

**25+ pages of
Test & Measurement**

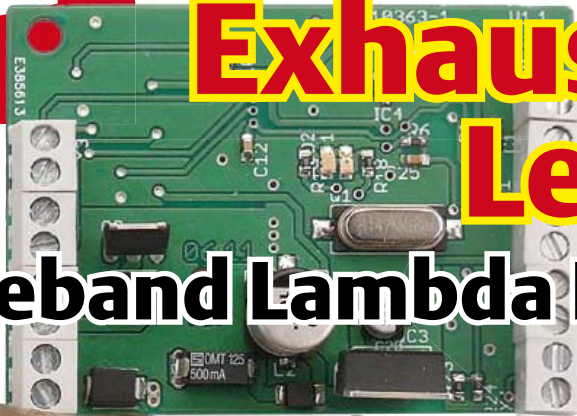
New course: Electronics for Starters

Elektor

www.elektor.com

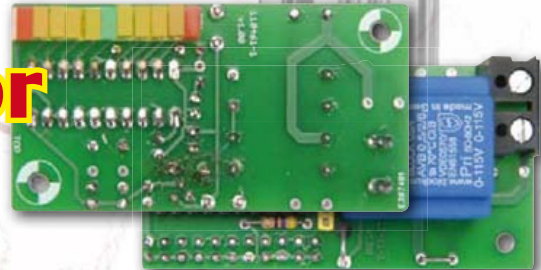
Exhaust Oxygen Level Tester

Wideband Lambda Probe Interface



Grid Frequency Monitor

Tiny fluctuations evidenced by LEDs



+ Wavelet Analysis
On a PIC32 development system

DCF77 Timecode Receiver

Ultra accurate & DSP based



+ Colette, Annette, Babette, Georgette
Tubed portable radios from the 1950s





DESIGNSPARK

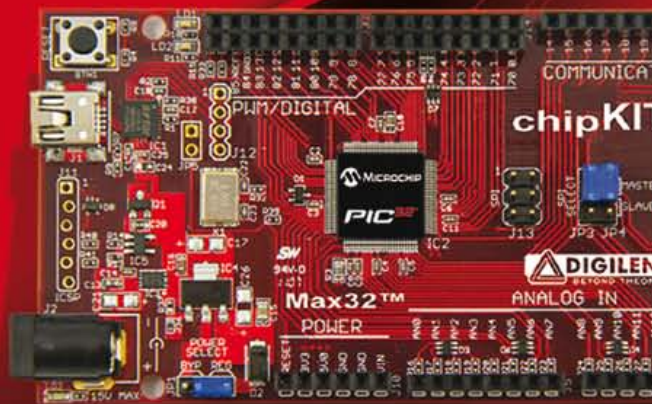
Turn a **hot**
idea into a
cool
solution.

DesignSpark chipKIT™ Challenge

The DesignSpark chipKIT™ Challenge is well under way! Have you registered at www.chipkitchallenge.com yet? Hurry over and get involved in the DesignSpark community today. By submitting your proposal for an energy-efficient design solution, you will automatically be considered for a **chipKIT™ Community Choice Award***. At the end of January, one lucky participant will be rewarded for having the most creative project proposal.

Awarded by the DesignSpark team, the winner of this sub-challenge will receive a **\$100 voucher for RS Components/Allie Electronics** and a **free digital subscription to Circuit Cellar and Elektor magazines!**

Register your project today!
Visit www.chipkitchallenge.com to participate.



Visit www.chipkitchallenge.com

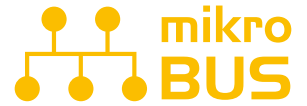
for complete rules and details.

* Participation in the Community Choice Awards does not increase your chances of winning the Grand Prize with your Final Project(s) submission. The deadline for Final Project submissions is March 27, 2012. See website for more information.

IN ASSOCIATION WITH:

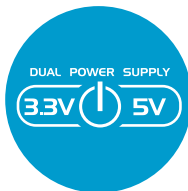
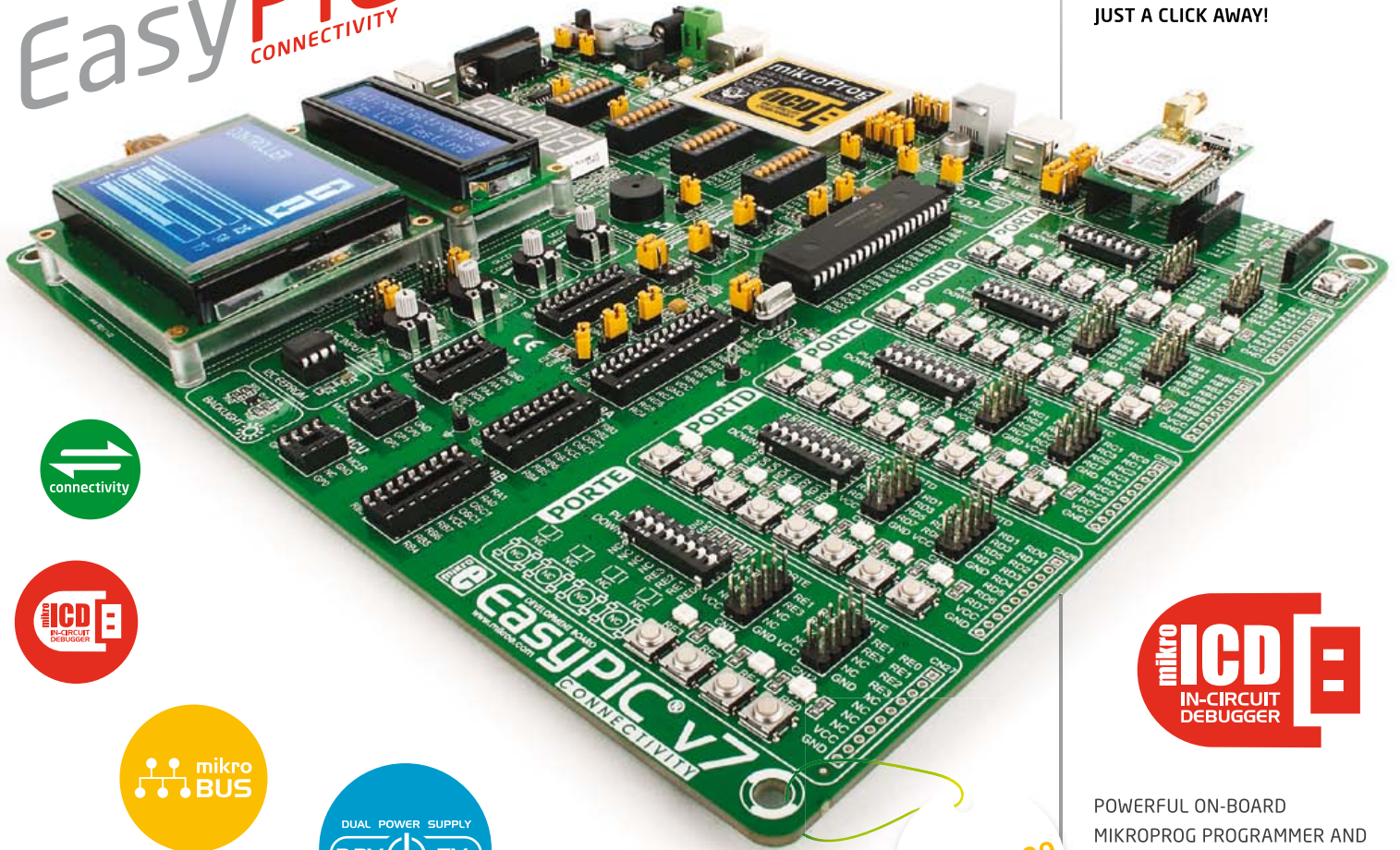


IT'S HERE!

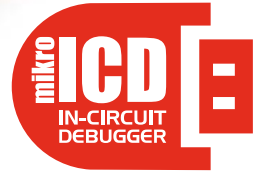


JUST PLUG IN ONE OF YOUR **CLICK BOARD™**, AND IT'S READY TO WORK. SPECIALIZED **MIKROBUS™ COMMUNICATION INTERFACE** MAKES YOUR DEVELOPMENT EASIER, AND ALLOWS SIMPLE, YET HIGHLY EFFECTIVE CONNECTIVITY. EVERYTHING IS NOW **JUST A CLICK AWAY!**

EasyPIC^{v7} CONNECTIVITY



\$149⁰⁰



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.

What's for starters? LEDs and micros!

Electronics is becoming increasingly complex and difficult to grasp in all its enormity. At least, that's what we are told occasionally by our readers, usually in personal conversations like phone calls or with a cup of coffee at shows. Over the past 50 years, electronics has evolved, grown, expanded — whatever you want call it — at a terrific speed, resulting in a constant need for training and refresher courses, just to stay up to date.

Over the years electronic circuits have grown in size and complexity. If many moons ago you could produce a project with a handful of transistors and gates, today you resort to some special IC that requires a pile of datasheets to be read before you can even get started. Still, that does not appear the main reason for many electronics fans, especially the older ones, for feeling lost in the proverbial woods.

In reality the biggest stumbling block appears to be the acute combination of hardware and software. Many electronics engineers, from the very beginning, have difficulty with programming and prefer to work with components only. For the younger generation, the opposite applies: they are totally at ease with programming, but find hardware design a difficult affair to say the least. Not surprisingly, today's electronic circuits invariably seem to combine software and hardware, forcing you to be well versed in both disciplines!

To suit both newcomers and old hands, old and young, subscribers and non-subscribers, this month we kick off a course in basic electronics that goes back to the roots. Like: how to dimension simple basic circuits; or how do certain components work? Furthermore, in each instalment we show a piece of software that allows the subject discussed to be put into practice using a microcontroller rather than discrete parts. Hopefully the LEDs covered this month go down well “for starters”.

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 DesignSpark chipKIT™ Design Challenge

A new global electronics design competition brought to you by Circuit Cellar, Elektor and RS Components.

16 Wideband Lambda Probe Interface

Link a lambda probe interface IC with a microcontroller and you have a stand-alone oxygen level meter for exhaust gases.

24 Audio DSP Course (7)

This month we use our knowledge and hardware to build a digital peak level meter with some pretty advanced features.

32 Grid Frequency Monitor

The frequency of the AC power grid is usually assumed to be very accurate and stable, but is it? Let's find out.

36 Scilab

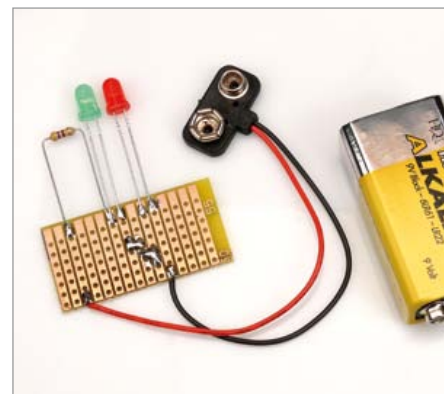
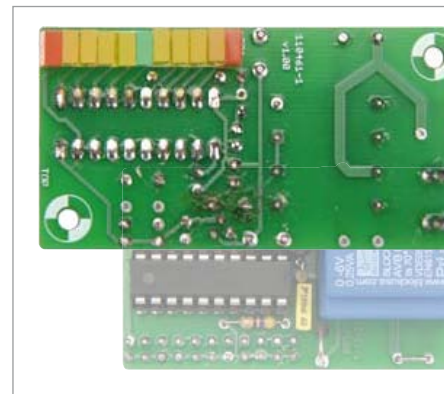
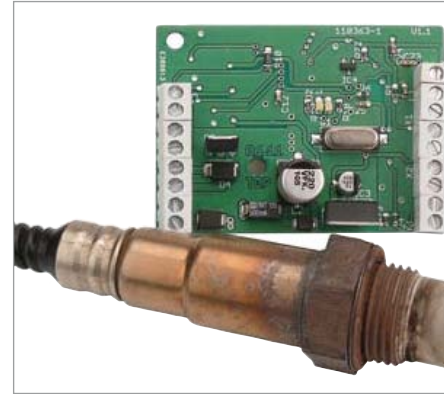
Introducing numerical calculation software for engineers into simulation and modelling.

43 E-Labs Inside: Bat—batter—best!

The operation of the Elektor Bat Detector can be substantially improved by using directive microphones.

44 E-Labs Inside: Radiation Meter: mounting the sensor

Here we elaborate on methods and materials to ensure the best operation of the meter for alpha, beta and gamma radiation.



CONTENTS

Volume 38
January 2012
no. 421



16 Wideband Lambda Probe Interface

This design marries a lambda probe interface IC with a microcontroller to produce a stand-alone lambda measurement device. Built-in self calibration gives hassle-free setup and ensures measurement accuracy. This flexible unit outputs the lambda value both as an analogue voltage level and as digital values using a standard serial interface.



32 Grid Frequency Monitor

The monitoring device described here detects tiny deviations in frequency with a range of just ± 0.2 Hz, allowing you to keep an eye on the load on the AC power grid from any convenient socket. An array of LEDs is used for the readout.



