy harvesting circuit.

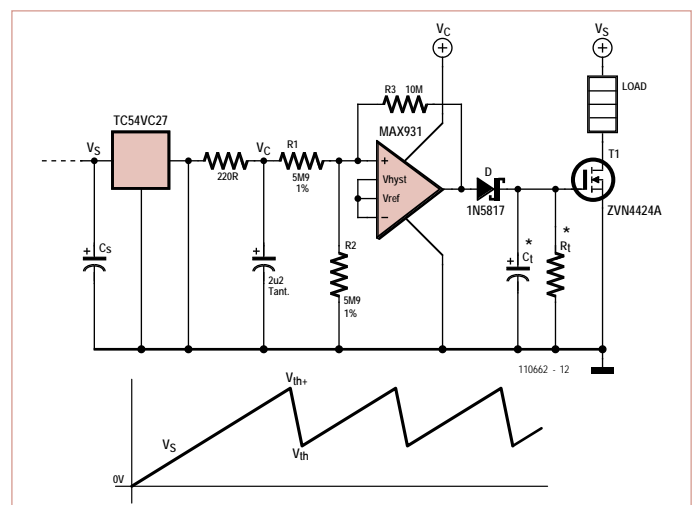


Figure 2. Schematic diagram of the energy-efficient voltage monitor. All semiconductor components are available as SMD and through-hole types.

usable but not entirely satisfactory. This IC together with its configuration resistors draws around  $6\ \mu\text{A}$  – more than 25% of the charging current on a cloudy morning.

### Practical solution

**Figure 2** shows a better solution. The TC54VC27 is a voltage detector. This little three-terminal IC consumes less than  $1\ \mu\text{A}$  as long as the supply voltage is below the built-in threshold voltage (2.7 V), which is the situation during the vast majority of the charging cycle.

Above the threshold level the TC54 passes the input voltage to the output. The transition must occur under controlled conditions, since the output of the TC54 also supplies power to the comparator – a Maxim MAX931 with a quiescent operating current of  $4\ \mu\text{A}$ . The comparator output may go high right away if the supply voltage is applied abruptly, causing the transistor to conduct briefly and pull down  $V_s$ . If this happens the voltage on  $C_s$  will never rise to the desired level, which is not what we want, so we use an RC network ( $220\ \Omega / 2.2\ \mu\text{F}$ ) to prevent this sort of behavior. The MAX931 remains off until the threshold voltage  $V_{th+}$  is reached. This voltage is determined by  $R_1$ ,  $R_2$  and  $R_3$  and is approximately 3.1 V with the specified component values. The difference between this and 2.7 V may appear small, but it is significant. You notice the difference if the load is a motor. Calculating the values of  $R_1$  and  $R_2$  is described in detail in the MAX931 data sheet [4]. At voltages between 2.7 V and the threshold voltage, the monitor IC draws a total of  $5\ \mu\text{A}$ .

TC54 devices are available with various built-in threshold voltages in the range of 1.4 to 7.7 V (see the data sheet [5]). In fact, we could have used one with a threshold voltage closer to our  $V_{th+}$ . You may wonder whether the comparator is really necessary. The answer is yes, because the transistor must have a clearly defined switching point. Here the TC54 starts conducting at around 2.76 V. If the MOSFET were connected directly to the TC54, it would start to conduct a little bit, the voltage on  $C_s$  would drop and the TC54 would stop conducting – and that would get us nowhere.



Figure 3. The chime sounds when enough energy has been harvested. This saves watching the meter all the time.

A feedback resistor between the input and the output of the TC54 is not a solution because it would also draw current from  $C_s$ .

The time constant of  $C_t/R_t$  determines how long the transistor continues to conduct.  $C_s$  should not be discharged any more than is necessary to allow the load to do its job. Suppose the load resistance plus  $R_{dsON}$  of the MOSFET amounts to  $25\ \Omega$  and  $C_s$  consists of three  $4700\text{-}\mu\text{F}$  capacitors for a total capacitance of  $14.1\ \text{mF}$ . This means that  $C_s$  would be fully discharged in 0.35 s. If 0.2 s is sufficient, one-third of the charge can be retained in  $C_s$ , which reduces the time necessary for it to recharge. If we make  $C_t$   $2.2\ \mu\text{F}$ , then  $R_t$  can be  $100\ \text{k}\Omega$ . The value of voltage  $V_s$  versus time is shown in the diagram.

Although the differences may seem small, they are clearly noticeable. The author was not keen to keep his eyes glued on a meter while testing the circuit, so he put together a chime with an electric motor so he could hear how often the energy supply was topped up (or what the weather was

like outside). With the design described here the chime sounds several times a day even during rainy weather, while with less energy-efficient designs it remains still under these conditions.

(110662)

### Internet Links

- [1] <http://library.solarbotics.net/circuits/se.html>
- [2] [www.maxim-ic.com/datasheet/index.mvp/id/1273](http://www.maxim-ic.com/datasheet/index.mvp/id/1273)
- [3] [www.iamwhen.com/archives/53-Herbert-1701-Species-B-Generation-1.html](http://www.iamwhen.com/archives/53-Herbert-1701-Species-B-Generation-1.html)
- [4] [www.maxim-ic.com/datasheet/index.mvp/id/1219](http://www.maxim-ic.com/datasheet/index.mvp/id/1219)
- [5] <http://ww1.microchip.com/downloads/en/DeviceDoc/21434h.pdf>

# Time Domain Reflectometry

## Beginner's guide to locating shorts and opens in long cables

Sure, an ohmmeter is a handy instrument for checking wires and cables for opens and shorts. But finding an open or short is not easy when the cable is long, or if the fault is hidden inside a wall or under a street. Finding partial opens or shorts, or just bad connections, is sometimes even tougher. That's where TDR comes in useful.



By Peter A. Stark (USA)

TDR stands for two things — the process of Time Domain Reflectometry, or the device that does it, called a Time Domain Reflectometer. There is also OTDR, which is Optical TDR for optical fibers. The basic idea is to send a signal, usually a pulse, into the cable or optical fiber, and then see whether a problem in the cable reflects some or all of that pulse back to the input. The nature of that reflection gives us an idea of what the problem is, and the time it takes for the reflection to get back to the input tells us how far away the problem is located.

TDR test instruments are generally quite expensive, and not found in the typical enthusiast's workshop. Even many commercial labs can't afford one. But if you're willing to make do, you can make useful TDR measurements with just a pulse generator and an oscilloscope. A useful pulse generator can be easily built with little effort, and the oscilloscope need not be expensive. Obviously the better it is, the better your results will be, but even a 15-year-old 10 MHz scope is good enough — that's what we used for the photos in this article.

### Transients and steady-state

The circuit in **Figure 1** is simple — with the switch open, there's no current. But close the switch, and Ohm's Law says that a current of 0.1 ampere (10 volts divided by 100 ohms) flows in the circuit.

But let's make the circuit a bit more interesting: let's make the wires from the switch to the resistor very long — for instance 186,000 miles long. To keep things simple, assume the wire is perfect and has zero resistance. Now what? Ohm's Law doesn't care how long the wire is — it says that  $I = V/R$ , so the current is still 0.1 A. But there is more to the story!

The reason we picked 186,000 miles for the wire is that light travels 186,000 miles per second. If we're at the switch with a telescope, looking at the resistor, it takes the light 1 second to reach us. What we see is not what's there right now, but what was there 1 second ago. When we close the switch, we can't even be sure that that resistor is still there! Suppose there was some practical joker out there who swapped out that resistor for some other value, or even disconnected it, just before we closed the switch? We won't know until a second later.

So how can we know (or better yet, how can the battery know) that it's supposed to send exactly one-tenth of an ampere down the line to the resistor when you close the switch?

The answer is that it doesn't. The current that starts down the wire depends on the wire, not the load. Even though we're assuming that the wire has no resistance, it does still have some inductance (even though it isn't wound into a coil) and some capacitance between the two wires. This inductance and capacitance limits the inrush current to some value. In our case, it will take that current somewhere over one second to get to the resistor (because current in wires travels slower than the speed of light), but if the cable happened to be infinitely long, that current would continue going in forever.

Knowing the voltage and current, we could calculate an equivalent resistance which would cause the same current to flow. That value is what we call the *characteristic impedance* of the cable. The symbol for that is  $Z_0$ . (We call it an impedance, even though it behaves like a resistance, just to be more general.)

For instance, the coaxial cable used for many antennas has a characteristic imped-



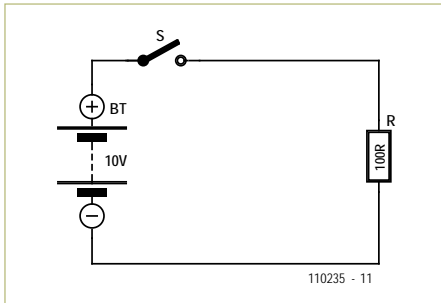


Figure 1. A simple DC circuit ... or is it?

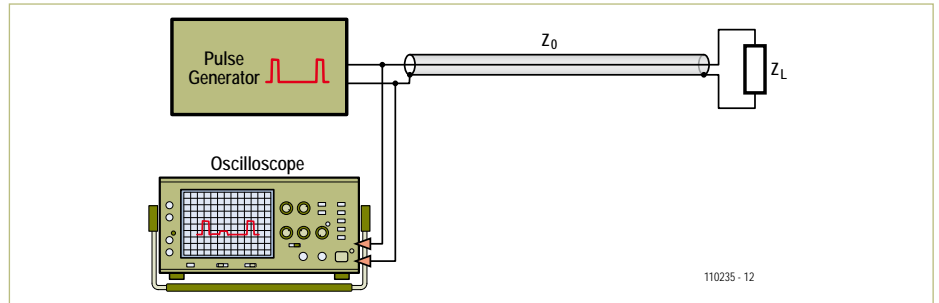


Figure 2. Circuit for sending pulses into a cable.

ance  $Z_0$  of  $50\ \Omega$ , so if we connected an infinitely long length of that cable to our 10-V battery, the current going into the cable (still using Ohm's Law) would be

$$I = V / Z_0 = 10 / 50 = 0.2\ \text{A or } 200\ \text{mA}.$$

OK, let's assume we're using something like that coax (but made of perfect wire with zero resistance) between the battery and our 100- $\Omega$  resistor in Figure 1. The battery has no way of knowing how long the cable really is, so it starts pumping in 200 mA just in case. In a bit over one second, that 10 volts and 200 milliamperes hits the resistor. At this point, the 100- $\Omega$  resistor raises an objection: "You've got either too much current or too little voltage! If you insist on 10 V, then I want 0.1 A or 100 mA. But if you give me 200 mA, then the correct voltage is 20 V ( $V$  equals  $I$  times  $R$ ). Fix it!" The cable then comes back with, "Let's compromise. Let's raise the voltage a bit and drop the current by the same percentage." So they do a quick calculation and decide to change by 1/3: they will raise the voltage by 1/3 to 13.3 V, and drop the current by the same 1/3 to 133 mA. Now the resistor is happy because

$$I = V / R = 13.3\ \text{V} / 100\ \Omega = 0.133\ \text{A}.$$

which for sure is 133 mA.

But now there's a new problem — the voltage and current at the right end of the cable is different from the left end. So the new voltage and current start traveling back toward the battery.

Let's just consider the voltage: 10 V is traveling left-to-right, but one-third of that

(3.3 V) is being reflected back from right to left. About a second later, it gets back to the battery, but now the battery says, "No way, %&\* \*&! I want 10 V!" So it forces the voltage back to 10 V, which also happens to reduce the current. Another second or so later, the resistor ...

Well, you see what's happening. The battery keeps dropping the voltage, the resistor keeps trying to raise it, and all this time the current is slowly dropping. Eventually things will settle down to 10 V and 0.1 A you would expect, but it takes a while. That initial set of compromises is called a *transient*; the eventual settling down to a steady value is called *steady state*.

This interplay between voltage and current happens no matter how long the cable is, but if the cable is very short, it happens very fast. As a rough value, electric signals in a cable travel about eight inches per nanosecond (which is a billionth of a second), so if the wire from the battery to the resistor is only a foot or two, then the whole thing is over in a few nanoseconds.

Before you blink an eye, the voltage and current have settled down to their steady-state values, and nobody ever realizes that a transient ever occurred. You'd need an extremely fast storage oscilloscope to even see it happen. On the other hand, if you have a long enough cable, then even a moderately decent scope can see the transient. That's what TDR or *Time Domain Reflectometry* is all about.

### The Reflection Coefficient

How did the cable and resistor decide that 1/3 or 33% would be a good value to

compromise by? They used the following equation:

$$\text{Reflection coefficient} = (Z_L - Z_0) / (Z_L + Z_0)$$

where  $Z_L$  is the resistance of the load, and  $Z_0$  is the characteristic impedance of the cable. (As before, they are usually resistive, but engineers use  $Z$  for impedance ... just in case.) With our numbers, we have the reflection coefficient

$$= (100 - 50) / (100 + 50) = 50 / 150 \\ = 0.333 = 1/3.$$

This is called the *reflection coefficient* because it tells us what fraction of the outgoing signal gets reflected back to the beginning.

The reflection coefficient can be positive, as here, or negative (if  $Z_0 - Z_L$  is negative) or even zero (if  $Z_L$  equals  $Z_0$ ), so the reflection can be positive, negative, or zero.