48 Ultra-accurate DSP-based DCF77 Timecode Receiver

To extract the highest possible accuracy from the German DCF 77.5 kHz time-code broadcast this project uses DSP algorithms running on a low-cost dsPIC33 microcontroller to filter and demodulate both the AM and phase modulated signals, while also producing a very stable 10 Hz carrier-locked reference clock output.



60 Electronics for Starters (1)

In this series we get back to basics, and in electronics the basics are undisputedly analogue. However, we realise that many beginners are interested in digital technology too, so a microcontroller circuit is also included in the course material. We kick off with LEDs and diodes.

46 E-Labs Inside: Debugging the debugger

There's an issue with the Microchip ICD3 In Circuit Debugger 3 (ICD 3). Here's how we fixed it.

48 Ultra-accurate DSP-based DCF77 Timecode Receiver

This project uses DSP algorithms running on a dsPIC33 micro to extract time signals with extreme accuracy.

54 Here comes the Bus! (11)

This month we discuss new PC software that simulates up to three freely programmable bus nodes.

60 Electronics for Starters (1)

Welcome to our new course! We kick off with LEDs and diodes.

64 Wavelet Analysis

Here we examine how Mikroelektronika's PIC32 development system can be used to analyse wavelet files.

70 Time/Interval Meter with ATtiny

Very little is required in terms of hardware to make an accurate time/interval meter.

73 Hexadoku

Elektor's monthly puzzle with an electronics touch.

74 Retronics: Philips 'Colette' Portable Radio (1956)

Series Editor: Jan Buiting

77 Gerard's Columns: Product Development

From our monthly columnist Gerard Fonte.

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. Analogue and digital; practical and theoretical; software and hardware.



**ANALOGUE • DIGITAL
MICROCONTROLLERS & EMBEDDED
AUDIO • TEST & MEASUREMENT**

Volume 38, Number 421, January 2012 ISSN 1757-0875

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

Publishers: Elektor International Media, Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, England. Tel. (+44) 208 261 4509, fax: (+44) 208 261 4447 www.elektor.com

The magazine is available from newsagents, bookshops and electronics retail outlets, or on subscription. Elektor is published 11 times a year with a double issue for July & August.

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

International Editor:
Wisse Hettinga (w.hettinga@elektor.nl)

Editor: Jan Buiting (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.

Editorial secretariat:
Hedwig Hennekens (secretariat@elektor.com)

Graphic design / DTP: Giel Dols, Mart Schroijen

Managing Director / Publisher: Don Akkermans

Marketing: Carlo van Nistelrooy

Subscriptions: Elektor International Media, Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, England. Tel. (+44) 208 261 4509, fax: (+44) 208 261 4447 Internet: www.elektor.com/subs

Elektor eC-reflow-mate

New



Professional SMT reflow oven with unique features

The eC-reflow-mate is ideal for assembling prototypes and small production batches of PCBs with SMD components. This SMT oven has a very large heating compartment, which provides plenty of space for several PCBs. The accompanying PC software allows you to monitor the temperature curves of all sensors precisely during the soldering process, and it enables you to modify existing temperature/time profiles or create new ones.



Special features:

- Optimal temperature distribution thanks to special IR lamps
- Drawer opens automatically at end of soldering process
- Glass front for easy viewing

Technical specifications:

- Supply voltage: 230 V / 50 Hz only
- Power: 3500 W
- Weight: approx. 29 kg
- Dimensions: 620 × 245 × 520 mm (W × H × D)
- Heating method: Combined IR radiation and hot air
- Operation: Directly using menu buttons and LCD on oven
- Remotely using PC software and USB connection
- Temperature range: 25 to 300 °C
- Maximum PCB size: 400 × 285 mm
- Temperature sensors: 2 internal and 1 external (included)

Price:

£2170.00 / € 2495.00 / US\$3495.00
(plus VAT and Shipping)

Further information and ordering at

www.elektor.com/reflow-mate

Email: subscriptions@elektor.com
Rates and terms are given on the Subscription Order Form.

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

Distribution:
Seymour, 2 East Poultry Street, London EC1A, England
Telephone: +44 207 429 4073

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

Email: t.vanhoesel@elektor.com
Internet: www.elektor.com
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, 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

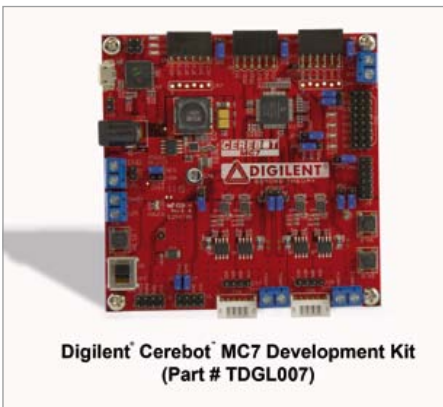
system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The submission of designs or articles implies permission to the Publisher 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 to them.

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

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

Embedded motor control Cerebot™ MC7 Development Kit for academia and hobbyists

Microchip Technology Inc., announced the availability of a Microchip dsPIC33 Digital Signal Controller (DSC)-based development kit. The Digilent® Cerebot™ MC7 Development Kit addresses the growing interest in embedded motor control from the academic and hobbyist markets, and is ideal for learning about microcontrollers and solving real problems. The kit includes a demonstration board that provides four half-bridge circuits, eight RC servo motor connectors, the ability to use Digilent Pmod™ peripheral modules, and an integrated programming/ debugging circuit that is compatible with the free MPLAB® IDE. Example applications include university embedded-systems and communications classes, senior capstone



Digilent Cerebot MC7 Development Kit (Part # TDGL007)

projects, and numerous other academic and hobbyist projects.

A video demo of the kit can be viewed on YouTube (link below).

The Cerebot MC7 board features four half-bridge circuits that are rated for 24 V at up to 5 A. These half bridges can be used to control two Brushed DC motors, two bipolar stepper motors, one brushless DC motor, and one uni-polar stepper motor. An onboard 5 V, 4 A switching regulator with an input voltage up to 24 V simplifies operation of the board, enabling it to operate from a single power supply in embedded applications such as robotics. The onboard dsPIC33 DSC features 128 KB internal Flash program memory and 16 KB internal SRAM, as well as numerous on-chip peripherals, including an advanced 8-channel motor-control PWM unit, an enhanced CAN controller, two Serial Peripheral Inter-

faces (SPIs), timer/counters, serial-interface controllers, an Analog-to-Digital Converter (ADC), and more. The Cerebot MC7 board combines two push buttons and four LEDs for user I/O, as well as connections for two I2C™ busses, one of which contains an integrated serial EEPROM device.

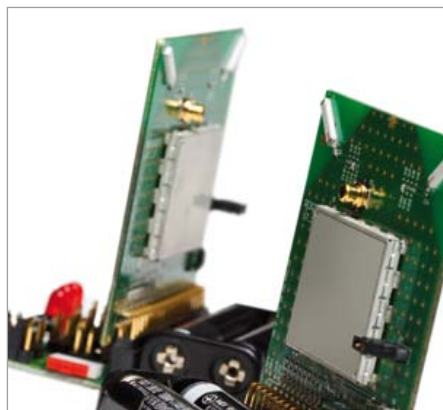
Video: <http://www.microchip.com/get/8HF8>

Buy: <http://www.microchip.com/get/DUMT> (110698-VIII)

Low-power RF Transceiver for cost-sensitive consumer applications

Atmel® Corporation's new AT86RF232 RF transceiver supports the high-volume consumer markets in the 2.4GHz ISM (Industrial, Scientific and Medical) band. The Atmel AT86RF232 transceiver includes all the necessary features to support the latest wireless applications in the consumer segment including excellent RF performance, lower power consumption, high-link budget and antenna diversity. The new devices also support ZigBee® RF4CE, a specification designed to control a wide range of wireless consumer products including remote controls for home entertainment devices, human interface devices such as mice and keyboards, and 3D glasses.

Applications in the high-volume portable consumer segment, including key fobs, remote controls for toys and game con-



soles, require very low-power transceivers to extend the overall battery life of the device. The Atmel AT86RF232 transceiver offers 50 percent lower power than the competition. The Atmel transceiver supports automatic antenna diversity to improve RF performance and link reliabil-

ity. Additionally, the transceiver includes onboard AES encryption for secure wireless end-to-end communication.

The new Atmel transceiver offers a voltage range of 1.8 V to 3.6 V, -100 dBm in sensitivity and an output power of 3dBm for extended battery life in various applications. The device also offers a receiver current consumption of 11.8 mA, transceiver current consumption of 13.8mA, antenna diversity and AES encryption. All these features are important to offer a robust wireless transceiver for high-volume consumer applications.

The Atmel AT86RF232 transceiver is available with the REB232ED-EK evaluation kit which includes two AT86RF232 radio evaluation boards, combined with an Atmel ATXmega256A3 microcontroller. These boards come with free downloadable evaluation software on the Atmel website. These evaluation boards and transceiver are also ideal for designers moving towards ZigBee RF4CE and ZigBee Remote Control profiles.

www.atmel.com/AT86RF232 (110698x-IX)

RF6555 2.0 V to 3.6 V, 2.4 GHz front end module

RFMD's new RF6555 integrates a complete solution in a single Front End Module (FEM) for ZigBee® and Wi-Fi applications in the 2.4GHz to 2.5GHz band. This FEM integrates the PA plus harmonic filter in the transmit path and an internally integrated LNA with bypass mode. The

RF6555 provides a single balanced TDD access for Rx and Tx paths along with two ports on the output for connecting a diversity solution or a test port. The RF6555 also has current consumption for ZigBee applications enabling extended battery life with only 70mA at rated power.

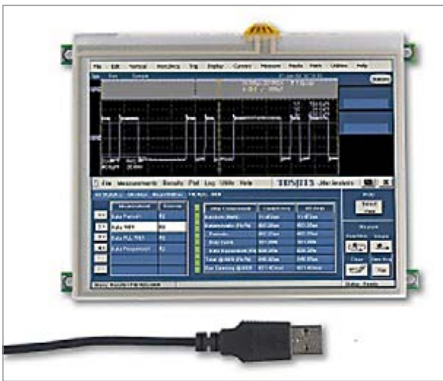
Applications include ZigBee® 802.15.4 Based Systems for remote monitoring and control, AA cell battery operated equipment, and Wi-Fi 802.11b/g.

www.rfmd.com (120031-1)



New ezLCD smart touch LCD for 5.6" embedded display applications

EarthLCD.com, recently announced their new ezLCD-005 graphic display module. Its all-in-one design unites a color LCD, touchscreen, control electronics, memory and I/O, with an easy to use, command driven programmable firmware environment. Helping engineers minimize development time and reduce hardware costs, while speeding time to market for applications requiring a color touchscreen display, the ezLCD-005 proves to be an effective solution as a graphical user interface (GUI).



The ezLCD-005 features:

- 5.6 Inch color TFT LCD with LED Backlight
- 320 x 234 resolution, supporting 65,536 colors
- 330 Nit brightness
- 300 to 1 contrast ratio
- Integrated 4 wire resistive touchscreen
- Intelligent control Module with 32 Bit ARM microprocessor
- 1 Mbyte serial flash memory
- SD/MMC card slot for memory expansion up to 4 GB
- USB, SPI, and RS232/TTL interfaces
- +5.0V supply voltage
- 0 to 60°C operating temperature
- Command driven programmable firmware environment
- EarthSEMPL (simple embedded macro programming language)
- Works with any microcontroller

The ezLCD-005 is claimed to be the OEM's fastest way to integrate a 5.6 inch color LCD with touchscreen into their new or existing embedded application, allowing engineers to focus on interfacing the ezLCD-005 into their product design instead of sourcing all the individual components necessary to accomplish their goal. Plus, not having to manage the supply chain for all those

Pololu: dual VNH5019 motor driver shield for Arduino

Pololu announces the release of the dual VNH5019 motor driver shield for Arduino, an easy way to control up to two high-power DC motors with an Arduino or Arduino-compatible device.

The shield's twin robust VNH5019 motor drivers operate over a wide 5.5 to 24 V range and can each deliver a continuous 12 A (30 A peak) to an independently controlled motor, or the driver outputs can be combined to deliver a continuous 24 A (60 A peak) to a single motor. The motor drivers also offer current-sense feedback and accept ultrasonic PWM frequencies for quiet operation.

This motor driver is intended for a wide range of users, from beginners who just want a plug-and-play motor control solution for their Arduinos (and don't mind a little soldering) to experts who want to directly interface with ST's motor driver ICs. The Arduino pin mappings can all be customized if the defaults are not convenient, and the VNH5019 control lines are broken out along the left side of the board for general-purpose use without an Arduino. This versatility, along with an option to power the Arduino directly from the shield, sets this board apart from similar competing motor shields.

The shield (item #2502) is available for \$59.95. For more information, including a detailed user's guide and an Arduino library for this shield, please visit the url below.

www.pololu.com/catalog/product/2502 (120031-II)



unique parts saves them time, money and reduces time to market.

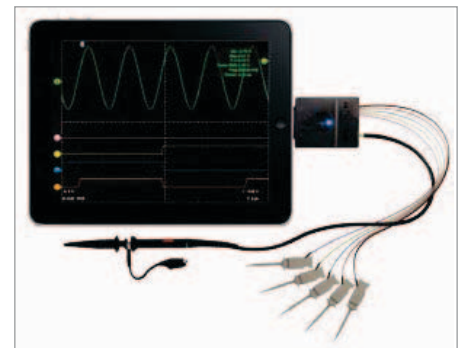
The ezLCD-005 was developed for industries such as Industrial Control, Instrumentation, Test Equipment, Kiosk, Point of Sale, Medical, Automotive, and others where products might require a color LCD with touchscreen to be utilized as a user interface. Pricing for the ezLCD-005 starts at \$330.00 ea.

store.earthlcd.com/ezLCD-005 (120031-III)

Oscium mixed signal oscilloscope for iPad adds FFT, data logging, and config saving

Oscium yet again enhances the iMSO app, unlocking additional value for customers. Since releasing iMSO-104 in the Apple App Store, the following have been added:

- Ability to save configurations – FAE's and on-the-go engineers can now save their configurations
 - Data Logging – record and export history via *.csv for post-processing
 - FFT – enables a different view of the signal, a view from the frequency domain
- Today's release marks another level of con-



tinuing upgrades to the iMSO interface. The performance of FFT has been improved by increasing the resolution of the FFT display, providing an even clearer diagnostic picture. The performance of rolling mode has also been improved with better resolution in time scales greater than 40 ms.

iMSO version 2.1.0 is available to download free in the Apple App Store. The iMSO app is compatible with all generations of iPhone, iPod touch, and iPad devices running iOS version 3.1.3 or higher. It is made for: iPod touch (1st, 2nd, 3rd, and 4th generation), iPhone 4S, iPhone 4, iPhone 3GS, iPhone 3G, iPhone, iPad 2, and iPad. iMSO-104 hardware can be purchased for \$297.99 from Oscium directly or from one of their partners.

www.oscium.com (120031-V)

Cypress's PSoC[®] 3 device powers hot new GuitarJack Model 2

Cypress Semiconductor Corp. (announced that Sonoma Wire Works has selected the PSoC[®] 3 programmable system-on-chip for its new GuitarJack Model 2. The GuitarJack Model 2 is a digital audio interface accessory product that connects a musical instrument, microphone, or audio hardware with iPod touch, iPhone or iPad. The PSoC 3 device in the GuitarJack seamlessly manages the digital audio interface and handles Apple's proprietary MFi protocol to communicate with iOS devices.

GuitarJack Model 2 is optimally designed to work with Sonoma Wire Works iOS apps including GuitarTone, FourTrack, StudioTrack, and TaylorEQ, as well as Apple's GarageBand. It streams digital audio out of and into Apple's latest iOS devices, including iPad 2, iPad, iPhone 4, and iPod touch (2nd, 3rd, and 4th generation). It offers stereo recording and simultaneous voice and instrument recording via Apple's proprietary connector. Since GuitarJack Model 2 is fully synchronous between iOS and PSoC 3, it maintains high quality audio streaming with no dropped audio packets. It can be powered directly by the iOS device and does not need its own power supply or batteries. More information is available at the Sonoma Wire Works website.

The PSoC 3 device manages the interface between GuitarJack Model 2 and any iOS device via Apple's proprietary dock connector. With Cypress's patent-pending clock synchronization and recovery scheme, PSoC 3 also delivers the highest quality digital audio with minimal external components while supporting multiple audio sample rates. Details on Cypress's MFi (Made for iPod, iPhone and iPad) solution offering are available at www.cypress.com/go/MFi. More information about Cypress's clock synchronization and recovery scheme is available at www.cypress.com.



com/?docID=25374. "It's an exciting time for the music creation and instruments market," said Leon Tan, marketing manager for Cypress's MFi solutions. "With a large installed base of iOS devices in the market today, and growing at an unprecedented rate, Sonoma Wire Works is able to quickly capitalize on that with its unique ability to create high-quality hardware and software audio products that the music industry demands. We're excited that the PSoC 3 technology advantages are well suited for this market."

"PSoC 3's unique, patent-pending technology enables digitally streaming audio between iOS devices and connected accessories," said Gahan Richardson, vice president of PSoC platform products for Cypress. "This enables Sonoma Wire Works to deliver an exciting, versatile product with great sound for music enthusiasts."

"PSoC 3's low power; flexible architecture; and high-quality USB audio solution helped Sonoma Wire Works upgrade GuitarJack from its predecessor (Model 1 to Model 2)," said Douglas Wright, Founder and President of Sonoma Wire Works. "Together with GuitarTone, Model 2 provides the complete package for musicians to create music here, there and everywhere."

www.sonomawireworks.com/guitarjack
www.cypress.com/psoc
www.cypress.com/psoctraining
 (120031-IV)

HADES hi-temp hi-rel isolated gate driver for high density power converters

CISSOID launched HADES, claiming it's the first isolated gate driver solution designed to drive high temperature power transistors, specifically (but not exclusively) Silicon carbide (SiC) and Gallium nitride (GaN) fast-switching devices.

With HADES[®], system engineers can develop power converters that are 5 times smaller and lighter than before, with better efficiency. They will also get power converters able to operate in high temperature ambience if required. No matter what the ambient temperature is, the lifetime of the system will be an order of magnitude longer than traditional solutions.

HADES has been designed to drive seamlessly Silicon Carbide (SiC) power tran-



sistors, which have low switching losses. HADES can switch them at high frequencies, which means smaller and lighter passive and magnetic components. Furthermore, thanks to its ability to sustain high temperatures, HADES can be located next to the power transistors which reduces parasitic capacitances and inductances, and that further improves the associated losses and delays in the system.

HADES is a reference design and an Evaluation Board delivered with full documen-

tation. It can drive two SiC MOSFET power switches on a DC bus voltage up to 1200 V. The Reference design is scalable up to ± 20 A gate current, while the Evaluation Board features ± 4 A. A specific board flavor for normally-On JFETs will also be available, and other types of switching devices (normally-On/Off JFETs, BJTs and IGBTs) can be supported with minor changes.

As an example, HADES operation and performances were demonstrated in a 3 kW Buck DC-DC converter, driving SiC MOSFETs, at 175°C ambient and switching at 150 kHz, with rise times of less than 25 ns. In these operating conditions, HADES, which has been designed for high dV/dt immunity (50 kV/ μ s) and IC junction temperatures up to 225°C, runs with comfortable safety margins.

In terms of efficiency, the combination of HADES with the newest SiC switches in advanced power converter topologies will bring efficiencies in excess of 98%, even at

switching frequencies above 100 kHz. HADES™ gate driver is the ideal solution for high-power converters such as motor drives, battery chargers and power distribution used in applications like railway, aircrafts, renewable energies and hybrid / electric vehicles. It delivers high power density, simplified cooling and high reliability. The fast-switching ability of HADES™ plus the fact it can operate reliably at the same temperature as the switches (200°C junction and above) makes it a solution of choice for the new generation of Intelligent Power Modules (IPM).

www.cisoid.com (120031-VI)

New range of high specification AT-cut crystals

IQD Frequency Products' new range of high specification crystals is aimed at specialist applications such as military, radar and pagers. Manufactured at its IQD FOQ division in Germany, the crystals are available in a variety of package styles including UM1, HC49, HC43CW (TO8) and HC45CW. For the highest possible 'Q' factor, customers should specify the Cold Weld (CW) packages. The 'Q' factor is a measurement of the loss of energy within the quartz. This is effected by the quality of the blank, surface finish (lapping), mounting technique and sealing. The higher the 'Q' factor the better the crystal. The higher the frequency the more important the 'Q' factor is to the design engineer. The maximum attainable short term stabil-



ity of a crystal also depends on the 'Q' value. A wide frequency range is available including 10 to 42MHz at fundamental mode, 10 to 125 MHz at 3rd Overtone and 70 to 175 MHz at 5th Overtone. This is pushing the design limits of quartz crystals due to the thinness of the blank that becomes unworkable beyond this.

The new range offers exceptionally low aging down to 0.1 ppm per year and tight frequency tolerance @ 25 degrees C down to ± 3 ppm. This compares to standard crystals where typical figures would be 3 ppm per year aging and ± 10 ppm frequency tolerance. Temperature stabilities down to 4 ppm @ 0 to 70 degrees C and 12 ppm @ -40 to 85 degrees C can be specified depending upon customers applications. Phase noise is critical in many applications and these high specification crystals offer excellent performance in this respect, including at high frequencies up to 100 MHz at 5th overtone.

IQD offer an express manufacturing service for these parts as short as 3 days dependent upon the specification required.

www.iqdfrequencyproducts.com (120031-VII)

'Mini8' Ballast Control IC

International Rectifier's new IRS2526DS 'Mini8' is a compact fluorescent lamp (CFL) ballast control IC that offers full programmability and a high degree of accuracy and control for all lamp types. Available in an 8-pin SO-8 package, the new feature-rich IC reduces component count, simplifies circuit design and increases efficiency in a compact footprint.

The IRS2526DS features a 600 V half-bridge control circuit working at 50 percent duty-cycle and variable frequency for driving the resonant mode lamp output circuit. The high accuracy oscillator is controlled by a single analog-to-frequency input pin used to set the different operating frequencies of the ballast. Complete fault protection circuitry is also included for protection against such conditions as mains interrupt



or brown-out, lamp non-strike, lamp filament failure and end-of-life. The new device also incorporates an internal frequency dither to reduce conducted EMI, ignition control to reduce inductor size, and end-of-life detection.

The IRS2526DS is the third generation ballast IC from IR and utilizes the proven technology also featured in the previously released IRS2580DS 'Combo8' that combines a full featured fluorescent ballast with power factor correction (PFC) controller in a compact 8-pin package. The new IRS2526DS is targeted at applications that do not require PFC or utilize an external PFC controller.

A datasheet and application note are available on the International Rectifier website. A reference design, the IRPLMB7E 220VAC/50Hz, 18 W TCL lamp, featuring the IRS2526DS and Ballast Design Assistant (BDA) V5.0 design software are available on request. The new devices are lead free and RoHS compliant.

www.irf.com (120031-VIII)

DC-DC switching charger ICs for single cell Li-Ion batteries in mobile devices

austriamicrosystems has announced two new battery charger ICs for Li-Ion batteries in mobile devices. The AS3610/11 step-down DC-DC chargers offer fast and highly efficient charging of Li-Ion batteries in mobile devices with up to 1.25 A output

Advertisement



**PCBs
Muuuuch Cheaper...**

No-frills policy

16.94 EURO*

5 pcs, 100 mm x 100 mm
*per piece, incl. VAT (21%)
+ shipping costs e. g. Germany 10.71 EURO



www.jackaltac.com

Under £350 USB mixed-signal oscilloscope

The PicoScope 2205 MSO gives you a two-channel oscilloscope combined with a 16 channel logic analyzer, all in one compact, portable USB instrument. Now you can view analog waveforms and digital data on the same screen with the efficient and easy-to-use PicoScope software.

The analog bandwidth is 25 MHz and the digital channels can accept signals as fast as 100 MHz. Maximum sampling rate is 200 MS/s. The PicoScope 2205 MSO is suitable for general-purpose analog and digital circuit design, testing and troubleshooting. As it's USB-powered, there is no AC adapter to carry; just plug it into your PC or laptop and start the software. The 48k-sample buffer is large enough to store multiple captures in rapid sequence as little as 2 microseconds apart. The PicoScope software, included, delivers a high-resolution, uncluttered display and a range of advanced signal processing features: spectrum analyzer, automatic measurements with statistics, channel math, reference waveforms, multiple scope and spectrum views, I²C, UART, SPI and CAN bus serial decoding, XY mode, advanced triggers, mask limit testing, and color persistence display modes.

The 16 digital inputs can be displayed individually or in arbitrary groups labeled with binary, decimal or hexadecimal values. A separate logic threshold from -5 V to +5 V can be defined for each 8-bit input port. The digital trigger can be activated by any bit pattern combined with an optional transition on any input. Finally, analog and digital triggers can be combined using Boolean logic to enable complex mixed-signal triggering. The Software Development Kit (SDK), also included, allows you to control the new scopes using your own software. The SDK and PicoScope are compatible with Microsoft Windows XP, Vista and Windows 7. Example programs in C, Excel and LabView are included. The new PicoScope 2205 MSO is available now, priced at only £349 for the oscilloscope alone or £399 as a kit with two passive x1/x10 probes, a logic cable and test hooks. A generous 5-year warranty is included.

<http://www.picotech.com/mixed-signal-oscilloscope.html> (120031-XII)



sion. The AS3610/11 is available in a small 3x3 mm MLPD14 package, and operates from a supply of 2.7 V to 5.5 V over a temperature range of -40 °C to 85 °C. A WL-CSP package option is available for PCB space-critical applications.