### Pulses are better

Suppose that, instead of closing the switch and leaving it closed, we close it for just an instant and then immediately open it again. The 10 V will be just a short pulse traveling right; the reflection of 3.3 V will also be just a short pulse coming back a while later.

Since there is just one pulse going down and one back, we'd need a storage scope to capture it, so let's repeat the pulses at a high rate with a pulse generator, connected as in Figure 2. (If possible, use a 10:1 probe on the scope.) Although the diagram shows a coaxial cable, any kind of cable would do.

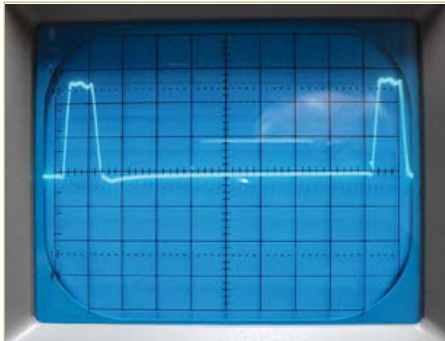


Figure 3. Pulse generator output.

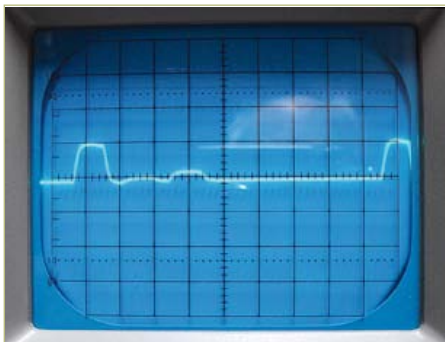


Figure 4. Pulses reflecting from a 100-Ω resistor.

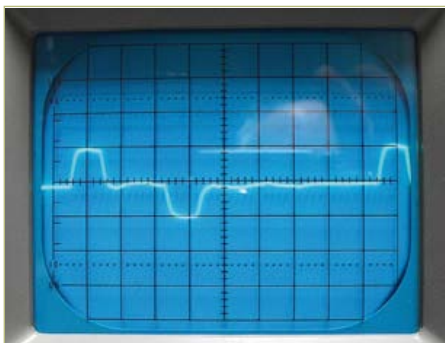


Figure 5. Reflection from a short circuit.

**Figure 3** shows the generator's output when fed directly into a scope and no cable. The scope was set to a vertical sensitivity of 5 volts per division, and a horizontal sweep speed of 0.1 μs per division,

so the pulses are about 0.1 μs wide, and about 0.9 μs apart.

Let's now connect 100 feet of RG-58U 50-Ω coax cable, with the 100-Ω resistor at the far end, which gives us **Figure 4**. We see two things: our original pulses have gotten smaller because the cable's characteristic impedance is loading down the generator, and there is now a small third pulse in the photo.

That new pulse (second from the left in the photo) is the reflection of the first pulse from the open end of the cable. It took the pulse about 150 ns to travel to the far end of the cable, and another 150 ns to return to the scope, so the total delay is about 300 ns or 0.3 μs. So the reflected pulse shows up about 3 divisions to the right of the outgoing pulse. (If the scope screen was wider, we could see a reflected pulse to the right of every outgoing pulse.) Since we calculated the reflection coefficient as 1/3, the reflected pulse is about 1/3 the height of the outgoing pulse. There is some loss in the cable, so the returning pulse will generally be a bit smaller than you'd expect.

Let's look at the delay a bit closer. The scope was set to 0.1 μs per division, and the distance between the two pulses is about 3 divisions, so the round-trip time was about 0.3 μs, which translates to 300 ns. To know whether this is right, we need to know how fast electric signals travel in that particular cable (the speed depends on the cable). Looking up the specs for normal RG-58U coaxial cable, we see a parameter called the *velocity factor* of 0.66. This means that signals in this particular cable travel at 0.66 times the speed of light. Since the speed of light in a vacuum is 186,000 miles per second, this translates to 186,000 times 0.66, or about 123,000 miles per second. We multiply by 5280 to convert miles to feet, which gives us about 648,000,000 feet per second. This works out to about 648 feet per μs, or 0.648 foot (about 7.8 inches) per ns. Therefore the 300 ns delay in Figure 4 works out to 300 times 0.648 feet, or about 194.4 feet. Remember — that's the round-trip length, which is about right for a 100-foot cable. If we could measure

the delay more accurately on the scope, we could use this method to measure the length of the cable quite accurately without ever unrolling it off its spool.

There's a catch, of course — we need a good calibrated scope, and we also need to know the exact value of the velocity factor. But a commercial TDR instrument can still do a fairly accurate job at it.

## What about other loads?

Besides the 100-ohm resistor, there are another three values that are of interest — a short (0 ohms), an open (infinity ohms) and a resistor exactly equal to the cable's characteristic impedance (i.e.,  $Z_L = Z_0$ ). Let's take a look at those three cases:

A. If the end of the cable is shorted,  $Z_L$  is 0, and so the reflection coefficient is

$$(Z_L - Z_0) / (Z_L + Z_0) = (0 - 50) / (0 + 50) = 0 / 50 = -1$$

This means that 100% of the outgoing pulse is reflected, but the voltage is negative so the polarity is reversed. This is shown in **Figure 5**.

B. If the end is open,  $Z_L$  is infinite, and so  $Z_0$  is so tiny in comparison with the huge  $Z_L$  that we might as well forget it:

$$(Z_L - 50) / (Z_L + 50) \approx Z_L / Z_L = +1$$

The reflection coefficient is plus 1, which means that the entire pulse is again reflected, but this time the polarity is the same. This is shown in **Figure 6**.

C. Finally, suppose the resistor at the end of the cable is equal to the characteristic impedance. Now we have

$$(Z_L - Z_0) / (Z_L + Z_0) = (50 - 50) / (50 + 50) = 0 / 100 = 0$$

This time the reflection coefficient is 0, so there is no reflection at all. This case is shown in **Figure 7**, where I put a 51-ohm resistor at the end of the 50-ohm cable. (There is a slight bump after the outgoing pulse, caused by my 100-foot cable actually consisting of two 50-foot lengths coupled

## Homework 4 U

Once you get your pulse generator working, here are some simple experiments you can try. You may need to play with the scope sweep settings to get a steady display:

1. Get 50 or 100 feet of some cable, stretch it out, and ask a friend to connect something to the far end. Practice your TDR techniques with the pulse generator and scope to try to guess what is connected there. The type of cable is not important.
2. See whether it makes much difference whether the cable is stretched straight, or rolled up.
3. Measure the actual length of the cable, see how long it takes a pulse to make the round trip, and calculate the velocity factor.
4. See how short the cable can be before the outgoing pulse and the returning pulse blend into each other so you can no longer see them as separate pulses. If the cable is short enough, you may see the two pulses add together to make a pulse of double the voltage.
5. Take some unknown cable, such as either speaker cable or the zip-cord that is used for table lamp wiring, and measure its characteristic impedance.
6. Connect two different cables end-to-end. If their characteristic impedances are different enough, you should be able to see a reflection from the connection where the two different wires join.
7. Using a resistor equal to  $Z_0$  of the cable, connect it to the end of the cable with very short leads. You should see no reflection. Now connect it to the same cable but with 1- or 2-foot clipleads. You will probably see some small but clear reflections — what's happening? Do the clipleads make a difference?
8. Spend some time thinking about this: Suppose that the signal going into the cable was a sine wave rather than pulses. What would happen?

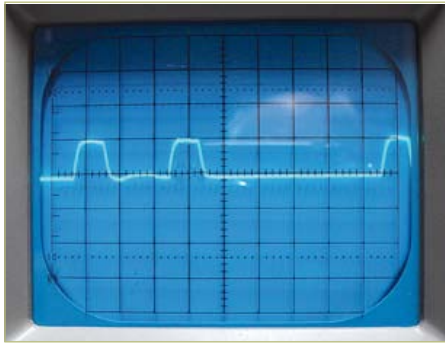


Figure 6. Pulses bouncing off an open circuit are positive.

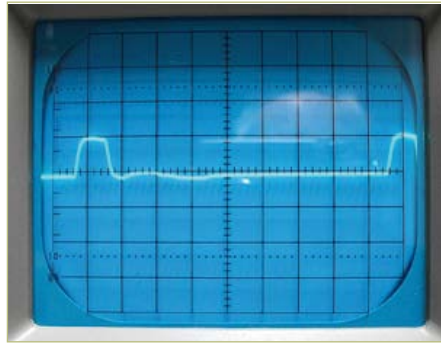


Figure 7. There is no reflection when correctly terminated.

together. This shows a slight reflection from the point where they connect.)

Making the load resistor equal to the characteristic impedance of the cable avoids reflections, and is called *properly terminating the cable*.

So here we have an easy way of finding the characteristic impedance of a cable — try putting different value resistors at the end of the cable until there is no reflection, and then  $Z_0$  is equal to the load resistor value.

### So all we need is a pulse generator

If you happen to have one on your workbench, you're all set. But if not, you can build one for less than five dollars. **Figure 8** shows a simple schematic to work from. Nothing but a 555 timer which provides very short negative pulses at a rate of about 1 MHz (the exact frequency isn't important, and will vary because of tolerances), followed by an amplifier to drive the cable. Six digital inverters in parallel invert the pulse to make it positive, and also provide sufficient output power. The 33- $\Omega$  resistor at the output increases the output resistance to prevent returning pulses from bouncing off the inverters and going back into the cable a second time.

Despite the apparent simplicity of the circuit you need to select our parts carefully. Normal 555 timer ICs don't like to operate at the fast speeds needed, so the best choice is a TLC555 CMOS timer from Texas Instruments. Fortunately, Radio Shack carries it as their part number 276-1718. For

the inverters, a 4069 CMOS hex inverter IC is recommended mainly because it can be operated at 9 V, so the entire generator can be powered from a 9-volt battery. A TTL 7404 or any of the 74xx04 chips would be pin-compatible, but then you'd need to power both the 555 and the inverter from a 5-volt supply, which increases the complexity. The circuit should be simple to construct on a small piece of perfboard, or, if you want, a purpose designed printed circuit board.

(110235)

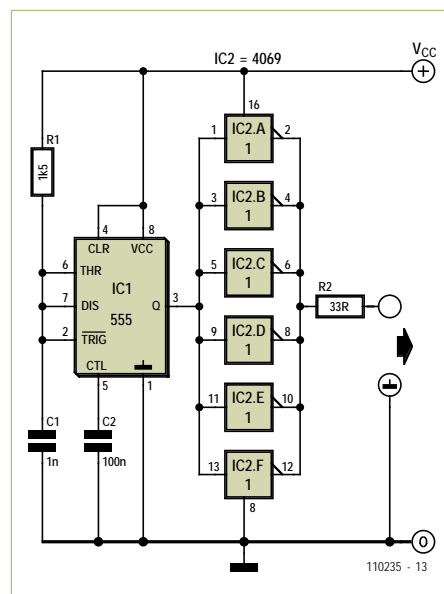


Figure 8. Pulse generator schematic diagram.



# SPICE It Up!

## Circuit simulation with LTspice

By Raymond Vermeulen (Elektor Labs)

Circuit simulation is useful for everyone who works with electronics — from students and enthusiasts to professionals. It lets you quickly verify and modify designs without making a lot of calculations, and it is helpful for dealing with non-ideal component properties that are ignored in formulas, among other things. In this article we provide a concise introduction to the subject for readers who are not already familiar with SPICE-based simulators.

We selected LTspice for this introductory description. LTspice is a SPICE-based simulation program from Linear Technology. It is available free of charge (and without registration) and is easy to use. However, it has the drawback that it comes as standard with component libraries containing Linear Technology components, as might be expected. Components from other manufacturers are not included, but they can be added by the user. The program is highly suitable for simulating circuits built with Linear Technology products, and a large number of simulation examples for this purpose are available on the Linear Technology website. For increased functionality and more libraries, you can of course choose a commercial program such as Micro-Cap or Orcad Pspice, but these programs are not cheap. In any case, getting familiar with LTspice definitely has one advantage: once you know how to use one version of SPICE, you also know how to use the other ones.

### Origin

SPICE was originally developed by the University of Berkeley in the 1970s for the US Department of Defense, for the purpose of analyzing the radiation hardness of circuits. In subsequent years its capa-

bilities were expanded to include the simulation of more complex components and additional simulation functions. In the course of time commercial versions appeared, many of them based on the original SPICE program surrounded by a vendor-specific user interface called a 'shell'.

### Getting started with LTspice

Go to the website <http://www.linear.com>, click 'Design support' at the top of the page, and then click the link under the 'Design Simulation' heading. This takes you to the download page for LTspice. Launch the downloaded file and install the program in the desired location. For Linux users, the program can be run under Wine. This has been tested with Ubuntu 11.04 Natty Narwhale. Launch the program after it has been installed.

Create a new schematic sheet by clicking the icon at the far left:

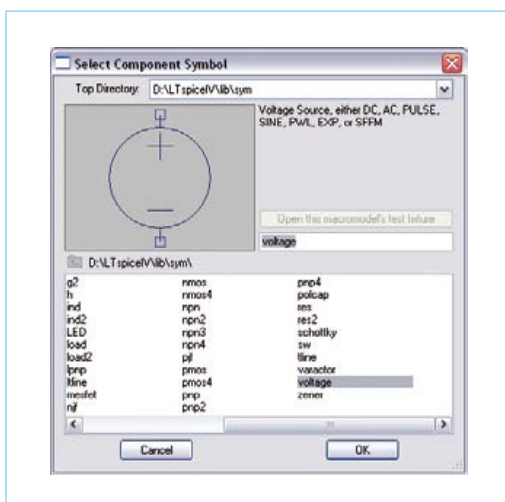


Figure 1. Component selection.