The AS3610/11 switch mode chargers are available now and are priced at \$1.20 in 1000-piece quantities. A demo board is also available to reduce development time.

www.austriamicrosystems.com (120031-X)

Low-Speed CAN and LIN interfaces for NI CompactDAQ

National Instruments has introduced the C Series NI 9861 CAN interface and NI 9866 LIN interface, the newest modules in the NI-XNET family of products and the first low-speed CAN and LIN modules that integrate with the entire NI CompactDAQ platform. As part of the NI-XNET family, the new modules provide engineers with productivity advantages such as hardware-accelerated messaging and onboard processing. The single-port, low-speed C Series NI 9861 CAN interface module features integrated CAN database support for importing, editing and using signals from FIBEX, .DBC and .NCD files. It is capable of 100% bus load communication up to 125 kbit/s without dropping any frames. The NI 9866 LIN module is also a single-port interface with integrated support for importing and using signals from LDF databases along with master/slave support and hardware-timed scheduling for master tasks. It is capable of 100% bus load communication up to 20 kbit/s without dropping any frames.

Engineers can use the new CAN and LIN modules with the same NI LabVIEW or ANSI C/C++ software code on a variety of platforms including NI CompactDAQ, CompactRIO, PXI and PCI. Project reuse saves time as the same applications can be used, for example, in labs with PCI; in manufacturing end-of-line tests with PXI; in portable in-vehicle communication settings with NI CompactDAQ; and in headless in-vehicle logging with CompactRIO.

With native support in NI VeriStand real-time test development software, the new modules are ideal for real-time automotive testing applications, including hardware-in-the-loop simulation and test cell appli-

current. The ICs include numerous safety and protection features, internal current measurement, and USB Host/OTG (on the go) boost mode operation. The AS3610 DC-DC charger provides an I²C interface for external control and the AS3611 operates independently.

The performance and features of the AS3610/11 make the charger ICs very well suited for applications that are powered by

one Li-Ion battery. Such applications include high-end blood glucose meters, remote controls, GPS outdoor navigation or tracking equipment, mobile phones, e-dictionaries and e-book readers.

The AS3610/11 switching charger ICs were designed with inputs from the handset and portable markets to use a high efficiency switch mode charger with minimum ripple. Implementation is simplified with most options and features controlled by pin strapping or OTP. These features include current limit set, charging current, OTG boost, and more. The AS3610 step-down charger also supports 900mA input current limitation for USB 3.0.

Other features austriamicrosystems' AS3610/11 switching chargers include: 22 V over-voltage protection; reverse polarity protection; chip and battery temperature supervision, and charger time-out supervi-





cations. Both modules support synchronization and triggering with other CompactRIO and NI CompactDAQ modules. NI CompactDAQ offers a platform for portable in-vehicle network communication, in-vehicle logging and basic automotive electronics communication in a USB, Wi-Fi or Ethernet form factor.

The NI-XNET family provides a common programming interface for multiple automotive networks such as CAN, LIN and FlexRay. With NI-XNET interfaces, engineers can develop applications for prototyping, simulating and testing these networks faster and more easily in LabVIEW and LabVIEW Real-Time software as well as ANSI C/C++. The interfaces combine the performance and flexibility of low-level microcontroller interfaces with the speed and power of Windows and LabVIEW Real-Time OS development. Engineers can easily integrate them with desktop real-time PCs and real-time PXI systems.

www.ni.com/can www.ni.com/lin
(120031-XVI)

First ARM-Cortex™ microcontrollers with on-board 10/100 Ethernet, CAN2.0B and USB

Toshiba Electronics Europe's next series of high-performance, low-power 32-bit ARM Cortex™-M3 microcontrollers reportedly are the first to combine Ethernet, CAN and USB Host and Device connectivity in a single IC. Potential applications for the highly integrated TMPM369Fxxx family include industrial control systems, barcode readers, motion control, home appliances and solar inverters.

Based around an ARM Cortex-M3 core running at 80 MHz, the four microcontrollers

in the TMPM369Fxxx series integrate single-channel CAN2.0B, a full-speed USB Host controller, a full-speed USB device controller and a 10/100BASE single-channel Ethernet MAC. Two independent analogue-to-digital converters (ADCs) with conversion times of 1 μ s – or 0.5 μ s in interleaved mode – meet the requirements of barcode readers and other applications requiring ultra-fast conversion. In addition, each of the devices features a special Multi-Purpose Timer (MPT). This MPT combines three-phase PWM control with an ADC trigger making the new devices ideal for motor control applications. The TMPM369Fxxx series offers on-board ROM options of either 512 Kbyte or 256 Kbyte using Toshiba's NANO FLASH technology, which runs at 80 MHz without wait states and allows for very high-speed programming. Integrated RAM of up to 128Kbyte provides significant on-board capacity for key IP, while an industry-leading 32-channel DMA controller increases the



overall system performance dramatically. All of the new microcontrollers feature two 10-bit digital-to-analogue converter (DAC) channels and a 2-channel encoder signal input for motor control. Also on-board are a Real Time Clock and an Oscillation Frequency Detector (OFD). The latter provides hardware monitoring of the CPU clock in accordance with the IEC60730 (Class B) safety standard for home appliances. Power on Reset (PoR) functionality is provided as standard.

Featuring on-chip regulators, TMPM369Fxxx microcontrollers are designed for operation from a single 2.7 V to 3.6 V power supply – or 3.0 V to 3.6 V when USB is in operation. Clock gearing functionality and 'IDLE', 'STOP1' and 'STOP2' standby modes help to keep power consumption to a minimum.

As well as the CAN, USB and Ethernet functionality the new microcontrollers also have a variety of other interfaces to further speed embedded system design and reduce application component count. These include a 3-channel synchronous serial interface (SSP), 6-channel UART/general-purpose serial I/O (SIO) and three I²C channels. The availability of various Starter Kits and Software packages supports a fast prototyping. Toshiba's TMPM369Fxxx microcontrollers are available in LQFP-144 and FBGA-176 package options.

www.toshiba-components.com (120031-IX)

'Electric Sheep' Android app development kit

SparkFun Electronics' new product, aptly named the Electric Sheep, is designed to take advantage of the Android system's open accessory protocol. By communicating via USB, Electric Sheep gives users complete dynamic access to the phone's systems and enables the easy creation of custom applications and accessories such as controls for an autonomous vehicle or coupling your phone with a microcontroller such as an Arduino. These boards are not restricted to only Android phones, but can be used on any platform with the Android operating system and a USB port.

"The Electric Sheep is a product we are really excited to announce," said SparkFun Director of Marketing AnnDrea Boe. "This product represents months of prototyping and design by the SparkFun engineers and will give its users free reign to create accessories for the Android system."

On the technical side, the Electric Sheep is based on the ATmega2560 microcontroller and bootloader and features a USB-host connector on-board for quick connection to Android devices. The board allows for the creation of accessories for Android using the Arduino IDE and HandBag and is Arduino-shield compatible. Even if you don't have an Android device, the Electric Sheep can double as a development platform with all the functionality of the host microcontroller and a USB shield. The Electric Sheep is priced at \$79.95 and is now available on SparkFun's website.

www.sparkfun.com (120031-XIII)



DesignSpark chipKIT™ Design Challenge



Apply the chipKIT™ Max32™ development kit and the award-winning DesignSpark PCB software tool to create environmentally friendly applications

By Ian Bromley (UK)

Towards the end of 2011, the world's population passed the seven billion mark and UN estimates are that it could be closer to nine billion within a few decades. Add to that the increasing energy demands to meet growth required by major emerging economies, and there is no doubt that meeting the world's energy needs while also looking after our environment is one of the great challenges of our time. In the drive to achieve greater energy sustainability, clearly the development of innovative new solutions are required, and I believe embedded electronics can make a contribution to meet this global challenge.

It isn't just about developing ultra low power electronic devices — in fact, arguably more important is increasing the energy efficiency of electronic systems. Maximizing power output and delivering the required performance while reducing the environmental and energy footprint is becoming a highly significant ratio. For example, in the embedded world, a 32-bit microcontroller is most likely to consume more power than an 8-bit device, but it also has the ability to perform very fast calculations and process algorithms that can significantly increase energy efficiency in motor-control applications. So, when we were approached last year by Elektor and Circuit Cellar to participate in a worldwide design competition, the choice was an easy one: it had to be about energy efficiency.

Officially launched in late November at Elektor Live! and offering total cash prizes of \$10,000, including a first prize of \$5,000, the DesignSpark chipKIT™ challenge is all about encouraging engineers, students and hobbyists to think about the use of power and develop solutions that will increase energy efficiency while also maintaining an eco-friendly footprint. The prize-winning application can be anything that reduces the energy footprint of a system. For example, it could be an energy-efficient battery charger, a controller for a windmill, or perhaps an energy-usage management device for use in the home.

Competition entrants are now developing energy-efficient and environmentally friendly applications based on the chipKIT™ Max32™ development platform from Digilent, which feature Microchip's 32-bit PIC32 microcontroller. The first 1,000 registered entrants have already received a complimentary chipKIT™ Max32™ development board.

The chipKIT™ Max32™ development platform is a 32-bit Arduino-

compatible solution that enables engineers, students and enthusiasts to easily and inexpensively integrate electronics into their projects. The chipKIT™ hardware is compatible with existing Arduino shields and applications, and can be developed using a modified version of the Arduino IDE and existing Arduino resources, such as code examples, libraries, references and tutorials. The chipKIT™ Basic I/O Shield is compatible with the chipKIT™ Max32™ board, and offers users simple pushbuttons, switches, LEDs, I²C EEPROM, I²C temperature sensor, and a 128 x 32 pixel organic-LED graphic display.

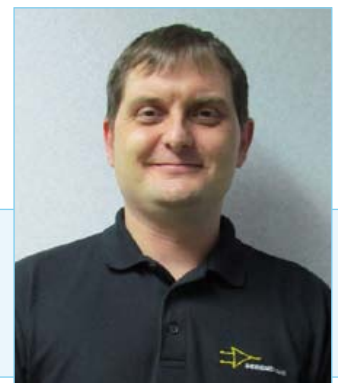
All entries must include an extension card developed using RS' free-of-charge DesignSpark PCB software tool with code compiled using Digilent's MPIDE software. The DesignSpark PCB software tool is unique in the industry, and since its launch in July 2010, more than 100,000 users have downloaded DesignSpark PCB, RS' professional-standard PCB design software, which has proven to be one of the most popular free PCB design software packages available. Thousands of users have contributed suggestions for its continued development via the DesignSpark community, which is available at www.designspark.com.

During the competition, which finishes at the end of March 2012, entrants are being strongly encouraged to engage and interact with other members of the online DesignSpark community by posting information on their projects, providing updates on progress, and sharing comments and ideas on their respective designs. Participants will automatically qualify for entry into bonus Community Choice Awards, in addition to admission into spot prize draws for the best collaboration to win vouchers exchangeable for products ordered from RS Components/Allied Electronics.

The competition entries will be judged on the level of energy efficiency and the quality of the extension card's PCB design. Entries are due on March 28, 2012 and the winners will be announced in April 2012.

(120020)

Further details and registration for the DesignSpark chipKIT™ challenge are available at: chipkitchallenge.com.



Ian Bromley is a Technical Marketing Engineer at RS Components and the Project Manager for the DesignSpark PCB software tool. Prior to working for RS, Ian worked for many years as a design support consultant with Texas Instruments, in addition to working as a field applications engineer immediately following his graduation in 1994 with an honours degree in microelectronic engineering.

01279 Credit Card Sales **467799**

Ho! Ho! Ho! Christmas 2011 is on it's way but **DON'T PANIC!!**
We have some fantastic gift ideas for young (and old) enquiring minds

Electronic Project Labs
An electronics course in a box! All assume no previous knowledge and require NO solder. See website for full details



30 in ONE - £19.95
Order Code EPL030KT



130 in ONE - £59.95
Order Code EPL130KT



300 in ONE - £79.95
Order Code EPL300KT



500 in ONE - £179.95
Order Code EPL500KT



Robot Sensor - £21.95
Order Code EPLR20KT



Digital Recording Laboratory - £34.95
Order Code EPLDRKT

Solderless Electronic Project Kits



AM / FM Radio Kit - £11.95
Order Code ERKAFKT



Short Wave Kit - £11.95
Order Code ERKSWKT



Crystal Radio Kit - £8.95
Order Code ERKCKT



Electronic Bell - £8.95
Order Code EAKEBKT



Electronic Motor - £8.95
Order Code EAKEMKT



Generator - £8.95
Order Code EAKEGKT



Room Alarm - £4.95
Order Code EAKRAKT



Hand Held Metal Detector - £9.95
Order Code ELMDX7



Metal Detector - £9.95
Order Code ELMDKT

Robot & Construction Kits
Future engineers can learn about the operation of electronics, robotics and transmissions systems.



Tyrannomech - £14.95
Order Code C21-601KT



Robotic Arm - £49.95
Order Code C9895



Crawling Bug with Case - £21.95 - Order Code MK165



Running Microbug - £12.95
Order Code MK127KT



Trainmech—£14.95
Order Code C21-606KT

Festive Electronic Project Kits



Riding Santa - £17.95
Order Code MK116KT



60 LED Multi-Effect LED Star - £17.95
Order Code MK170KT



Musical LED Jingle Bells - £21.95
Order Code MK176KT



Flashing LED Christmas Tree - £7.95
Order Code MK100KT

See our website for special offers and even more great gift ideas!



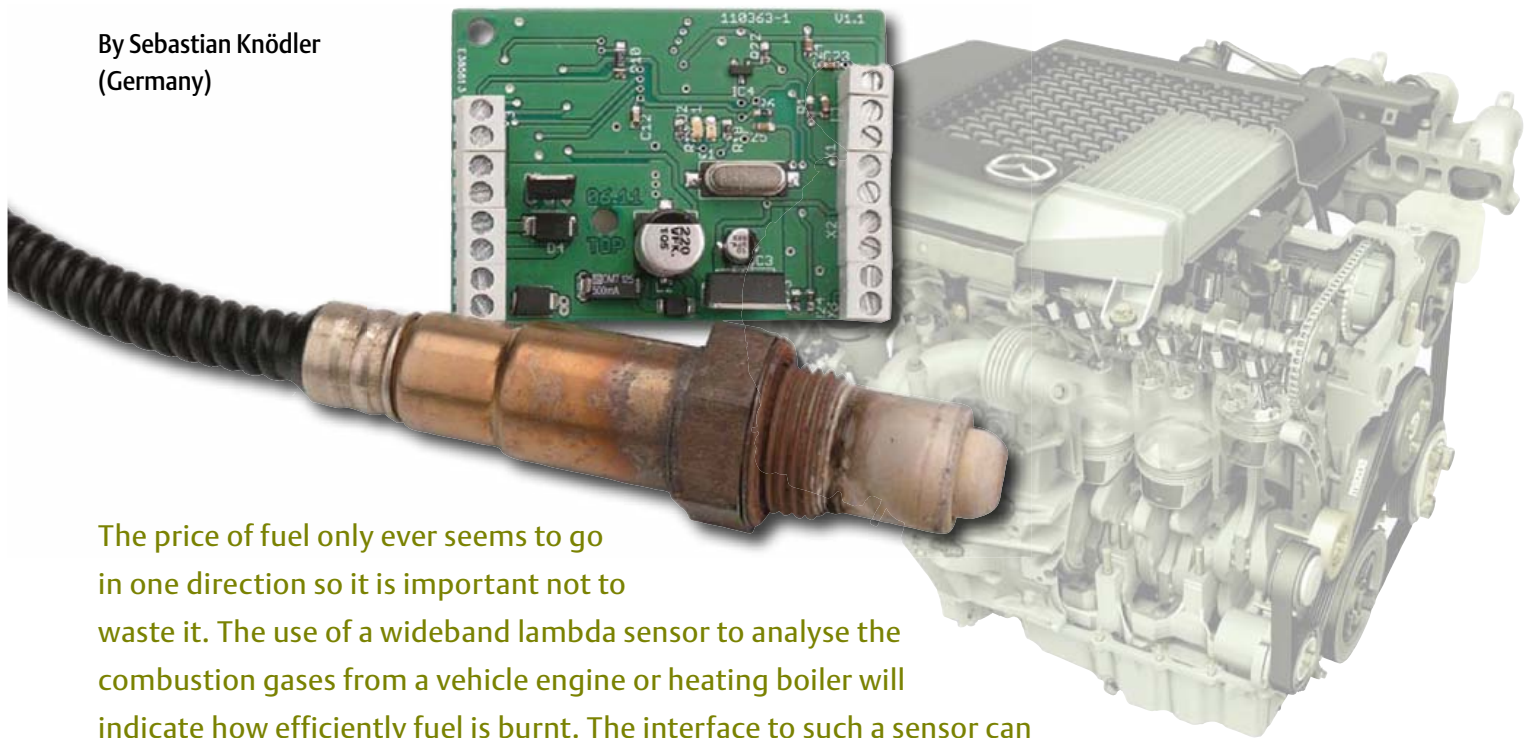
www.QuasarElectronics.com

Secure Online Ordering Facilities • Full Product Listing, Descriptions & Photos • Kit Documentation & Software Downloads

Wideband Lambda Probe Interface

Test exhaust oxygen levels

By Sebastian Knödler
(Germany)



The price of fuel only ever seems to go in one direction so it is important not to waste it. The use of a wideband lambda sensor to analyse the combustion gases from a vehicle engine or heating boiler will indicate how efficiently fuel is burnt. The interface to such a sensor can however be quite complex. This design marries a lambda probe interface IC with a microcontroller to produce a stand-alone lambda measurement device. Built-in self calibration gives hassle-free setup and ensures measurement accuracy. This flexible unit outputs the lambda value both as an analogue voltage level and as digital values using a standard serial interface.

Measuring the amount of oxygen in exhaust gases gives an indication of how complete the combustion process proceeded. From the oxygen content the so-called lambda (λ) value can be calculated. Lambda is a ratio of the actual induced air mass divided by the theoretical air requirement. A lambda reading of 1.0 therefore indicates optimal combustion conditions where the air content

(oxygen in the volume of air) of the fuel/air mixture matches the requirements. A reading of $\lambda > 1$ indicates a 'lean' burn (i.e. too much air in the mixture) while a lambda reading of less than 1.0 indicates a 'rich' burn (too little air in the mixture). A lean mixture condition has more oxygen available than is required by the combustion process giving rise to molecular oxygen in

the exhaust gas. This combustion condition is of primary interest for operators of diesel motors and heating boilers (e.g. wood pellet or oil-fired) which do not operate correctly when the mixture is too rich. A rich mixture has too little air to allow all the fuel to be burnt so the excess unburnt fuel remains in the exhaust gas where it can be detected.

Elektor Products & Services

- PCB: # 110363-1
- Programmed controller: # 110363-41

- PCB artwork, PDF, free download: 110363-1.zip
- Project software, free download: 110363-11.zip

All products accessible via www.elektor.com/110363

Features

- Interface for wideband lambda probe.
- Suitable for wide range of Lambda values.
- Fault evaluation via diagnostics register.
- Digital output with RS232 interface.
- Output of lambda value and operational status.
- Analogue output 0 to 4 V.
- Automatic calibration.

Petrol engines operate most efficiently when they run on a slightly lean mixture ($\lambda = 1.1$ to 1.25), whilst for maximum power a slightly rich mixture ($\lambda = 0.9$ to 0.95) is required. A rich mixture also provides some degree of 'inner cooling' to the engine parts (unburnt fuel cools the combustion chamber, valves and other internal components).

The earlier narrow-band or two-state sensors are only able to indicate a lambda value in the region of $\lambda = 1$, they effectively operate as a switch, indicating if the lambda value is either greater than or less than 1. By contrast the more recent wide-band lambda sensors give a continuous reading of values in the range of 0.65 to ∞ (= fresh air). Their increasing use by automobile manufacturers has resulted in price falls over the last few years so that they could now be considered for use in such applications as heating boiler exhaust monitoring and general measurement of combustion products.

First principles

The wide-band lambda sensor uses the same type of Nernst cell as the simpler two-state sensor. The Nernst cell consists of a sensing cell made of Zirconium dioxide (ZrO_2) ceramic material with two electrodes. At temperatures above approximately $300\text{ }^\circ\text{C}$ the ceramic material becomes conductive to negatively charged oxygen ions but not for electrons, atoms or ions of any other gas. A galvanic potential is generated between two electrodes when there is a difference in the oxygen concentration in the gases on either side of the ceramic cell. This is the case when there is fresh air (oxygen rich) on one side of the cell and exhaust gases (oxygen poor) on the other. The voltage produced is dependant on the difference in the oxygen partial pressure of the two gasses. The wideband lambda probe is an expansion of this basic principle with the addition a second chamber called a 'pump cell' together with the monitoring or 'measurement cell' (Figure 1). A diffusion gap exists between the pump cell and monitoring cell and the pump cell is positioned between the flow of exhaust gasses and the diffusion gap. The diffusion gap links the pump cell to the exhaust gasses (shown in light green in Fig-

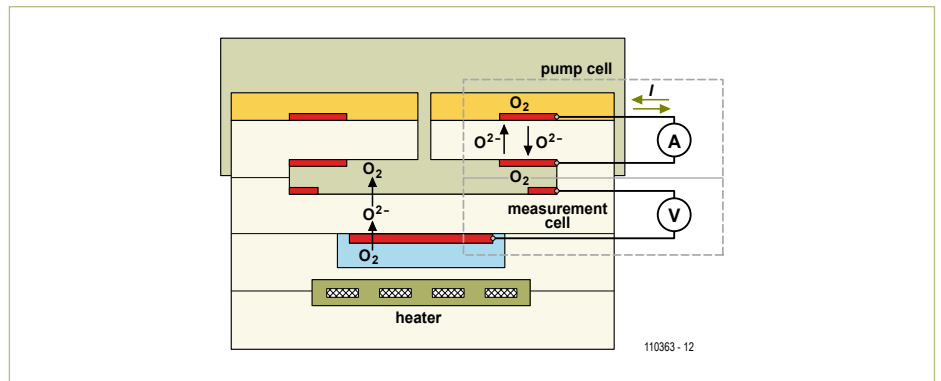


Figure 1. The wideband lambda probe uses two ceramic chambers; one is the pump cell and the other the monitor cell.

ure 1). The monitoring cell has the diffusion gap on one side and oxygen reference gas (fresh air) on the other side.

The monitoring cell (Nernst cell) of a simple narrow-band probe has the electrical characteristics shown in Figure 2. It simply detects the lambda value around $\lambda = 1$ when the curve passes through 450 mV . The trick with the wide-band probe is that current through the pump cell is regulated so that the lambda value of the gas in the monitoring cell is maintained at $\lambda = 1$. The pump cell is composed of the same material as the monitoring cell (zirconium dioxide ceramic stabilised with Yttrium oxide) and acts as a membrane through which its permeability to oxygen ions is con-

trolled by the application of a current (the 'pump current'). The monitoring chamber has a passage to the exhaust gases via the pump cell. The pump cell's job is to maintain the fuel/air mix in the monitoring chamber constant with a lambda value of 1 (450 mV at the monitoring cell). The pump cell reacts to a lean gas mix by discharging oxygen from the diffusion gap to the outside and conversely responds to a rich gas mix by changing the pump current to pump oxygen from the exhaust gas into the diffusion gap. The value of current through the pump cell required to keep the lambda value equal to 1 is an indication of lambda value of the gas since it changes almost linearly with the exhaust gas lambda value (see Figure 3).

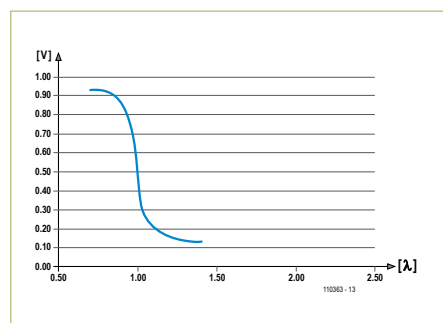


Figure 2. The Nernst cell in a wideband probe has the same characteristics as a narrow band probe.

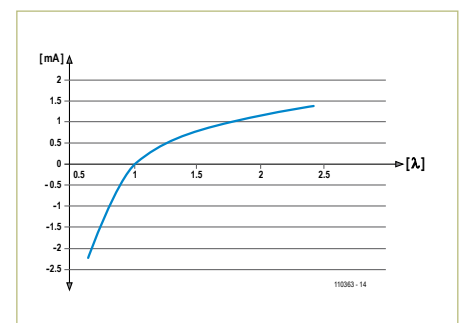


Figure 3. Current flow through the pump cell in the wideband probe is proportional to the lambda value of the gas.