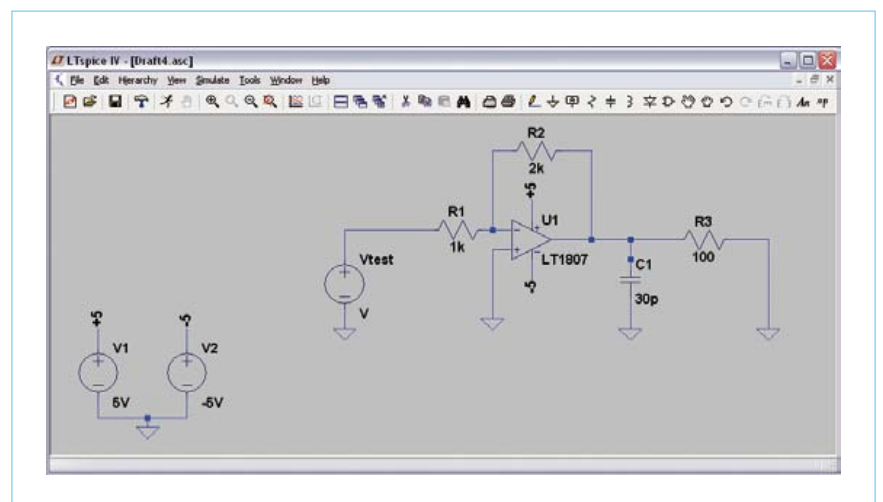


Figure 2. Our first schematic.



Now you can start drawing your first schematic. You can select component symbols from a dialog (**Figure 1**) that appears after you press the 'Component' button:



Standard components such as ground points, resistors, inductors, capacitors and diodes can be selected directly from the menu bar:



You can use the buttons marked below to draft your desired schematic:

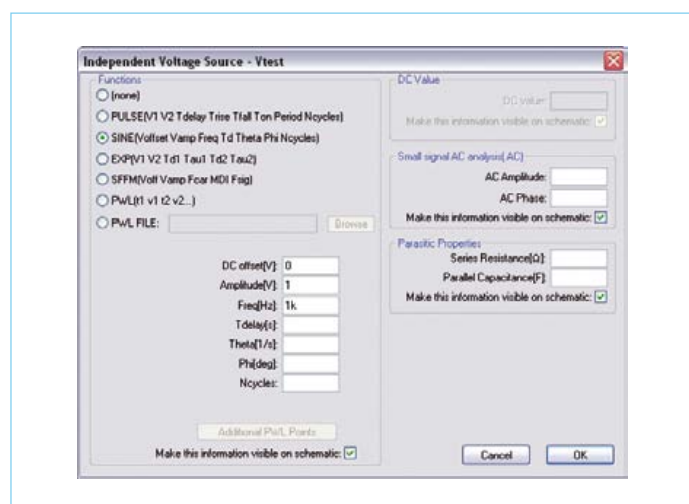


Figure 3. Advanced voltage source settings.

We recommend spending a bit of time experimenting with these buttons in order to get an idea of what they do.

A couple of handy hotkeys are Ctrl-R for rotating a component and Ctrl-E for flipping a component.

**Figure 2** shows the first circuit schematic that we drew. Now we would like to simulate it, but first there are two things that have to be configured: the voltage source 'Vtest' and the simulation parameters.

If you right-click 'Vtest', you will see a dialog where the basic properties of the voltage source can be set (DC voltage and internal resistance). However, in this example we want to simulate the AC

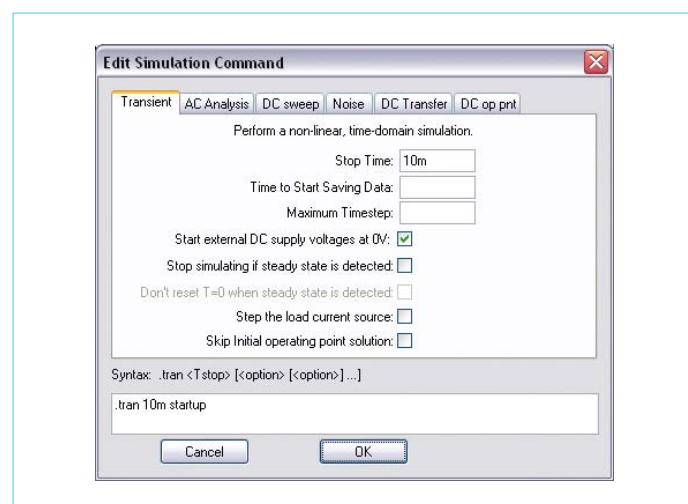


Figure 4. Configuration options for the simulation.

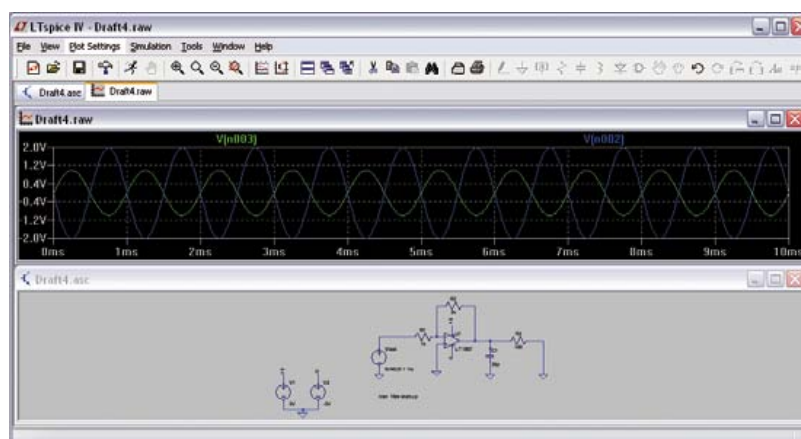


Figure 5. The simulated signals (top) and schematic diagram (bottom).

behavior of the circuit. For this we use a sine-wave test signal with a frequency of 1 kHz, an amplitude of 1 V and an offset of 0 V, which you can configure by clicking 'Advanced'.

This opens the dialog shown in **Figure 3**, where you can configure a

variety of signal waveforms. Here we selected the radio button next to 'SINE' and entered the desired values.

The next task is to simulate the circuit behavior in the time domain. This requires modifying the simulation settings, which we do with 'Edit Simulation Cmd':

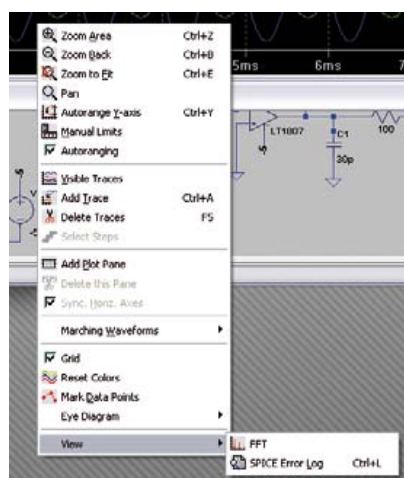


Figure 6. Selecting an FFT analysis.

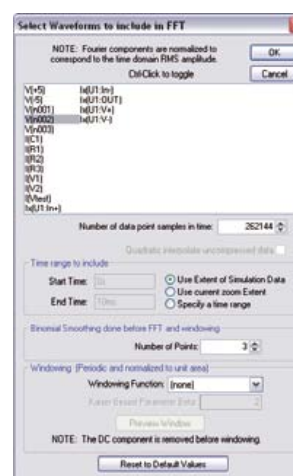


Figure 8. The calculated FFT spectrum.

## Tips

## Standard components

When you're drawing a schematic diagram, you'll notice that when you select some components a whole list of type numbers is displayed in the selection window shown in Figure 1, while with other components no type numbers at all are displayed. For example, if you select 'Opamps' you see a long list of type numbers (primarily Linear Technology devices). However, a type number list is not displayed if you select an NPN transistor. Nevertheless, this list does exist. First place a standard NPN transistor on the drawing area and

then right-click it. Select 'Pick new transistor', and a list of commonly used types from various manufacturers will be displayed. This also works with other standard components.

- More components

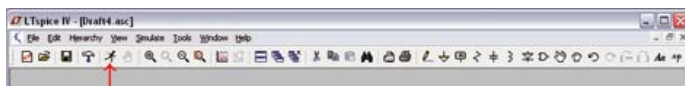
Of course, only a limited number of components are provided with LTspice. Many more components can be found in the Yahoo user group for LTspice: <http://tech.groups.yahoo.com/group/LTspice/>





This opens the window shown in **Figure 4**. We wish to view the output signal in the time domain, so we select the 'Transient' tab. There we can change the simulation parameters. In this case we decided to have the simulation last for 10 ms and have the DC sources start at 0 V. After clicking 'OK', place the box with the configuration settings that appears next to the mouse cursor at any desired location on the schematic. For an analysis in the frequency domain, the next thing we have to do is to select the 'AC analysis' tab and configure its settings.

Now we're ready to run the simulation. Start it by clicking the 'RUN' button.



A new pane opens to display the simulated signal. Now click the points in the schematic diagram where you want to measure the signal (the schematic is still visible below the display pane). In this example we chose the output of the test signal source and the output of the opamp. The signals present at these points then appear in the signal pane (see **Figure 5**).

If you right-click the waveform, a pop-up menu appears (**Figure 6**). You can use it to modify many of the parameters for displaying the simulated signal. Among other things, you can select FFT analysis to display the frequency spectrum. To do this, click 'FFT' under 'View'. This opens a window where you can select a large variety of FFT settings (**Figure 7**). At the top, select the desired circuit node for the FFT analysis. In this case we chose the output of the opamp. The spectrum of the signal at this point is shown in **Figure 8**.

Now it's time for you to try LTspice for yourself in order to learn what the program has to offer. You should also try out other simulation programs. Many of them are available in demo or student versions, which you can use to get a bit of hands-on experience. Although the user interfaces often vary, the most important things that you need to do are always the same: drawing a schematic diagram, configuring one or more test voltage sources, configuring the simulation, and viewing the output.

(110543)

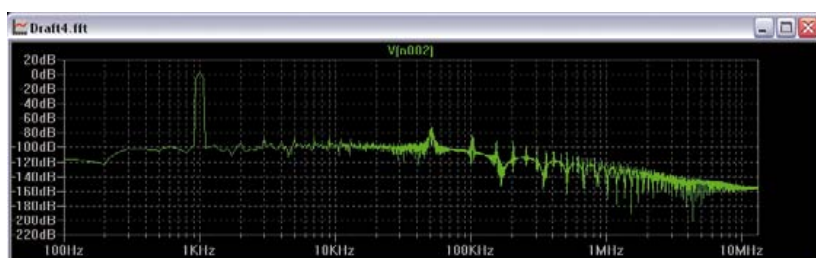


Figure 8. The computed FFT spectrum.

## BEST SCOPE SELECTION & lowest prices!

### 25MHz Scope

Remarkable low cost 25MHz 2 channel plus trigger USB bench scope with 8 inch full color LCD display. Spectrum analysis and autoscale functions.

**PDS5022S \$287**



### iPhone Scope

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

**iMSO-104 \$297.99**

### 50MHz Scope

High-end 50MHz 16Sa/s 2-channel benchscope with 1MSa memory & USB memory port. Comes with a FREE scope carry case!

**DS1052E \$399**



### 60MHz Scope

60MHz 2 channel digital scope with a 500 MSa/s sample rate, 10MSa memory & 8" color TFT-LCD screen.

**SDS6062 \$399**

### World's Smallest

The world's smallest MSO! This DIP-sized 200kHz 2 channel scope includes a spectrum analyzer and arbitrary waveform generator. It measures only 1 x 1.6 inches in size!

**Xprotolab \$49**



### 100MHz MSO

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

**CS328A \$1359**



### Handheld 20MHz

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

**HDS1021M \$299.95**

More selections at:

**www.saelig.com**

Follow us on:



**Saelig.com**  
unique electronics



# Temperature Gradient Meter

## Detect and report the tiniest changes

By Dr Dietmar Schröder (Germany)

Sometimes it is not absolute temperature value that is of interest, but rather any small changes in that value. The circuit described here does not just measure these changes: it reports them both visibly and audibly. Using just a few active components it achieves a temperature resolution of just one ten-thousandth of a degree.



Measuring temperature is a fairly simple task, but doing so precisely takes rather more skill, particularly when it is desired to obtain readings to a resolution of better than one tenth of a degree. The circuit presented here measures temperature to a resolution of one ten-thousandth of a degree, using just four active components and an optional display. The purpose of the circuit is to be able to detect even the smallest change in temperature: the emphasis is much more on extreme sensitivity and resolution than on absolute accuracy.

### Features and possibilities

Without any tedious calibration the circuit measures temperatures to a basic absolute accuracy of about two degrees. It also indicates the temperature gradient, or rate of change of temperature with time. Output

is via a digital display and an analog pointer, as well as being represented in the changing pitch of an acoustic output. Readings are also made available on a serial port for transfer to a PC.