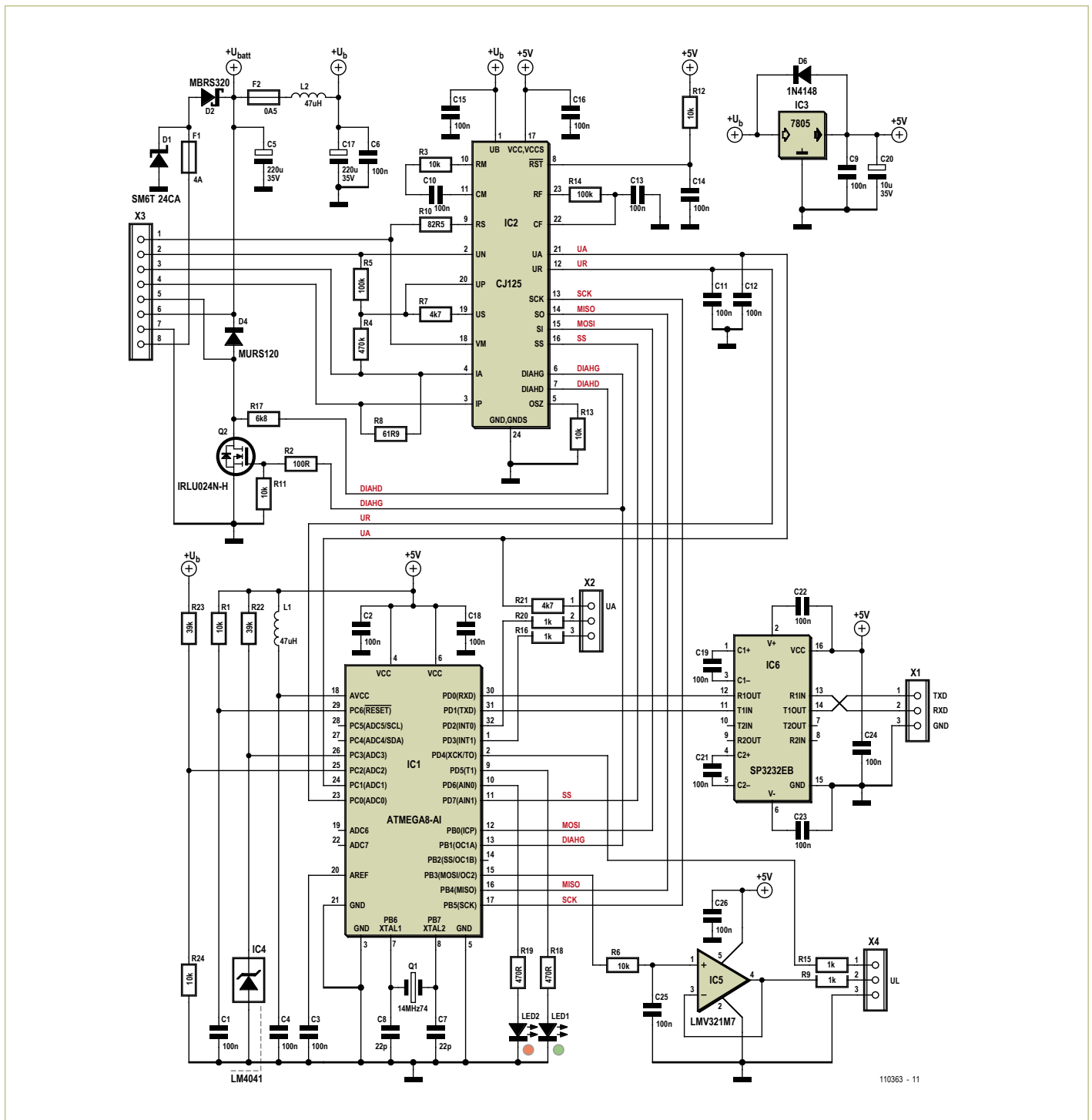


Figure 4. The circuit consists of the CJ125 lambda probe interface chip and a microcontroller which allows operation in stand-alone mode or via an RS232 data link.

Operation

Pump cell current is governed not only by the oxygen content of the exhaust gas but also temperature of the probe ceramic material. It begins to conduct oxygen ions at about 300 °C but has a more usable lower resistance when it is operating at around 750 °C.

The probe has a built-in heater to help reduce the effects of temperature variations on measurements. The temperature dependant internal resistance of the monitoring cell (Nernst cell) is used as a sensor to regulate the energy to the built-in heater. Different types of probe have different values of resistance at the operating tempera-

ture. A wide-band Bosch type LSU4.2 has a resistance of 82.5 Ω at the desired temperature while the more recent LSU4.9 has a resistance of 300 Ω. When the probe resistance is lower than this value it indicates that the probe is running too hot so the heater current is reduced. Starting with a cold sensor (immediately after a cold start for exam-

ple) it is necessary to gently ramp-up the heat to the ceramic element.

The wideband lambda sensor interface requires a minimum of two regulation circuits; one controls the pump current and the other controls current to the heating element. When Bosch introduced the CJ110 it was the first lambda probe interface IC, the CJ125 that we use here is a more recent incarnation. Its successor, the CJ135 has been announced but at the time of writing is not widely available.

The CJ125 provides all the most important functions necessary to interface to a wide-band lambda probe of the LSU4.x series:

- Pump current regulation
- An output signal proportional to pump current.
- Selectable amplification of the pump current signal (8 or 17).
- Measurement of Nernst cell internal resistance (R_i) for temperature control.
- Control signal O/P for probe temperature regulation.
- Comprehensive fault diagnostics.
- Calibration functions for internal resistance and pump current.
- Reference current function for the LSU4.9 probe.

The measured values of pump current and probe internal resistance are output as analogue values. Settings such as the calibration functions, amplification setting, fault and diagnostic registers are written and read via the SPI interface.

A somewhat simplified block diagram of the CJ125's complex internal structure is included in the data [1].

The circuit

The circuit shown in **Figure 4** consists of a CJ125 lambda probe interface chip and a microcontroller which allows the design to operate in stand alone mode or via an RS232 serial data link to an external PC. The addition of an RS232/USB adapter would allow connection to a PC's USB port. The use of the CJ125 here is based upon the application suggestion outlined in the Bosch data sheet for this device (**Figure 5**). A simple block diagram of the wide-band

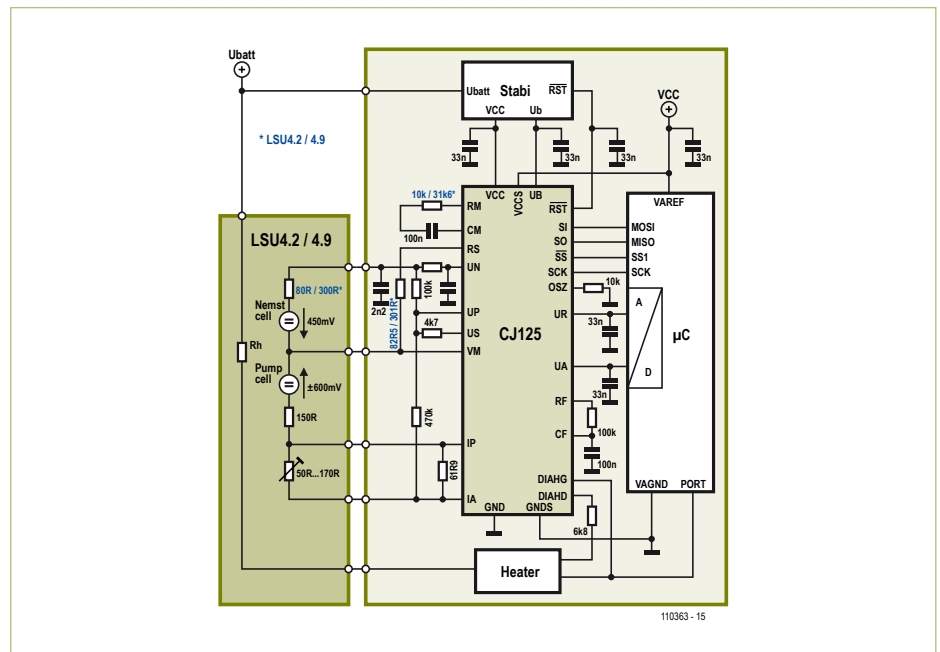


Figure 5. The data sheet application suggestion for the CJ125 includes a block diagram of the wide-band probe.

probe (LSU4.2/4.9) is also shown. An overview of the CJ125 interface signals are given below (Components are shown in the circuit diagram Figure 4):

- U_b (pin 1): filtered U_{batt} (vehicle battery supply 12 to 15 V).
- V_{CC}, V_{CCS} (pin 17): regulated 5 V supply.
- GND, GNDS (pin 24): circuit ground.
- I_A (pin 4): output of the pump current regulator. Pump current flows from IA through the shunt resistor R8 and the pump cell in the lambda probe and back to the VM pin (18).
- I_p (pin 3): inverted input of the pump current amplifier which gives the voltage drop across the shunt resistor R8.
- V_M (pin 18) is the virtual ground point for the pump current regulator and the λ probe. The virtual ground point is at half the operating voltage i.e. 2.5 V.
- U_S (pin 19): Nernst cell reference voltage (450 mV), also via R7 to the current regulator input UN.
- U_p (pin 20): non-inverted input to the pump current regulator.
- U_N (pin 2): The inverted input to the pump current regulator and also an input/output for measurement of the Nernst cell's internal resistance (R_i).
- R_S : calibration input/output for R_i (Nernst cell).
- RM/CM (pin 10/11): values of R3 and C10 define the R_i measurement current (AC current).

- CF/RF (pin 22/23): R14 and C13 form a low pass filter for the analogue Lambda signal (between the pump current amplifier and the Lambda signal output buffer).
- U_A (pin 21): analogue lambda output (voltage proportional to the lambda value).
- U_R (pin 12): analogue R_i output (voltage proportional to the Nernst cell R_i).
- DIAHG and DIAHD (pins 6/7): Diagnostic input for probe heater supervision (connects to the gate and drain respectively of the external power FET Q2 which switches current to the heating element).
- SCK/SO/SI/SS (pins 13/14/15/16): SPI interface to μC .
- /RST (pin 8): power-up reset using R12/ C14.
- OSC (pin 5): external 10 k Ω resistor to ground for the internal 192 kHz oscillator.

The pump current control is performed wholly by the CJ125 but the probe heater requires an external temperature regulator device and driver. To perform this function in the design we have used an ATmega8 microcontroller (IC1) together with a power FET. In operation the CJ125 outputs an analogue level representing the Nernst cell temperature at U_R (pin 12) but during power-up calibration it outputs the desired value of optimum probe temperature from

COMPONENT LIST

Resistors

R8 = 62Ω (SMD1206)
 R10* = 82.5Ω (SMD1206)
 R2 = 100Ω (SMD0603)
 R18,R19 = 470Ω (SMD0603)
 R9,R15,R16,R20 = 1kΩ (SMD0603)
 R21,R7 = 4.7kΩ (SMD0603)
 R17 = 6.8kΩ (SMD0603)
 R1,R3*,R6,R11,R12,R13,R24 = 10kΩ (SMD0603)
 R22,R23 = 39kΩ (SMD0603)
 R5,R14 = 100kΩ (SMD0603)
 R4 = 470kΩ (SMD0603)

*For LSU4.9:

R3 = 31.6kΩ (SMD0603)
 R10 = 300Ω (SMD1206)

Capacitors

C7,C8 = 22pF (SMD0603)
 C1,C2,C3,C4,C6,C9,C10,C11,C12,C13,C14,C15,
 C16,C18,C19,C21,C22,C23,C24,C25,C26 = 100nF (SMD0603)
 C20 = 10μF 35V (Panasonic B)
 C5,C17 = 220μF 35V (Panasonic F)

Inductors

L1,L2 = 47μH (SMD1210)

Semiconductors

D1 = SM6T30CA (SMB)
 D2 = MBRS320 (SMC)
 D4 = MURS120 (SMB)
 D6 = 1N4148 (Minimelf)
 LED1 = LED, red (SMD0805)
 LED2 = LED, green (SMD0805)
 Q2 = IRLU024N (TO251)
 IC1 = Atmega8-16TQ (Atmel), programmed, Elektor #110363-41
 IC2 = CJ125 (Bosch)
 IC3 = 7805
 IC4 = LM4041 (SOT23)
 IC5 = LMV321 (SC70)
 IC6 = SP3232EB (TSSOP16)

Miscellaneous

Fuse, 4A (Schurter type OMT 125 4.0A)
 Fuse, 500mA (Schurter type OMT 125 0.5A)
 Q1 = 14.7456MHz quartz crystal (HC49-SMD)
 X1,X2,X4 = 3-way PCB terminal block, 3.5mm pitch (AKL059-03)
 X3 = 8-way PCB terminal block, 3.5mm pitch (AKL059-08)
 PCB # 110363-1

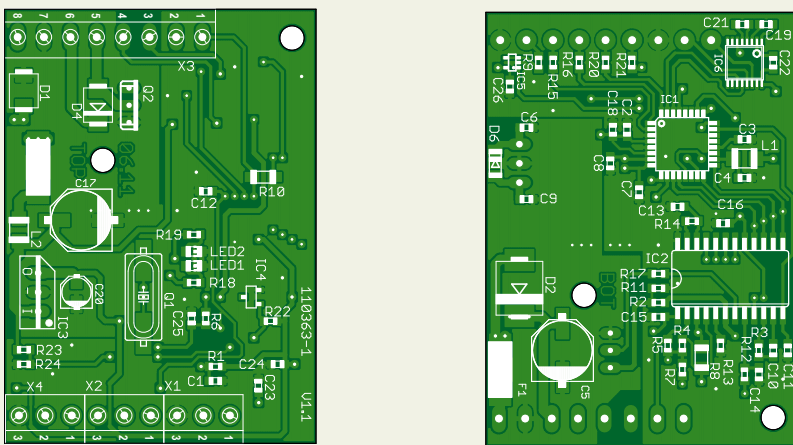


Figure 6. The double-sided PCB is almost completely populated with SMDs.

the same pin. The microcontroller measures the voltage level at U_R using its A/D converter ADC0 (pin 23) and regulates the probe heating current via a relatively low frequency PWM output from OC1A (pin 13) to the power FET Q2.