The instrument lends itself to a range of interesting applications. Its sensitivity is so high that its reading shows clear changes in temperature when the sensor is raised or lowered, reflecting the fact that in a room the warm air rises towards the ceiling; if the sensor is positioned by a PC's fan exhaust it is possible to see changes in the load on its CPU or graphics card reflected in the temperature of the expelled air; or alternatively the sensor can be attached to a domestic water pipe to detect when water is being used.

The acoustic output allows 'no hands' monitoring of temperature. For example, the

sensor can be attached to a delicate component in a circuit to give an immediate audible warning when it suddenly starts to dissipate more power.

It is important to insulate the sensor well to keep the display stable. It can, for example, be embedded deeply in an insulating foam such as expanded polystyrene. If the time constant of the temperature gradient is chosen appropriately, it is still possible to measure slow changes in ambient temperature reliably despite the insulation.

The author originally used the circuit to analyze the behavior of the thermostatic valve in the cooling system of a car.

### Construction

The unit essentially consists of an NTC thermistor used as the temperature sen-

*Note. Readers' Projects are reproduced based on information supplied by the author(s) only.*

*The use of Elektor style schematics and other illustrations in this article does not imply the project having passed Elektor Labs for replication to verify claimed operation.*

## Features

Operating range: 0 °C to 40 °C

Absolute accuracy: approximately 2 °C uncalibrated

Resolution: in theory  $4 \times 10^{-5}$  °C, in practice approximately  $3 \times 10^{-4}$  °C; displayed resolution 10<sup>-4</sup> °C

Range: degrees per 2 seconds to degrees per 100 minutes

Power supply: 6 V (four AA cells)

sor, a voltage reference, and A/D converter and a microcontroller driving an LCD panel.

To minimize interference and noise the thermistor must be placed as close as possible to the A/D converter. So that we can operate with the sensor at a relatively large distance from the body of the unit the circuit is partitioned into two circuit boards.

The sensor sticks out from the very narrow sensor board, which also carries the voltage reference and the A/D converter. The sensor is thus only about 10 mm from the A/D converter. The sensor board can be enclosed in heatshrink tubing, the whole thing forming the unit's temperature probe (**Figure 1**).

Readings are conveyed in digital form to the processor board (**Figure 2**), which carries the ATmega88 microcontroller, over a four-wire cable using a serial protocol. The board has provision for connecting a display, a piezo sounder, an analog meter and a serial interface (or USB-to-serial interface cable). The serial interface uses TTL signal levels. Three potentiometers allow adjustment of basic sensitivity, time constant, and piezo sounder volume.

## The circuit

The circuit diagram for the sensor board is shown in **Figure 3**. The 5 V supply is provided from the processor board over the four-way interconnection cable and filtered by the RC-network comprising R100 and C100. From this the TL431 voltage reference diode generates a low-noise reference at 2.5 V, which is again filtered by R102 and C103 and then provided to the MCP3551 analog-to-digital converter. The absolute value of the reference voltage is not particularly critical, as the A/D conversion result depends only on the ratio between the resistance of R103 and that of the thermistor. R103 is chosen so that the characteristic curve of the resulting conversion results is more-or-less linear over the temperature range from 0 °C to 40 °C: see **Figure 4**. Capacitor C104 at the input to the A/D converter acts as an anti-aliasing filter.

The MCP3551 is a delta-sigma converter with a resolution of 22 bits. In the range around 20 °C this corresponds to about 24 000 quantisation steps per degree of temperature, giving a (theoretical, at least) resolution of 40 millionths of a degree. In practice, however, component noise limits performance to about 19 bits of



Figure 1. The sensor board inside the temperature probe supplies a serial digital output signal.

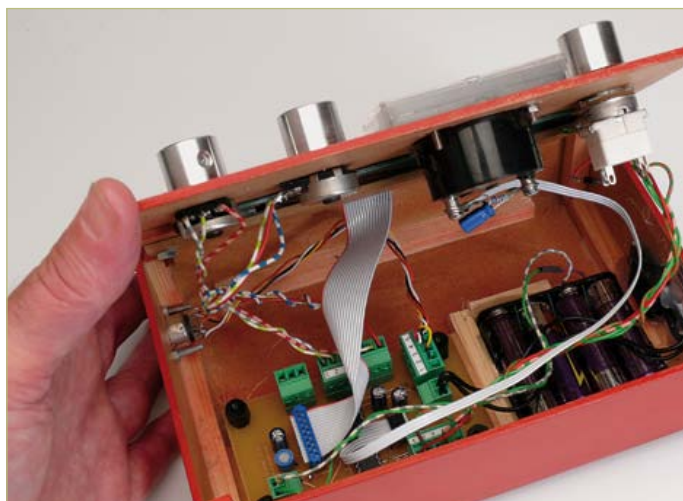


Figure 2. The processor board processes the serial data and drives an analog meter and an LCD panel.

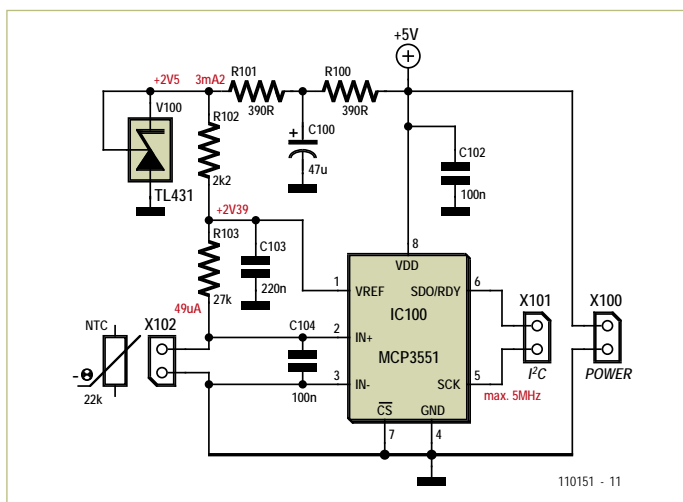


Figure 3. Circuit diagram of the sensor board.



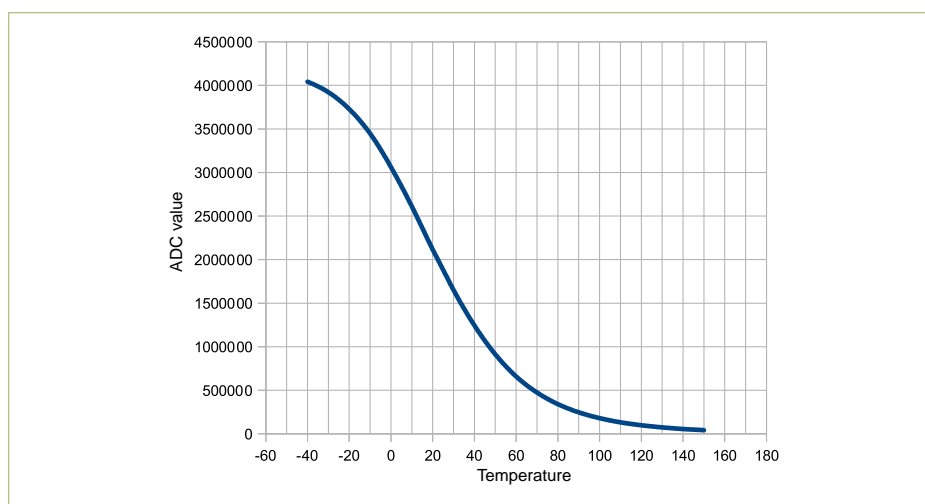


Figure 4. Characteristic curve of the sensor.

effective resolution.

The MCP3551 reads the voltage across the thermistor about 20 000 times per second and digitally filters the results. The chip contains not only a low-pass filter but also a sharp notch filter to suppress 50 Hz and 60 Hz mains-borne hum. A new reading is made available approximately 14 times per second on the serial interface, and these

readings are transferred to the processor board.

**Figure 5** shows the circuit diagram of the processor board. The four-way cable to the sensor board is connected at connector X202. Connectors X201 and X205 are provided for potentiometers P1 (10 k $\Omega$ ) and P2 (also 10 k $\Omega$ ), which are used to adjust the time constants of the digital process-

ing done in the microcontroller's software. A third potentiometer allows adjustment of the volume of the piezo sounder (which should have no internal oscillator circuit, and a capacitance of approximately 8 nF to 20 nF). The connector for this potentiometer, X206, has five pins, allowing the use of a 10 k $\Omega$  potentiometer with a built-in switch which can be used to turn the unit on and off. A fourth potentiometer (P500) is for setting the contrast of the LCD connected to X207. The author used a Wintek WD-C2704M display, which has four rows of 27 characters and an HD44780-compatible controller than can be run in four-bit or eight-bit mode.

The ATmega88 is clocked from its internal 1 MHz oscillator, avoiding the need for a crystal. The device can be programmed over ISP connector X204. The MOSI pin on this connector also serves for connecting an analog meter, which should have a full-scale deflection of 5 V.

The processor board can be powered from four (rechargeable or dry) AA cells. The diode at the power input protects against inadvertent reverse polarity connection and drops the battery voltage by about 0.6 V, thus ensuring that the 5.5 V maximum permissible operating voltage of the processor and A/D converter is not exceeded.

The printed circuit board layouts (**Figure 6** and **Figure 7**) for the two parts of the circuit are available for free download as PDF files and as Protel files at [1].

## Software

The microcontroller firmware can also be downloaded for free from the *Elektor* web-site at [1].

The software periodically reads conversion results from the sensor board and converts them to readings in degrees Celsius. This is then transmitted as ASCII characters over the serial interface (for example as '23,5341°C': note the use of a comma instead of a decimal point, you may want to rework this to give the decimal point). The characteristic curve of the sensor is stored as a table in increments of five degrees, and values in between table entries are linearly interpolated. The resulting overall absolute accuracy is about two degrees over the

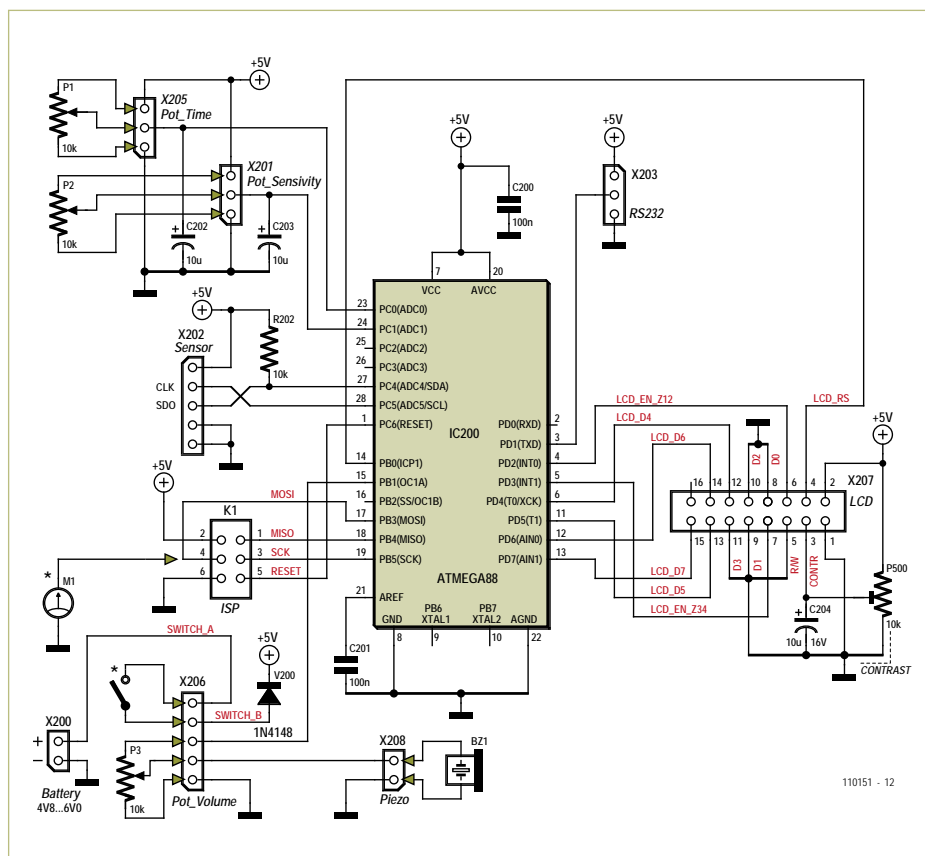


Figure 5. Circuit diagram of the processor board.

range from 0 °C to 40 °C, most of the residual error being attributable to part-to-part variation in the thermistor. If the thermistor is calibrated, an accuracy of better than 0.5 degrees can be achieved.

Temperature values are passed through a chain of two low-pass filters realized in software. The first filters out variations that occur over a short timescale, and the temperature gradient is calculated as the difference between the input and output of the second filter.

The voltage values arising from the settings of potentiometers P1 and P2 are converted

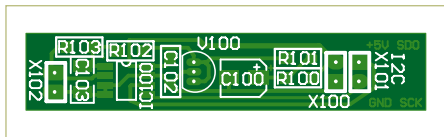


Figure 6. Component mounting plan for the sensor board.

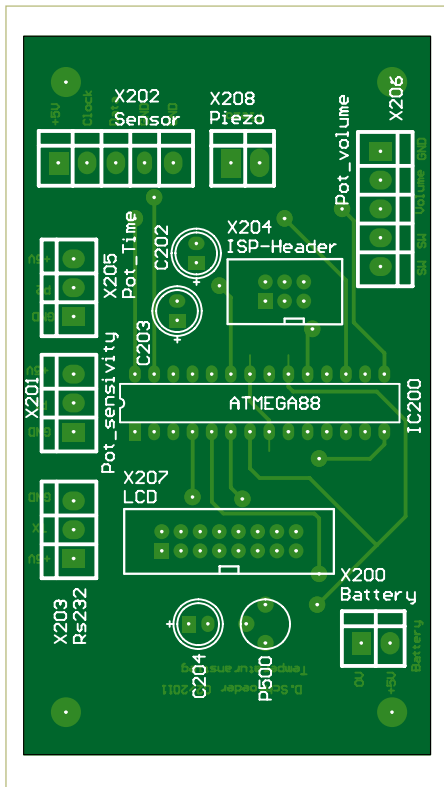


Figure 7. Component mounting plan for the processor board.



Figure 8. Readings are shown on the four-line display.

to digital form using the ATmega88's A/D converter, and from these are derived two time constants,  $t_1$  and  $t_2$ , one for each low-pass filter. The filters operate as simple RC-networks, and so with a voltage step on the input the output settles to within 99 % of its final value within five time constants. The potentiometer settings are squared before being used as time constants in order to allow for easier adjustment over a wide range of time periods. There is no merit in setting the time constant of the second filter shorter than that of the first, and so the first time constant is added to the second. The range available for  $t_1$  is from 0 seconds to 19 minutes and that for  $t_2$  is from 2 seconds to 100 minutes. In general, TP1 and TP2 should differ by a factor of 2. A small amount of hysteresis prevents jitter in the settings.

The temperature gradient can also be indicated on an analog meter. To this end the battery voltage is measured and the mark-space ratio of the PWM output set so that if the gradient is zero the average output voltage is 2.5 V. If the meter has a full-scale deflection of 5 V it will then point exactly to the middle of its scale.

The PWM output is arranged so that the full-scale reading of the meter is  $\pm 0.05$  °C per second.

A timer is used to create a sound whose pitch depends on the absolute value of the temperature gradient. If the temperature is falling, a second timer is used to interrupt the tone periodically, giving a pulsing sound rather than a steady note. If the temperature only varies within the typical noise level of the system, the tone is disabled.

A table is displayed on the connected LCD to show the temperature, the time constant settings and the calculated temperature gradient: see **Figure 8**.

The second line shows the current temperature to four decimal places. The last of these digits is not particularly meaningful as its significance lies below the typical noise

level of the system. If the probe is well insulated the last digit typically varies by about 3 or 4 units.

The third and fourth lines give the temperature values after processing by the low-pass filters, with the time constants as set by the potentiometers shown in the second column. The last column gives the number we are really interested in: the temperature gradient. The value in the third row is the difference between the current reading and the output of the first filter, while the value in the fourth row is the difference between the output of the first filter and that of the second.

### Operation

Initially set P1 and P2 to their extreme left positions, which sets the filter time constants to 0 s and 2 s respectively. It is now a good idea to wait a while for the temperature sensor to warm up: this is not a negligible effect as the thermistor dissipates about 0.2 mW, which leads to a small rise in temperature. Then hold your hand about 10 cm from the sensor for a couple of seconds and watch the meter's pointer move clearly to the right as a result of the thermal radiation. Take your hand away and the pointer will move to the left as the sensor cools.

For most measurements the author employs a value of 5-10 s in the 3rd LCD line (TP1) and 20-120 s in the 4th LCD line (TP2).  
(110151)

### Internet Reference

[1] [www.elektor.com/110151](http://www.elektor.com/110151)

### About the author

Dr Dietmar Schröder works as a software and hardware developer at aixcom PowerSystems GmbH in Germany. He has a personal website (in German) featuring a number of unusual projects at <http://www.zabex.de>.

# Resistive Bolometer

## Don't throw out those traditional lamp bulbs yet!

By Matthieu Denoual, Julien Gasnier and Sylvain Lebargy (France)

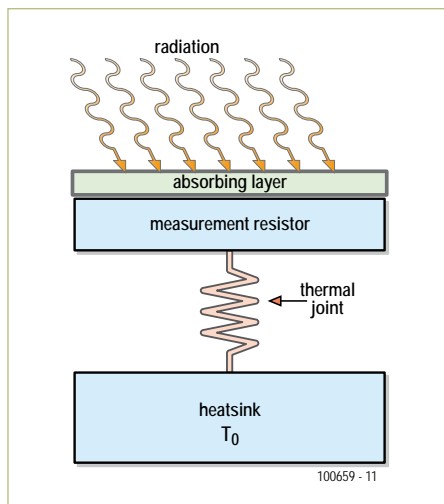


Figure 1. Principle of a bolometer.

If you have converted to modern, energy-saving lights, don't assume your old lamp bulbs are just junk now. You could put them to new use if you build our radiation meter project that works on the principle of the resistive bolometer. Bolometer is an odd-sounding word but the 'bol' element comes from the Greek root word  $\beta\omicron\lambda$ , which means to cast or throw. In the same way that objects can cast a shadow they can also emit radiation, which is what a bolometer detects. The sensor is sensitive to the infrared range and is used particular in thermal imaging systems and in systems used for space observation. In the latter case the sensor is cooled to very low temperatures in order to increase sensitivity and reduce thermal noise. Superconducting materials are also used.

A resistive bolometer consists of two elements: an absorption body (a body of constant temperature or 'thermal reservoir') that 'soaks up' radiation plus a temperature-sensitive precision resistor. The system must be isolated thermally from its surroundings. In **Figure 1** radiation absorbed heats up the body, raising the temperature in the vicinity of the sensing resistor. In a state of thermal equilibrium the temperatures of the absorption body and the sensing resistor match, at around a few degrees above ambient temperature. Any change

in the intensity of radiation will cause the resistance to alter. Since a constant current  $I$  flows, we need measure only the voltage drop across the sensing resistor  $R_m$ . The sensitivity of this kind of sensor can be expressed as follows:

$$S = \eta \times \alpha \times I \times R_m \times R_{th}$$

In this equation  $\eta$  is a material factor in the absorption body,  $\alpha$  is the temperature coefficient of resistance (TCR) and  $R_{th}$  is the thermal resistance that exists between the measurement system thermal reservoir and precision resistor) and the surroundings.

In this project the light bulb acts as a radiation detector according to the radiometer model. The matt black tungsten filament serves simultaneously as the absorption body and sensing resistor. Since the TCR of tungsten is  $0.48\% \text{ K}^{-1}$  we can expect relatively high sensitivity. The inert gas in the bulb, typically nitrogen or an inert gas such as xenon or argon, provides the thermal insulation required and with this a high value of  $R_{th}$ . The bulb holder, whose temperature will approximate to the ambient temperature, takes on the function of the radiation source. Our bolometer uses two lamps. One light bulb measures the radiation whilst the other bulb provides a reference. This twin-bulb solution has the advantage that ambient temperature does not influence the resulting measurement. As **Figure 2** shows, the bulbs are mounted in a cylindrical housing that is open on one side. In the associated circuitry (**Figure 3**)  $R_{p1}$  and  $R_{p2}$  are two precision sensing resistors whilst  $R_{m1}$  and  $R_{m2}$  are the two bulbs. It is vital that the two diverging resistance paths are electrically symmetrical. If the bulbs differ in resistance from one another, the two branches can be brought back into balance by adjusting the current sensing resistors ( $R_{p1}$  or  $R_{p2}$ ). The voltages at the test points can drive an instrument amplifier such as an AD620 op-amp.  $R_G$  sets the gain, which must be at least 1,000. The sensitivity can be adjusted by varying the current in the range of about 1 to 10 mA. Using an AC power supply is not recommended as external signals might affect the operation of the bolometer; a battery is much better.

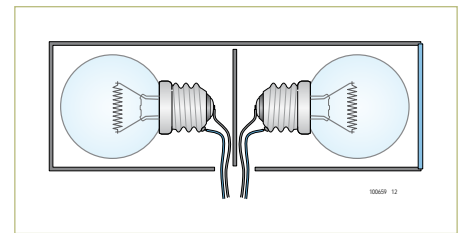


Figure 2. Mechanical construction.

All that's missing now is a suitable radiation source, such as a light fixture using a conventional light bulb, a switched-on soldering iron or a hot cup of coffee. A signal should appear on the output, as shown in **Figure 4**.

(100659)

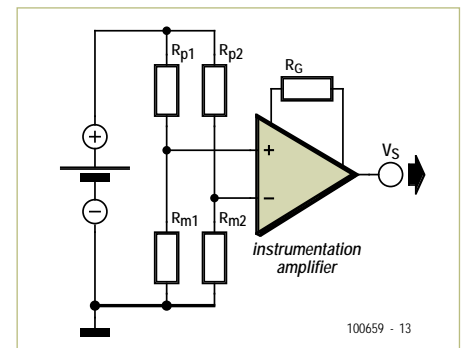


Figure 3. The diverging resistance paths must be symmetrical.

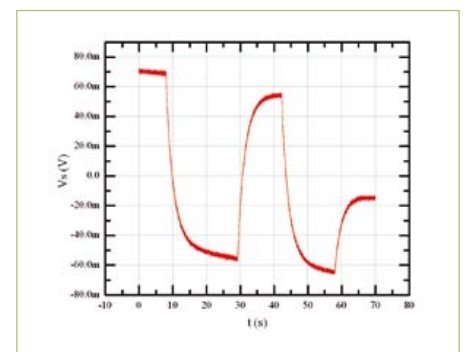


Figure 4. The output signal varies as the radiation source is brought closer and removed.



# The Entire Circuit Cellar Archive Ready to Go Where You Go

*Choose the Style that Fits You Best*

## CC Gold

A Stylish, Durable, Portable,  
Gold-Plated USB Drive



## CC Key

The Key to Your Success  
Is Always Handy

## CC Slim

The USB Archive that  
Fits in Your Wallet



The complete Circuit Cellar issue archive and article code stored on a USB flash drive. You can easily keep your CC Gold archive up to date by purchasing subsequent issues from our webshop or by downloading issues with a Circuit Cellar Digital 2.0 Subscription. Issues appear in searchable PDF format.

Complete archive includes PDFs of all issues in print through date of purchase.

**Visit [www.cc-webshop.com](http://www.cc-webshop.com) to purchase**



Circuit Cellar  
Archive Index

# Exotic Tubes Facebook

By Reginald Neale (USA)

After the XXL installments with ditto texts and theory published the previous two months, Retronics relaxes for a bit with a tube (British: 'valve') sightseeing tour or, if you like, a trip down Tube Memory Lane. Which of the tubes pictured here do you remember, vaguely or acutely? Which ones are gathering dust in your attic? And for tube haters: which ones are you in denial about?

Retronics contributor Reginald Neale writes from Farmington, NY: "When I started my engineering career in the early 1950s, everything was done with vacuum (and gas-filled) tubes. Transistors were still a laboratory curiosity. Here are a few of the more exotic and elegant types." (110387)



Tubes from the 30s, like the one on the left, had 4, 5, 6, or 7 base pins of different sizes. Then came the keyed octal base, the 9 and 7-pin miniature, the 12-pin compacton, and subminiature types.



An 'acorn' tube. With its planar construction and radial terminals, it was designed for the Ultra High Frequency band. In the late 40's, UHF was the high frequency frontier.



DuMont twin-gun cathode-ray tube from WWII radar display. Face of tube is about 5" diameter. One gun writes the scale/text information, the other writes the radar echo image.



Thyratrons are the tube equivalent of Silicon Controlled Rectifiers (SCRs). Conduction is initiated by a control grid, and continues until the anode-cathode current drops to zero.





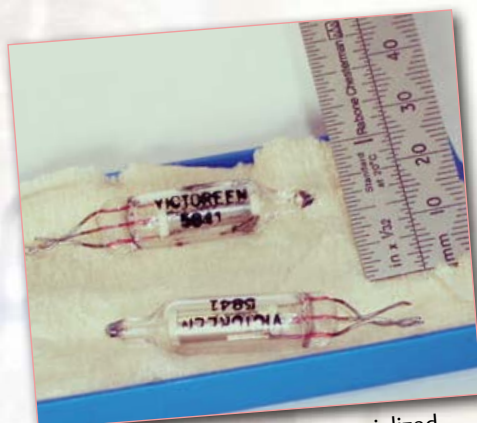
Plenty of big names in the radio/television/tube industry have vanished. Who remembers CBS-Hytron, Philco, Stromberg-Carlson, Capehart, National Union, Admiral, Emerson, Tung-Sol, Sylvania, Muntz, Dumont, Wells-Gardner, Crosley, Raytheon, Motorola, Zenith?



Transmit/Receive tube from a WWII radar system. Mounted in a waveguide and triggered by a high-voltage pulse, it shorted out a node in the waveguide, effectively switching between the transmitting and receiving modes at a high speed.



A 4X150 tetrode power transmitter tube. The metal anode has integral fins for forced-air cooling. Very famous — “every radio amateur has one”.



The Victoreen 5841 is a specialized subminiature high-voltage gas regulator tube. It regulates at 900 VDC for a load of 5-100  $\mu$ A.