The CJ125 measures the voltage drop across the 61.9Ω shunt resistor R8 (the same for all probes) to calculate the pump current. The voltage drop is amplified and output from pin 21 (U_A) of the CJ125. Using the pump current given by the value of U_A the value of lambda can be read off from the lambda probe characteristic curve. The raw

value of λ is available at pin 1 of X2 for measurement or test purposes, it is also input to the microcontroller A/D converter input ADC1 (pin 24). Just like the U_R output the U_A pin also outputs a special value during the calibration phase at power-up. It outputs a calibration value of U_A which corresponds to a lambda reading $\lambda = 1$ (normally in the region of 1.5 V). Using this calibration value together with the probe characteristics the ATmega can use the measurements of U_A to produce a linearised lambda value to control the PWM output OC2 (pin 15). The lambda value is also integrated

by the low pass filter R6/C25 and buffered by opamp IC5 before it is output (pin 2 of X4) as UL the linear proportional analogue lambda value in the range of 0 to 4.5 V.

The A/D converter in the ATmega uses the relatively imprecise 5 V supply voltage as a reference. A better reference voltage of 1.22 V supplied by IC4 is connected to the input of A/D converter ADC3 (Pin 26) which allows a correction value to be transferred over the RS232 interface. A/D converter ADC2 is used to monitor the supply voltage (+Ub).

Microcontroller port pins PD5 and PD6 (pins 9 and 10) are used to drive two LEDs which indicate the operational status of the unit. Serial data input and output come from RXD and TXD pins on the microcontroller and level shifted by IC6 to provide RS232 compatible signals TXD and RXD at connector X1.

Pin 1 on connector X4 connects to PD4 (pin 2) of the controller via resistor R15, when it is pulled low it initiates a calibration. Pin 2 on connector X2 will go high when the microcontroller detects an error. A low on pin 3 of connector X2 will start the measurement process.

A DC supply in the range from 12 to 15 V DC is required to power the circuit. When in use to measure vehicle exhausts this supply can be taken from the car battery. The connection is made to pin 7 (ground) and pin 8 (+12 to 15 V) on X3. Current consumption of the circuit with the probe heater off is around 70 mA rising up to 3.6 A with the heater on. The current flows from pin 8 of connector X3 through the 4 A fuse F1 and Schottky diode D2 to the heating element in the probe on pin 6 of X3. Power for the circuit also comes from the cathode of D2 through a 0.5 A fuse and filtered by L2 and C17. This supply is labelled + U_b and is used by IC2, voltage divider (R23/R24), voltage monitor input on IC1/pin 25 and to the 5 V regulator IC3, which in turn supplies the other three ICs.

PCB, cabling and Versions

The PCB (Figure 6) is almost completely populated on both sides with SMD outline components. The only exceptions are IC3, the voltage regulator and MOSFET Q2 together with its mounting screw fitted to

the component side of the PCB. Pin assignments of the terminal strip connectors X1 to X4 are given in **Tables 1 to 4**.

X1 provides connections for the supply voltage and the lambda sensor. The sensor can only be used with the supplied lead and connector. The connector contains a manufacturer adjusted preset resistor (see the sensor's internal circuit shown in Figure 5). Make sure that wiring to the sensor is correct; a mistake here can cause the sensor to overheat and be destroyed.

The Bascom AVR source code can be built into different firmware versions depending on which of the three Include files are used. The version should be chosen according to the probe used and range of lambda values to be measured:

- Version 1: LSU4.2, range $\lambda = 0.7$ to 1.3 (calibrate with $\lambda = 1.0$)
- Version 2: LSU4.2, range $\lambda = 1.0$ to 2.0 (calibrate with $\lambda = 1.37$)
- Version 3: LSU4.9, range $\lambda = 1.0$ to 2.0 (calibrate with $\lambda = 1.38$)

The pre-programmed ATmega8 controller available from [1] is supplied with version 1 of the firmware. Those of you wishing to use version 2 or 3 will need to download the source code (free from [1]) together with the corresponding include file, compile them and burn the resulting hex code to the controller.

The three firmware versions use different linearisation tables to calculate the analogue voltage output U_L (X4/Pin 2). The functions connected with the output of U_A (X2/pin 1) and the RS232 data output are the same in all three versions. When probe type LSU4.9 is used the value of resistors R3 and R10 must be changed to $31\text{k}\Omega$ and $300\ \Omega$ respectively.

In Use

The unit is designed for stand-alone operation but can also be used in conjunction (with a suitable RS232 or RS232/USB cable) with a computer. In both cases indicators LED1 and LED2 show the operational status of the unit.

The inputs and output signals available at connectors X2 and X4 are used when the

Table 1. X3 pin out description (Lambda probe and supply voltage)

Pin number	Description	Lambda probe connection/colour
1	Virtual ground	IPN/yellow
2	Nernst voltage	RE+/black
3	Preset resistor	RT/green
4	Pump current	APE/red
5	Heating -	H-/white
6	Heating +Ubatt	H+/grey
7	Ground (0 V)	-
8	+12 to 15 V	-

Table 2. X1 pin out

Pin	Signal	Level
1	TXD	RS232 (output)
2	RXD	RS232 (input)
3	Ground	0 V

Table 3. X2 pin out

Pin	Signal	Level/voltage
1	U_A (raw λ signal)	0.2 to 4.8 V (output)
2	Fault signal	TTL (output)
3	Start Measurement	TTL (input)

Table 4. X4 pin out

Pin	Signal	Level/voltage
1	Start calibration	TTL (input)
2	U_L (linearised λ value)	0 to 4 V (output)
3	Ground	0 V

Table 5. The status LEDs (LED1 = green, LED2 = red)

LED	Flash rate	Operational status
Green	continuous (0 Hz)	Standby mode
Green	fast (5 Hz)	Warming up
Green	slow (1 Hz)	Probe operational and measuring
Red	slow (1 Hz)	Fault (Fault code available via RS232)
Red and Green	fast (5 Hz)	Failsafe mode (reset necessary)

unit operates in stand-alone mode (see tables 3 and 4). With the lambda probe connected to X3 and the unit powered up the circuit first enters an automatic calibration mode and the green LED lights. During the calibration period the voltage output from U_A (X2/pin 1) will be the voltage level which corresponds to a lambda reading of $\lambda = 1.0$. A voltage on the linearised output U_L (X4/Pin 2), will be the calibration value dependant on the firmware version in use ($\lambda = 1.0, 1.37$ or 1.38 , see previous heading). Calibration mode can also

be triggered by pulling pin 1 on connector X4 to ground. Calibration should only be attempted with a **cold lambda probe** otherwise calibration will be invalid.

To use the probe pull pin 3 on X2 down to ground. This starts the heater (green LED starts flashing quickly), when the operating temperature is reached (the green LED switches to slow flashing) measurements begin. The linearised value of lambda is now available as a voltage on output U_L (X4/Pin 2):

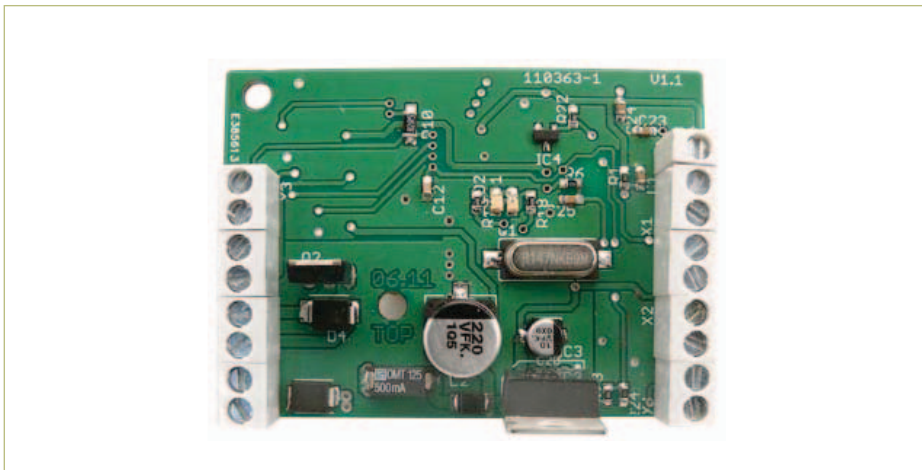
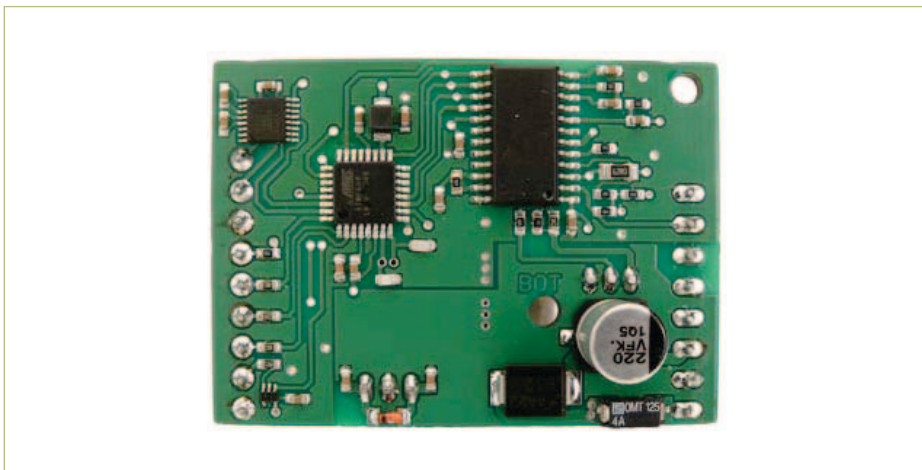


Figure 7. The stuffed Elektor prototype.



- Firmware-Version 1: $U_L = 0$ to 4 V corresponds to $\lambda = 0.7$ to 1.3
- Firmware-Version 2: $U_L = 0$ to 4 V corresponds to $\lambda = 1.0$ to 2.0
- Firmware-Version 3: $U_L = 0$ to 4 V corresponds to $\lambda = 1.0$ to 2.0

In the simplest case you can just hang a DC voltmeter (set to 5 V full-scale) onto U_L output and using a linear scale read off the value of lambda.
As already mentioned the second analogue output (X2/Pin1) outputs a voltage (U_A)



Figure 8. The lambda probe is supplied with a cable which includes a factory calibrated preset resistor.

which is proportional to the pump current. The pump current I_p can be calculated from:

$$I_p = \frac{U_A - U_{offset}}{A_p \times R_{shunt}}$$

where U_{offset} is the calibration value of the U_A voltage, i.e. the voltage level output from X2/pin 1 during calibration mode (around 1.5 V). A_p is the amplification factor of the pump current signal (8 in rich and 17 lean burn) and R_{shunt} is the shunt resistor (61.9Ω).

Referring back to the lambda probe data sheets (LSU4.2 or LSU4.9) it is possible for example to calculate the value of I_p from the lambda value, enter them in an Excel data table and display them using a trend line. Knowing the value of pump current allows us to calculate the exhaust gas oxygen content. Using the LSU4.2 in a lean burn situation (excess air in the combustion) the oxygen concentration is given by:

$$O_2 = \frac{I_p + 0.035}{0.1221}$$

Using the RS232 interface at connector X1 allows use of all of the diagnostic capabilities of the CJ125 and to read measurement values and the operational status of the circuit. The various settings can be made by using some very basic RS232 commands. These additional features available via the RS232 interface will be explored in a follow-up article.

(110363)

Internet Link

[1] www1.futureelectronics.com/doc/BOSCH/CJ125.pdf

www.**TOROIDY.PL**
TRANSFORMATORY LACHOWSKI



WE INDUCE QUALITY IN THE ROUND

Toroidal transformers for high quality audio equipment.
One and three-phase custom made transformers.
Step-up and step-down 230 / 110V transformers
AC/DC power supplies.
Inductive components.



P.W.P.H. Lachowski
ul. Mazowiecka 20, 16-002 Książyno - Białystok, Poland
e-mail: info@toroidy.pl
phone: +48 85 74-74-378, fax: +48 85 66-31-213

Lachowski Toroidy advertising campaign is co-financed by the European Regional Development Fund under the Operational Programme Innovative Economy




INNOVATIVE ECONOMY
NATIONAL COMMISSION STRATEGY




EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

SiliconRay
Online Electronics Store


- Boards • Kits • Modules • Components • Tools • Instruments
- FOR
- Companies • Professionals • Students • DIYers • Amateurs




NE555 Timer/Oscillator DIP/SOP
The most famous IC all the time
\$0.06



0Ω-20MΩ 177 Values Resistors Book
A "must have" book for all readers
\$60



CC2500 2.4G Transceiver Module
Small, versatile, simple, ultra low power
\$3.5



6E2 Cateye Signal Level Indicator
The cateye is dancing with music
\$35.5

Simplify your electronics projects by visiting
<http://www.siliconray.com>

Beta
LAYOUT

REFLOW-KIT
Soldering technology, tools and aids for SMD and THT assembly
€129.00

EVERYTHING'S EASY!