Inside this tube is a heater and a thermocouple, which generates a tiny vacuum-dependent voltage. An 1/8" NPT fitting connects to the vacuum system.



This 1950s X-Ray tube is about a foot long. It has a fixed anode and cathode. Modern tubes increase power by using a large anode that rotates at a high speed.



“Miniature” phototubes were one of the innovations that made sound movies possible in the 30s. Optical sound tracks recorded the audio information on the edge of the film. Later sound systems used magnetic stripes, which resulted in improved sound quality.



The 832-A twin tetrode transmitting tube. An old time favorite among radio hams. No neutrodynization as opposed to the European QQE06/40 (see also Retronics December 2008).

*Retronics is a monthly column covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcomed; please send an email to [editor@elektor.com](mailto:editor@elektor.com)*



# Speaking Out Loud

By Gerard Fonte (USA)

There are a number of surveys that show that the fear of public speaking is greater than the fear of death. And, unfortunately, most people (including engineers) will eventually find themselves all alone in front of a lot of staring people, explaining a new product or process. It's not really bad as a lot of people fear.

Nearly everybody gets nervous before their presentation. But most find that after a minute or two, it gets much easier. A lot find it enjoyable after they get past the initial self-consciousness. It's like jumping into a swimming pool and finding that the water is really nice. And, like most things, the more you do it, the easier it gets.

However, there are some people who can't get over the early jitters. For them the whole process seems like endless torture. These are the people who have a true phobia about public speaking. Unfortunately, these people generally can't get accustomed to talking in front of others no matter how many times they do it. Usually, counseling is the only way to manage this, or any, phobia.

## It's Showtime

It's useful to think of public speaking as a performance. As such, you have to know your lines and understand the mechanics of the show. Obviously, being prepared makes a huge difference in any undertaking. Conversely, trying to improvise your whole presentation is not likely to produce happy results. Therefore, the first tip is to practice your presentation a number of times before you get up on stage.

However, unlike a role in a play, the monolog isn't fixed. In fact, you shouldn't memorize your lines at all (or read your presentation). If you do, you can appear wooden and uninteresting, which turns people off. Instead you should have a set of talking points that you want to discuss. And the nice thing is that you can simply put your talking points on slides, so you don't have to memorize them, either. By putting just a few lines on a slide you also streamline the visuals. This is important. Too many novice speakers try to cram too much information on each slide. The result is that they are not readable from the back of the room and you spend an inordinate amount of time on one slide which causes people's interest to fade. Conversely, every time you present a new picture, the audience's interest perks up. That's just human nature. So, you should only have a visual last for no more than a minute or so. Thus, if you have a 15 minute presentation, you should prepare about 10 to 20 slides.

Like an actor, you need to have a character. By this, I mean that you should be enthusiastic and motivated about your subject. Let's face it, if you aren't excited about your show, why should anyone else be? Standing in one place and droning on and on in a monotone is not effective. Smiling, walking around and being (or acting) confident makes a tremendous difference in your credibility. Yes, it is possible to take this too far and start looking like a used-car salesman. But in reality, this is rarely a problem. And, being over-animated at least keeps the listeners attentive.

## One on One

One very effective method of reducing the stress of public speaking is to talk to one person. Pick someone in the crowd (preferably

in the middle) and pretend that you are discussing your topic only to him or her.

This helps you focus. It's easy to get overwhelmed by a sea of faces. But if you are just chatting with someone it's a lot easier. If you can, change your individual from time to time. Addressing one person for too long can make that person uncomfortable. It's also useful to spread your attention over the whole room.

Of course, there is no rule that you can't put a ringer in the crowd. Have a friend be present and address him or her. This can reduce your anxiety significantly. And, it's certainly possible to get feedback from facial expressions or other indicators. This lets you know how well you are doing. Having your ally give you a thumbs-up after you just explained a difficult topic is very comforting and reassuring.

## Speak the Language

It can't be stressed too much that you have to know your audience. You will discuss your material differently for managers, salespeople, customers or engineers. Clearly, technical jargon and esoteric details will be lost on salespeople. And failing to discuss the specialized aspects to a gathering of engineers will not garner much credibility. You should know in advance who is coming to hear you and target that group.

Of course, there will be times when there is a mixture of groups. This often happens at a general company meeting to discuss new projects, etc. In this case, there are a couple of things you can do. The first is to talk to the decision makers (most likely top management) using their vocabulary and discuss the points that concern them. These are the most important people to you and your presentation and are the ones you have to impress. There's nothing wrong with being pragmatic.

The other approach is to recognize the presence of the different groups and acknowledge them during your talk. This is more difficult and time consuming. You essentially have to translate technical terms used for the engineers into simplified concepts for the others (or vice versa). But, if you have the time, this method can be extremely effective. For example, you might say something like this: "The new analyzer uses a 6-pole elliptic filter to eliminate aliasing. What this means is that there is a special circuit that removes signals that could cause problems for the software." You probably won't have to translate every sentence like this. But, you have to make sure that both the technical and non-technical groups get enough information to satisfy them.

Speaking to a crowd doesn't have to be a death-defying act.

(110680)



A stack of three Elektor magazine covers is visible in the top right corner. The top cover features a red robot, the middle one shows a circuit board, and the bottom one has a yellow radiation symbol.

# 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 subscription!**

Subscribe now or upgrade: [www.elektor.com/usa](http://www.elektor.com/usa)

# Hexadoku

## Puzzle with an electronics touch

With the autumn leaves drifting by your window and not too many electronics projects on the workbench, why not relax for a bit and stimulate the gray matter at the same time! It sure is possible — start entering the right numbers in the puzzle below. Next, send the ones in the gray boxes to us and you automatically enter the prize draw for one of four Elektor Shop vouchers. Have fun!

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  $16 \times 16$  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  $4 \times 4$  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 \$ 140.00\* and three Elektor Shop Vouchers worth \$ 70.00\* each, which should encourage all Elektor readers to participate.

\*Subject to exchange rate.

### Participate!

**Before December 1, 2011**, send your solution (the numbers in the gray boxes) by email, fax or post to

Elektor Hexadoku – 4 Park Street – Vernon CT 06066 – USA

Fax 860 8751-0411

Email: [hexadoku@elektor.com](mailto:hexadoku@elektor.com)

### Prize winners

The solution of the September 2011 Hexadoku is: **4D0F6**.

The Elektor \$140.00 voucher has been awarded to Alf Eriksson (Sweden).

The Elektor \$70.00 vouchers have been awarded to M.F. Vidaud (France),

Parisi Vincenzo (Switzerland) and Monika Häfner (Germany).

Congratulations everyone!

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

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

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



Save

66%

off the cover price!

Delivering insightful analysis of the hottest topics in embedded tech

Subscribe to  
Circuit Cellar NOW!

**CIRCUIT CELLAR**  
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

[www.circuitcellar.com](http://www.circuitcellar.com)

# 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](mailto:ElektorUSA@smmarketing.us)

**SCOPES and more**

**HAMEG®**  
Instruments

A Rohde & Schwarz Company

[www.hameg.com](http://www.hameg.com)

**Great Value in  
TEST & MEASUREMENT**

**INTEGRATED**  
PLCs with Integrated Ethernet



From  
\$129

- 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](http://www.tri-plc.com/ek.htm)

**TRI** TRIANGLE  
RESEARCH  
INTERNATIONAL

**BPS** BusBoard Prototype Systems [www.BusBoard.us](http://www.BusBoard.us)

Many PCB styles available

BB400T Transparent BreadBoard

SB400 Solderable PC BreadBoard

Solder and Keep

Build and Test

Available from:

**JAMECO** ELECTRONICS

**amazon.com**

**MOUSER** ELECTRONICS

**Build Your Own:**

**THEREMIN**

Theremax  
The Ultimate Themin

**PAIA** [paia.com](http://paia.com) Music/Tutorials/Kits  
Analog Synthesizers  
Guitar Effects  
Tube Electronics  
Studio Gear

Makers since 1959

**SCIENCE, ROBOTICS & ELECTRONICS**

**IMAGES Scientific Instruments Inc.**

Microcontrollers  
Servo Motor Controllers  
Artificial Vision  
Speech Recognition  
Flex Sensors

[www.imagesco.com](http://www.imagesco.com)

Tele: (718) 966-3694 Fax: (718) 966-3695

**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

[www.pololu.com](http://www.pololu.com)

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

**Tag-Connect**

JTAG Cables Plug Directly To PCB!

The **GREENEST** Connector is **NO CONNECTOR!!!**

**Save Space & Cost on every Board!!**

[www.PlugOfNails.com](http://www.PlugOfNails.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](http://www.apcircuits.com)

VISA MasterCard PayPal JPC 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](mailto:ElektorUSA@smmarketing.us)

**Microcontrollers 8 & 32 bit**

MC9S08AS128	MCF51AC256
MC9S08JM64	MCF51CN128
MC9S08QE128	MCF51JM128
	MCF51QE128

Bluetooth • RS-232 • USB • RF • 2x20 LCD  
4x4 Keypad • Motor Control • Real Time Clock

Prototypes quickly and easily

Visit Us Now  
Learn programming

Explore microcontrollers  
New Modules For 2011  
Experiment

**BASIC ON BOARD !**

**ATRIA Technologies Inc.**  
[www.AtriaTechnologies.com](http://www.AtriaTechnologies.com)

**MaxSonar**  
Ultrasonic Range Finder

**XL-MaxSonar-EZ**

- Beam pattern choice
- High acoustic power
- Real-time calibration
- 39.95USD / unit

**MaxSonar-WRC IP67**

- Compact packaging
- Quality narrow beam
- 99.95USD / unit

[www.active-robot.co.uk](http://www.active-robot.co.uk)  
[www.coolcomponents.co.uk](http://www.coolcomponents.co.uk)  
[www.oceancontrols.com.au](http://www.oceancontrols.com.au)  
[www.maxbotix.com](http://www.maxbotix.com)

# products and services directory

## SHOWCASE YOUR COMPANY HERE

Elektor provides you with more ways to promote your business with our Showcase section -- a permanent feature of the magazine where you will be able to further highlight your products and services.

- For just \$1100 (\$100 per issue for eleven issues), Elektor will publish your company name, website address and a 30-word description.
- For \$1650 (\$150 per issue for eleven issues), we will publish the above plus run a 2-inch square, full-color image

– i.e., product shot, screen shot from your website, company logo, catalog cover – or whatever you choose to showcase.

Places are limited and spaces will go on a strictly first come, first served basis. So please fax back your order today!

I wish to promote my company, please book my space:

- Text insertion only for \$1100
- Text and photo for \$1650

NAME: ..... ORGANIZATION: .....

JOB TITLE: .....

ADDRESS: .....

..... TEL: .....

PLEASE COMPLETE COUPON BELOW AND FAX BACK TO 1-978-281-7706

COMPANY NAME .....

WEB ADDRESS .....

30-WORD DESCRIPTION .....

.....

.....

.....

.....

## Elektor OSPV

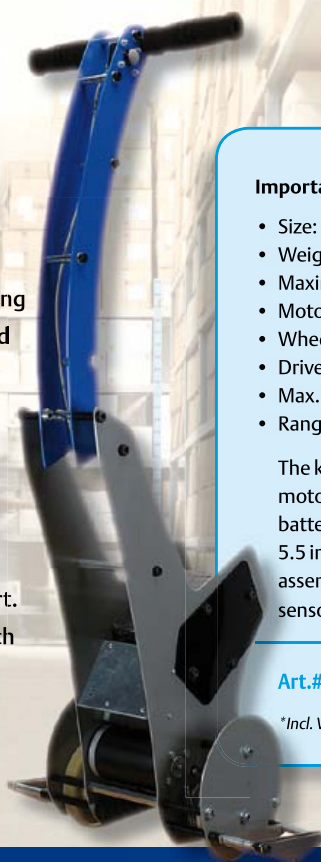
### ➡ Open Source Personal Vehicle

Last year we launched the Elektor Wheelie, a self-balancing personal transport device. Our new Elektor OSPV is based on the same concept, but with the difference that it's for indoors, it's easy to steer, it's light and foldable and...

it's open source. You can configure or modify it to suit your wishes! The OSPV is primarily intended for moving people, but it doesn't have to be limited to that.

A variety of other uses are conceivable, ranging from an electric wheelbarrow to a handy motorised shopping cart.

This is where the advantages of the open source approach come to the fore!



**Price Slashed!**

#### Important specifications:

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

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

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

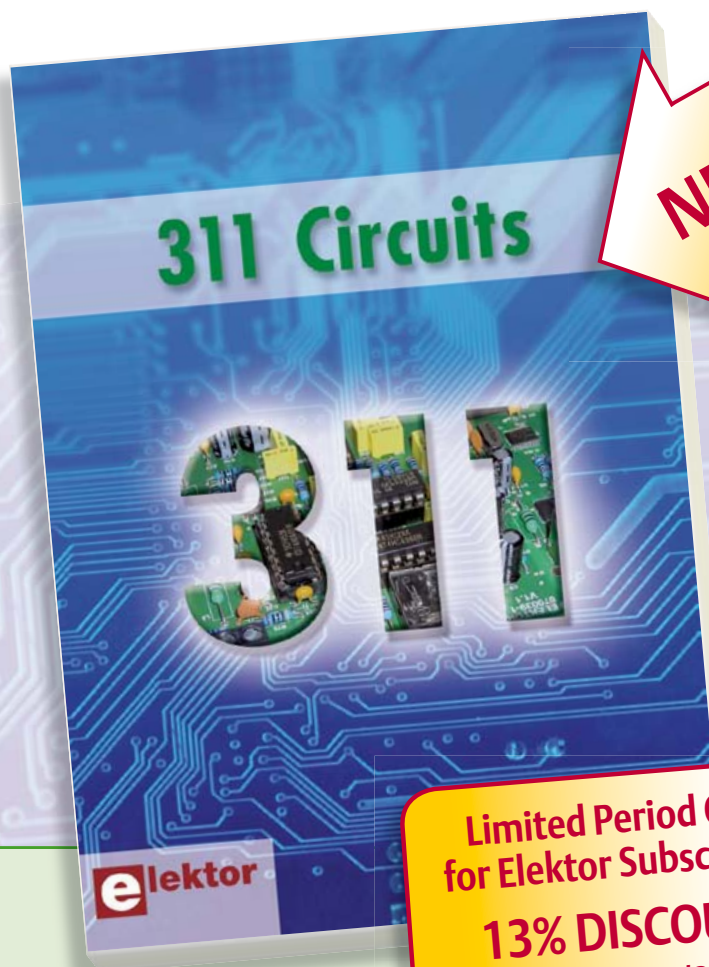
\*Incl. VAT, excl. shipping costs

Further information, demo video and ordering at [www.elektor.com/ospv](http://www.elektor.com/ospv)



# Going Strong

A world of electronics  
from a single shop!



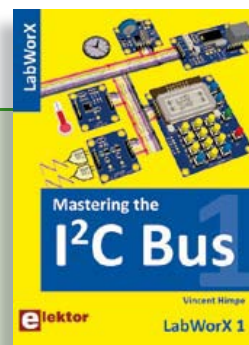
Limited Period Offer  
for Elektor Subscribers!  
**13% DISCOUNT**  
[www.elektor.com/311circuits](http://www.elektor.com/311circuits)

Creative solutions for all areas of electronics

## 311 Circuits

311 Circuits is the twelfth volume in Elektor's renowned 30x series. These Summer Circuits compilation books have been bestsellers for many years. This brand new book contains circuits, design ideas, tips and tricks from all areas of electronics: audio & video, computers & microcontrollers, radio, hobby & modelling, home & garden, power supplies & batteries, test & measurement, software, not forgetting a section 'miscellaneous' for everything that doesn't fit in one of the other categories. 311 Circuits offers many complete solutions as well as useful starting points for your own projects. Both categories and anything in between represent a veritable fountain of inspiration for cultivating your own ideas and learning about electronics. This book deserves a place not far from the workbench!

420 pages • ISBN 978-1-907920-08-0 • \$47.60

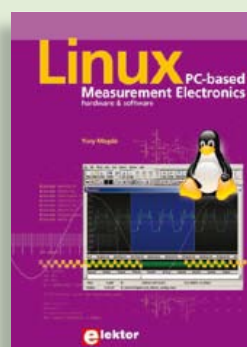


LabWorX: Straight from the Lab to your Brain

## Mastering the I<sup>2</sup>C Bus

Mastering the I<sup>2</sup>C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I<sup>2</sup>C Bus and its applications. Besides the Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I<sup>2</sup>C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I<sup>2</sup>C probe and a software framework to control I<sup>2</sup>C devices from your computer.

248 pages • ISBN 978-0-905705-98-9 • \$47.60

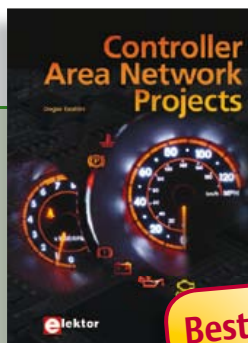


A highly-practical guide

## Linux – PC-based Measurement Electronics

If you want to learn how to quickly build Linux-based applications able to collect, process and display data on a PC from various analog and digital sensors, how to control circuitry attached to a computer, then even how to pass data via a network or control your embedded system wirelessly and more – then this is the book for you! The book covers both hardware and software aspects of designing typical embedded systems using schematics, code listings and full descriptions.

264 pages • ISBN 978-1-907920-03-5 • \$47.60



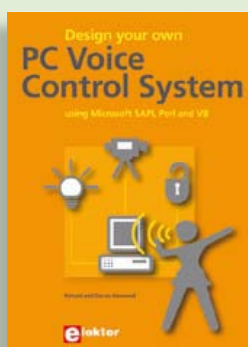
**Bestseller!**

Free mikroC compiler CD-ROM included

## 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

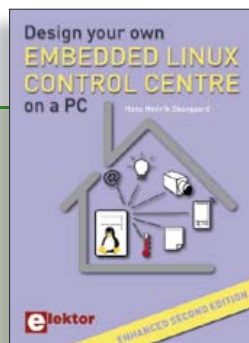


Talk with your computer

## Design your own PC Voice Control System

This book guides you through practical speech recognition, speech annunciation and control of really useful peripherals. It details a project which will enable you to instruct your computer using your voice and get it to control electrical devices, tell you the time, check your share values, get the weather forecast, etc. and speak it all back to you in a natural human voice. If you are interested in the practical technology of interfacing with machines using voice, then this book is your guide!

216 pages • ISBN 978-1-907920-07-3 • \$47.60



Enhanced second edition

## 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, which will be the interface to your very own home control centre. 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 low-cost 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

**More information on the  
Elektor Website:  
[www.elektor.com](http://www.elektor.com)**

**Elektor US**  
PO Box 180  
Vernon, CT 06066  
USA  
Phone: 860-875-2199  
Fax: 860-871-0411  
E-mail: [order@elektor.com](mailto:order@elektor.com)

**elektor**



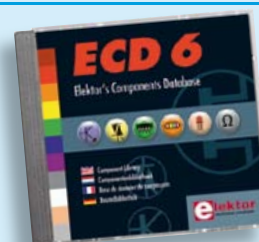
**Bestseller!**

Circuits, ideas, tips and tricks from Elektor

## 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 & measurement and of course a section miscellaneous for everything that didn't fit in one of the other sections. Texts and component lists may be searched with the search function of Adobe Reader.

ISBN 978-1-907920-06-6 • \$55.70



More than 70,000 components

## CD Elektor's Components Database 6

This CD-ROM gives you easy access to design data for over 7,800 ICs, more than 35,600 transistors, FETs, thyristors and triacs, just under 25,000 diodes and 1,800 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 data-bank applications are fully interactive, allowing the user to add, edit and complete component data. This CD-ROM is a must-have for all electronics enthusiasts!

ISBN 978-90-5381-258-7 • \$40.20

CD/DVD-ROMS



More than 25 projects based on the Elektor ATM18 board

### cd ATM18 Collection

This CD-ROM contains all articles from the popular ATM18-CC2 series published in Elektor magazine. From RFID Reader and Bluetooth linking right up to a chess computer! Project software and PCB layouts in PDF format are included. What's more, this CD also contains a Bascom AVR programming course and helpful supplementary documentation.

ISBN 978-0-905705-92-7 • \$39.60

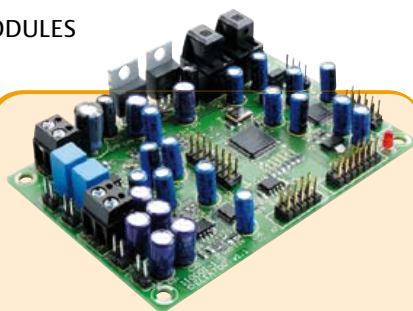


All articles in Elektor Volume 2010

### dvd Elektor 2010

This DVD-ROM contains all editorial articles published in Volume 2010 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, adapt PCB layouts using your favourite graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-267-9 • \$37.90



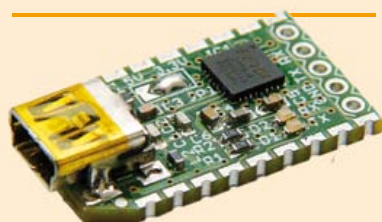
### Audio DSP Course

(September 2011)

This DSP board is the platform for the applications described in our Audio DSP Course. It is also intended to enable you to develop your own initial digital audio signal processing applications. The DSP board can be used stand-alone as is, and even though it is an ideal learning platform, with its 24-bit signal processing capability for sampling rates up to 192 kHz and its high-performance interfaces, it is also suitable for applications with very stringent quality requirements for both signal to noise ratio and DSP computing power.

*Populated and tested DSP board*

Art.# 110001-91 • \$186.70



### FT232R USB/ Serial Bridge/BOB

(September 2011)

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



### USB Long-Term Weather Logger

(September 2011)

This stand-alone data logger displays pressure, temperature and humidity readings generated by I<sup>2</sup>C bus sensors on an LCD panel, and can run for six to eight weeks on three AA batteries. The stored readings can be read out over USB and plotted on a PC using gnuplot. Digital sensor modules keep the hardware simple and no calibration is required.

*Kit of parts incl. PCB, controller, humidity sensor and air pressure sensor modules*

Art.# 100888-73 • \$50.20



### Pico C Meter

(April 2011)

RF and radio repair fans probably do need to be told, but when it comes to measurements below 200 pF or so, modern DMMs will produce coarse if not ridiculous results. Elektor's purpose-designed Pico C does a far better job. Beating many DMMs hands down, this little instrument easily and accurately measures capacitances down to fractions of a picofarad!

*Kit of parts incl. Elektor Project Case, programmed microcontroller, LCD and PCB*

Art.# 100823-71 • \$118.40



## November 2011 (No. 35)

\$

+ + + Product Shortlist November: See [www.elektor.com](http://www.elektor.com) + + +

## October 2011 (No. 34)

## Versatile Board for AVR Microcontroller Circuits

100892-1 ..... Printed circuit board.....18.70

## Audio DSP Course (4)

110001-91 .... PCB, populated and tested DSP board.....186.70

110001-92 .... Bundle DSP board (110001-92)  
with Programmer (110534-91) .....133.50

## Here comes the Bus! (8)

110258-1 ..... Experimental Node board.....8.60

110258-1C3 .. Printed circuit board Experimental Nodes (3 PCBs).....18.60

110258-91 .... USB/RS485 Converter, ready made module .....35.90

## September 2011 (No. 33)

## eC-Reflow-Mate

100447-91 .... Professional SMT reflow oven .....3495.00

## USB Long-Term Weather Logger

100888-1 ..... Printed circuit board.....25.90

100888-41 .... Programmed controller ATMEGA88-20PU .....14.30

100888-71 .... HH10D humidity sensor module .....11.50

100888-72 .... HP03SA air pressure sensor module .....9.30

100888-73 .... Kit of parts incl. PCB, controller, humidity sensor  
and air pressure sensor modules .....50.20I<sup>2</sup>C Sensors

100888-71 .... HH10D humidity sensor module .....11.50

100888-72 .... HP03SA air pressure sensor module .....9.30

## E-Blocks go Twitter

EB003 ..... E-blocks Sensor board .....34.90

EB005 ..... E-blocks LCD board.....38.80

EB006 ..... E-blocks PIC Multiprogrammer.....116.20

EB007 ..... E-blocks Switch board .....23.30

EB059 ..... E-blocks Servo board .....23.30

EB069 ..... E-blocks Wireless LAN board.....212.90

TEDSS14 ..... Flowcode 4 for dsPIC/PIC24 .....288.40

## FT232R USB/Serial Bridge/BOB

110553-91 .... PCB, assembled and tested.....20.90

## Here Comes the Bus! (7)

110258-1 ..... Experimental Node board, bare .....8.60

110258-1C3 .. 3 x Experimental Node board, bare .....18.60

110258-91 .... USB/RS485 Converter, ready made module .....35.90

## J2B: Universal MMI Module using ARM Cortex-M3

050176-74 .... Enclosure Bopla Unimas 160.....14.30

110274-71 .... Tested PCB with LPC1343 microcontroller, crystal,  
3V3 voltage regulator, LCD interface & USB interface mounted.  
LED and headers.....56.50110274-72 .... LC-display 4 x 20 characters  
(HD44780 compatible) .....14.40

## July/August 2011 (No. 31/32)

## 2/4/6-hour Timer

110219-41 .... Programmed controller PIC12F675 DIL8.....14.30

## Morse Clock

110170-41 .... Programmed controller ATtiny4520-PU dip 8.....14.30

## Return of the Elex Prototyping Board

ELEX-1 ..... Printed circuit board Universal exp. board.....7.90

ELEX-2 ..... Printed circuit board Double version universal  
exp. board.....14.25ELEX-4 ..... Printed circuit board Quadruple  
version universal exp. board .....25.80

## Arc Welding Effect for Model Railway Layouts

110085-41 .... Programmed controller PIC10F200-I/P (DIP8) .....14.30

## Slave Flash for Underwater Camera

100584-41 .... Programmed controller PIC12F675i/p DIL8 .....14.30

## WAV Doorbell

110080-41 .... Programmed controller ATmega328p DIL28 .....18.60

## Bestsellers

## Books

- 1 ➡ **Controller Area Network Projects**  
ISBN 978-1-907920-04-2 ..... \$47.60
- 2 ➡ **Mastering the I<sup>2</sup>C Bus**  
ISBN 978-0-905705-98-9 ..... \$47.60
- 3 ⬆ **Design your own  
PC Voice Control System**  
ISBN 978-1-907920-07-3 ..... \$47.60
- 4 ➡ **Linux – PC-based measurement electronics**  
ISBN 978-1-907920-03-5 ..... \$47.60
- 5 ➡ **Introduction to Control Engineering**  
ISBN 978-0-905705-99-6 ..... \$44.40

## CD/DVD-ROMs

- 1 ➡ **CD 1001 Circuits**  
ISBN 978-1-907920-06-6 ..... \$55.70
- 2 ➡ **DVD Elektor 2010**  
ISBN 978-90-5381-267-9 ..... \$37.90
- 3 ➡ **DVD Elektor 1990 through 1999**  
ISBN 978-0-905705-76-7 ..... \$111.30
- 4 ➡ **CD Elektor's Components Database 6**  
ISBN 978-90-5381-258-7 ..... \$40.20
- 5 ➡ **CD ATM18 Collection**  
ISBN 978-0-905705-92-7 ..... \$39.60

## Kits &amp; Modules

- 1 ➡ **USB Long-Term Weather Logger**  
Art. # 100888-73 ..... \$50.20
- 2 ⬆ **FT232R USB/Serial Bridge/BOB**  
Art. # 110553-91 ..... \$20.90
- 3 ➡ **Here comes the Bus!**  
Art. # 110258-91 ..... \$35.70
- 4 ➡ **Pico C Meter**  
Art. # 100823-71 ..... \$118.40
- 5 ⬆ **Audio DSP Board**  
Art. # 110001-91 ..... \$186.70

Order quickly and securely through  
[www.elektor.com/shop](http://www.elektor.com/shop)  
 or use the Order Form near the end  
 of the magazine!

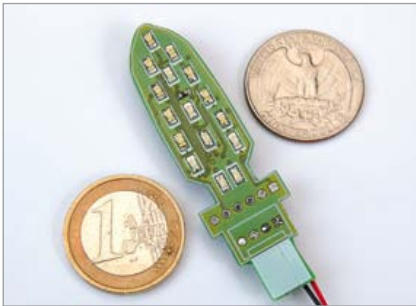


Elektor US  
 PO Box 180  
 Vernon, CT 06066  
 USA  
 Phone: 860-875-2199  
 Fax: 860-871-0411  
 E-mail: [order@elektor.com](mailto:order@elektor.com)



### USB Data Logger

A PC is often used to store data over a longer period of time but to be honest it's a big energy consumer. Using the USB Data Logger and in particular its serial interface it is possible for data collected by a microcontroller circuit to be stored on a standard USB stick. The solution requires little energy, provides secure storage and obviates the need for a separate logging function in the existing microcontroller system.



### Smart LED Candle

Well before the Christmas holidays every electronics fan should be on the hunt for circuits and gizmos that bring the right atmosphere to the living room. Allow Elektor to assist you with the publication of a unique Christmas circuit: an intelligent LED candle that lights up when you touch it and is quenched with a puff. The circuit is built around a PIC 16F1827 microcontroller.



### Lambda Probe Interface

This little circuit contains all functionality to drive a wideband lambda probe. It uses a special IC type CJ125 from Bosch, which gets combined with a Bosch LSU4.2 broadband probe. The pair allows accurate measurement of the remaining amount of oxygen in combustion gases; hence helps determine whether the inlet gas is lean or rich. The circuit has no external adjustment points, the communication is done via a digital interface that requires no more than hooking up a TTL UART interface.

Article titles and magazine contents subject to change; please check the Magazine tab on [www.elektor.com](http://www.elektor.com)

Elektor UK/European December 2011 edition: on sale November 17, 2011.

Elektor USA November 2011 edition: published November 14, 2011.

[www.elektor.com](http://www.elektor.com) [www.elektor.com](http://www.elektor.com) [www.elektor.com](http://www.elektor.com) [www.elektor.com](http://www.elektor.com) [www.elektor.com](http://www.elektor.com) [www.elektor.com](http://www.elektor.com)

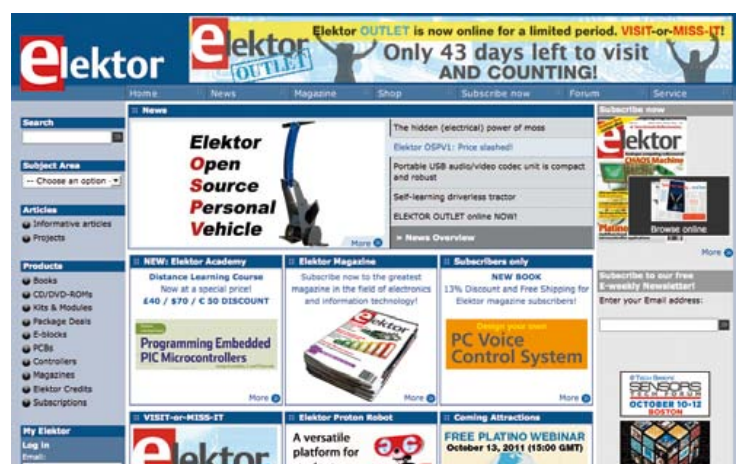
## Elektor on the web

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







## ORDERING INFORMATION

To order contact customer service:

Phone: 860-875-2199  
Fax: 860-871-0411  
Mail: Elektor US  
PO Box 180  
Vernon, CT 06066  
USA  
E-mail: [sales@elektor.com](mailto:sales@elektor.com)  
On-line at [www.elektor.com](http://www.elektor.com)

Customer service hours: 8:00 AM–4:30 PM Monday–Thursday. Voice mail available at other times.  
When leaving a message please be sure to leave a daytime telephone number where we can return your call.

**PLEASE NOTE:** While we strive to provide the best possible information in this issue, pricing and availability are subject to change without notice. To find out about current pricing and stock, please call or email customer service.

## COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in the magazine. If difficulties in obtaining components are suspected, a source will normally be identified in the article. Please note, however, that the source(s) given is (are) not exclusive.

## PAYMENT

Orders must be prepaid. We accept checks or money orders (in US \$ drawn on a US bank only), VISA, Mastercard, Discover, and American Express credit cards. We do not accept C.O.D. orders.  
We also accept wire transfers. Add \$20 to cover fees charged for these transfers.

## TERMS OF BUSINESS

**Shipping Note:** All orders will be shipped from Europe. Please allow 3–4 weeks for delivery. Shipping and handling via airmail: US \$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.

## SUBSCRIPTIONS (US & CANADA ONLY)

### Subscription rates (1 Yr.)

Standard Subscription: \$39.95  
Plus Subscription: \$59.95

Canada add \$11 per year for postage

All subscriptions begin with the current issue. Expect 3–4 weeks for receipt of the first issue. Subscriptions, renewals, and change of address should be sent to:

Elektor US  
PO Box 180  
Vernon, CT 06066  
USA

E-mail: [sales@elektor.com](mailto:sales@elektor.com)

Order subscriptions on-line at [www.elektor.com/subs](http://www.elektor.com/subs)

Subscriptions 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 subscriptions, please include gift recipient's name and address as well as your own, with remittance. A gift card will be sent on request.  
Subscriptions may be cancelled at any time for a refund of all unmailed issues.

**Does your subscription expire soon?**

Renew it online at [www.elektor.com/renew](http://www.elektor.com/renew)

# Elektor App for iPhone and iPad



## The Elektor Electronic Toolbox

Elektor now offers an App you cannot afford to miss on your iPhone, iPod Touch or iPad. The Elektor Electronic Toolbox is a collection of no fewer than 28 electronic tools that can be picked from a comprehensive set of icons!

### Highlights:

- Databases for Transistors, FETs, Triacs, Thyristors, Diodes and ICs
- NE555 circuit design
- An Ohms Law calculator
- Schematic Diagrams
- Number base converter
- LED / resistor calculation
- R/L and BJT calculations
- And more



Now available from the Apple iTunes Store for just \$5.99!

Further information at

[www.elektor.com/app](http://www.elektor.com/app)



## Take out a free subscription to E-weekly now

Do you want to stay up to date with electronics and computer technology? Always looking for useful hints, tips and interesting offers? Subscribe now to E-weekly, the free Elektor Newsletter.

### Your benefits:

- The latest news on electronics in your own mailbox each Friday
- Free access to the News Archive on the Elektor website
- You're authorized to post replies and new topics in our forum



Register today on [www.elektor.com/newsletter](http://www.elektor.com/newsletter)

## INDEX OF ADVERTISERS

AP Circuits, Showcase . . . . .	<a href="http://www.apcircuits.com">www.apcircuits.com</a> . . . . .	27, 78
Atria Technologies, Showcase . . . . .	<a href="http://www.AtriaTechnologies.com">www.AtriaTechnologies.com</a> . . . . .	78
AudioXpress . . . . .	<a href="http://www.cc-webshop.com">www.cc-webshop.com</a> . . . . .	2
BusBoard, Showcase . . . . .	<a href="http://www.BusBoard.us">www.BusBoard.us</a> . . . . .	78
2012 International CES . . . . .	2012 International CES . . . . .	12
DesignSpark chipKIT™ Challenge . . . . .	<a href="http://www.chipkitchallenge.com">www.chipkitchallenge.com</a> . . . . .	13
DLP Design . . . . .	<a href="http://www.dlpdesign.com">www.dlpdesign.com</a> . . . . .	39
ExpressPCB . . . . .	<a href="http://www.expresspcb.com">www.expresspcb.com</a> . . . . .	9
EzPCB . . . . .	<a href="http://www.stc-51.com">http://www.stc-51.com</a> . . . . .	27
Hameg Instruments, Showcase . . . . .	<a href="http://www.hameg.us">www.hameg.us</a> . . . . .	78
Images Scientific, Showcase . . . . .	<a href="http://www.imagesco.com">www.imagesco.com</a> . . . . .	78
Linx Technologies . . . . .	<a href="http://www.linxtechnologies.com">www.linxtechnologies.com</a> . . . . .	39
Loudspeaker Design Cookbook . . . . .	<a href="http://www.cc-webshop.com">www.cc-webshop.com</a> . . . . .	54

Maxbotix, Showcase . . . . .	<a href="http://www.maxbotix.com">www.maxbotix.com</a> . . . . .	78
Microchip . . . . .	<a href="http://www.microchip.com/smartphone">www.microchip.com/smartphone</a> . . . . .	88
MikroElektronika . . . . .	<a href="http://www.mikroe.com">www.mikroe.com</a> . . . . .	3
Paia Electronics, Showcase . . . . .	<a href="http://www.paia.com">www.paia.com</a> . . . . .	31, 78
Parallax . . . . .	<a href="http://www.parallax.com">www.parallax.com</a> . . . . .	27
Parts Express . . . . .	<a href="http://www.parts-express.com/sure">www.parts-express.com/sure</a> . . . . .	39
Pico . . . . .	<a href="http://www.picotech.com/pco462">www.picotech.com/pco462</a> . . . . .	11
Pololu Corporation, Showcase . . . . .	<a href="http://www.pololu.com">www.pololu.com</a> . . . . .	43, 78
Saelig . . . . .	<a href="http://www.saelig.com">www.saelig.com</a> . . . . .	65
Showcase . . . . .	. . . . .	78, 79

**Advertising space for the issue 8 December 2011  
may be reserved not later than 15 November 2011**

with Strategic Media Marketing, Inc. - 2 Main Street -  
Gloucester, MA 01930 - USA Telephone 1.978.281.7708 -  
Fax 1.978.281.7706 - e-mail: [ElektorUSA@smmarketing.us](mailto:ElektorUSA@smmarketing.us) to whom all  
correspondence, copy instructions and artwork should be addressed.

# Development Platforms for Smartphone and Tablet Accessories

Reduce Time to Market with Microchip Solutions



**Want to design and develop an accessory for smartphones or tablets? The mobile device marketplace is exploding with ideas and opportunity. While much of the high-tech world has settled down into evolutionary improvements, the mobile world offers the promise of entirely new uses for ever smarter devices.**

Microchip's new PIC24F Accessory Development Starter Kit for Android™ consists of a 16-bit PIC24 development board, a PICkit™ 3 In-Circuit Debugger and a royalty-free, no fee licensed software library (available via download), which enables the fast and easy development of Android smartphone and tablet accessories based on Microchip's large portfolio of 16-bit and 32-bit PIC® microcontrollers (MCUs).

## Development Board Key Features:

- PIC24F 16-bit PIC MCU with USB OTG
- Type A USB connector
- User interface buttons
- LEDs and potentiometer
- Device charger circuitry up to 500 mA
- I/O headers for prototyping and expansion

Let Microchip's PIC microcontroller-based development platforms help jump start your development of the next great accessory for remote control, home automation, financial transactions or portable medical devices.

Development resources are also available for iPod®, iPhone® and iPad®.  
For more information, visit: [www.microchip.com/MFI](http://www.microchip.com/MFI)

## IT'S EASY TO GET STARTED!

- Visit [www.microchip.com/android](http://www.microchip.com/android)
- Purchase the PIC24F Accessory Development Starter Kit for Android™
- Download the royalty-free, no-fee, licensed software library
- Contact [androidsupport@microchip.com](mailto:androidsupport@microchip.com) for additional support



PIC24F Accessory Development  
Starter Kit for Android  
(DM240415)