PCB-POOL
Free Stencil
Get a free SMD laser stencil with every Prototype order

WORLD FIRST
FITS-OR-NOT
3D PCBs: Hands-on collision check

Assembly service
Even one component possible

Cool
Alu-Core IMS PCBs

PCB-POOL RULES!

PANEL-POOL
FRONTPANEL
NEXT GENERATION
Quartus

DIGITAL PRINT TECHNOLOGY
• scratch resistant
• UV resistant
• solvent resistant

FREE DESIGNER

Pimp my electronic device!

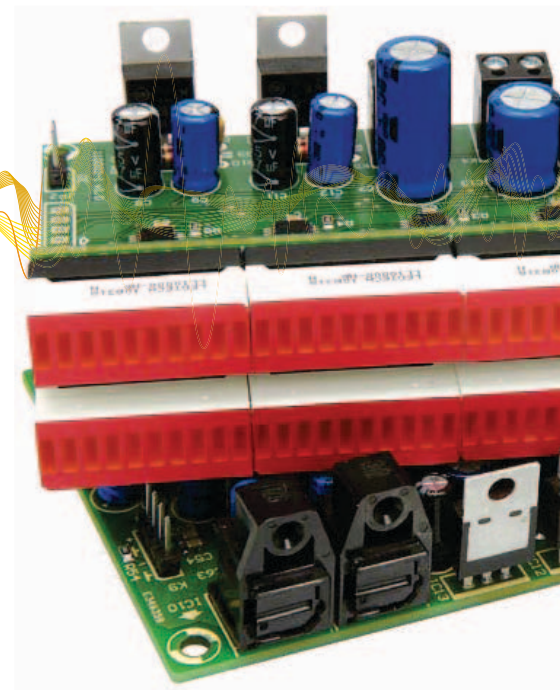
Free Phone UK: 0800 389 8560
sales@pcb-pool.com

All registered brands remain the registered trademarks of the respective manufacturer!

www.beta-layout.com

Audio DSP Course (7)

Digital peak level meter



In professional recording situations a peak level meter is an indispensable weapon in one's arsenal of equipment. It can be used to ensure that sound levels are within the dynamic range of electro-acoustic components and thus help in minimising distortion. It can assist in keeping noise to a minimum while avoiding clipping. This article describes a digital peak level meter built using the DSP board that accompanies this course along with a separate LED display.

By Alexander Potchinkov (Germany)

When recording music it is important to keep levels in the signal chain as high as possible to minimise the effect of the noise and interference that will inevitably be encountered; on the other hand it is essential to avoid clipping at all costs, as this can lead to unacceptable distortion. Although a little distortion might go unnoticed in a loud rock music passage, it can rapidly become excruciating when it occurs on an (unprocessed) human voice or on a woodwind instrument such as an oboe. To achieve the best compromise between these two antagonistic goals we need a peak level meter to determine the minimum acceptable level above the noise while simultaneously not exceeding the clipping threshold. An audio level meter will typically display not only the peak signal level encountered, but also the RMS (root mean square) level.

Formerly mechanical moving-pointer instruments (with inherent inertia) were used to display audio level, with mirror galvanometers used in professional applications. Nowadays inertia-free indicators such as LED bar graphs or plasma, fluorescent or liquid crystal displays are used. How-

ever, inertia was a valuable feature of the older instruments, and so in more modern devices it is simulated using (perhaps digital) signal processing. Our design in its basic form has a dual LED bar graph comprising two rows of forty LEDs each. The LEDs are inexpensive and easy to obtain and it is a simple matter to extend the display by adding more: the design allows for extension to eighty or even 120 LEDs per row. Although this might seem excessive, it is worth noting that in professional equipment the bar graphs often have at least 100 elements each. The display is constructed on a separate board.

Peak level meters are defined by the type of rectification used, their display resolution and their so-called 'ballistic' properties. In the United States the term 'VU meter' is common, whereas in Europe 'peak programme meter' is preferred. Displays vary in the range of levels that can be displayed (the 'modulation range' for normal audio and the 'headroom' to accommodate overdriven audio). The ballistic properties describe how the meter imitates the inertia of its mechanical ancestor, and include the 'attack time' or 'integration time', and the 'fallback time', 'release time' or 'decay time'. The ballistic properties can

be thought of as being characteristics of the rectifier stage in the meter. In an analogue peak level meter this stage is followed by a level converter and a display, and our design is just the same.

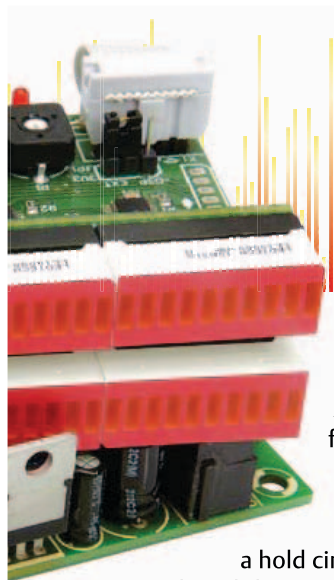
The desirable ballistic properties are different in analogue recording and digital recording applications, as in digital recording it is even more important to avoid clipping than in analogue recording. It is therefore desirable to provide a simple parameterisation of the signal processing code.

We will now go on to describe the signal processing stages and the DSP program that implements them, and after that we will describe the circuit of the display unit.

Signal processing

The digital signal processing steps involved in the operation of the peak level meter can be divided into the four blocks shown in **Figure 1**.

1. Peak value rectifier with time constants ('ballistic rectifier')
2. Level calculation
3. Level quantisation and scaling
4. LED driver



A 'peak value rectifier with time constants' is a combination of an ordinary rectifier with a hold circuit and adjustable attack and decay time characteristics.

Since our digital implementation of this unit is intended to be a direct replacement for the corresponding analogue circuit, it makes sense to look first at how the function is achieved in analogue electronics: see **Figure 2**. The circuit has two operating modes: attack mode, when $U_{in} > U_{out}$, in which capacitor C is charged with time constant $\tau_1 = CR_1R_2/(R_1+R_2)$, and decay mode, when $U_E < U_A$, in which capacitor C is discharged with time constant $\tau_2 = CR_2$. Time constant τ_2 determines the decay rate while $\tau_1 < \tau_2$ sets the attack rate. These two time constants can be thought of as the parameters of the ballistic rectifier. Now, assuming the value of C is given, time constant τ_2 depends only on R_2 , whereas time constant τ_1 depends on both R_1 and R_2 . So, given C and the two time constants, we can calculate first R_2 and then R_1 . The interaction between the two resistance values in the equation for τ_1 makes it difficult to provide for independent setting of the two parameters using two potentiometers in an analogue circuit. In the digital version there is no such problem.

Figure 3 shows the signal processing in the ballistic rectifier. The block labelled 'abs(.)' corresponds to a full-wave rectifier, the quantities α and β determine the time characteristics, and the block labelled 'max(0,.)'

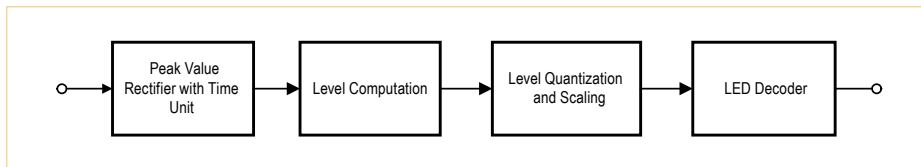


Figure 1. Signal processing blocks in the digital peak level meter.

distinguishes between attack mode operation and decay mode operation: the signal is allowed to pass only in attack mode. This would be achieved in the analogue domain using a reverse-biased diode to block the signal. If the signal is blocked, that is, when the system is in decay mode, the output signal is obtained from the stored value in the block labelled 'z⁻¹': here the time constant is determined by the value of β .

The level calculation involves converting the rectified signal value into decibels. More precisely, in the digital implementation we convert to units of dBFS (decibels relative to full scale), where the level is compared to the largest representable digital value: in our case this is 1. So, for example, a signal value of 0.1 corresponds to a level of -20 dBFS and a signal value of 0.05 to a level of -26 dBFS. In order to match up these values with the actual signal level present at the input to the board we need to know both the characteristics of the ADCs (which can often be tricky to determine precisely from their datasheet) as well as the exact gain of the input amplifier. A better plan is to feed a signal of known amplitude into the board (for example 0 dBm = 0.775V) and

use this to calculate a suitable scaling factor to apply to obtain the correct final answer. We could add a calibration cycle to the unit, as is employed in some high-end multimeters. In this connection we need to keep in mind the difference between peak level and RMS level. A sine wave with an amplitude of 1 V has a peak level of 1 V and an RMS level of 0.7071 V: these might correspond to 0 dB and -3 dB. So as we can see, care is required in interpreting the decibel values: we need to know what we are measuring.

The result of the level calculation can be anywhere in the range from about -110 dB to 0 dB; however, because of the ADC's noise floor the useful lower limit is in practice around -90 dB. We need to choose some subset of this range for display: for this project we have decided to allow a choice between a range covering -39 dB to 0 dB and a range covering -78 dB to 0 dB. It is straightforward enough to modify the relevant constants in the DSP program to select a different range if required. Values within the selected range need to be quantised for display on the bar graph, and the quantisation process will depend on the total number of LEDs in the bar. In the case

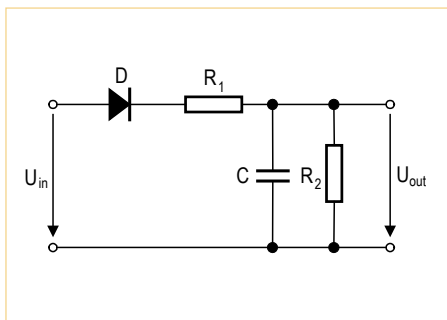


Figure 2. Analogue peak value rectifier with time constants.

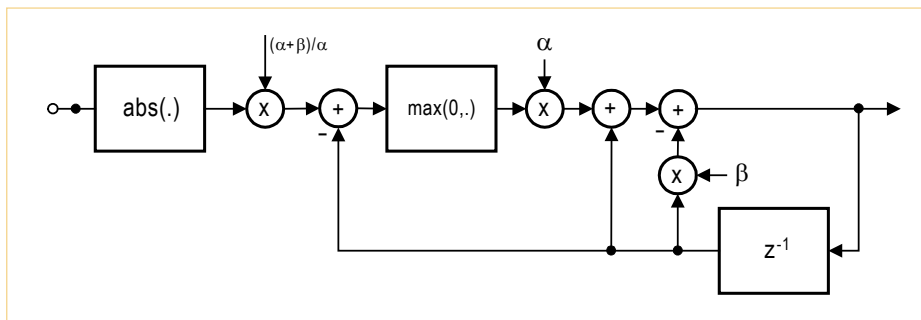


Figure 3. Digital peak value rectifier with time constants.

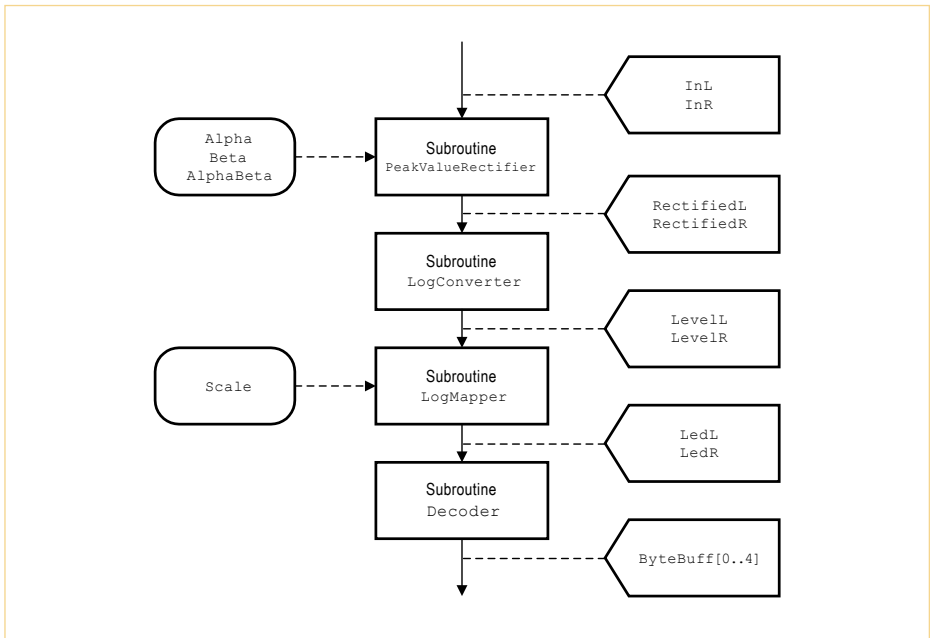


Figure 4. Subroutines and signals in the audio loop.

of forty LEDs the first range will quantise to a 1 dB step per LED, while the second range will quantise to a 2 dB step per LED. This is carried out in the third block, labelled 'level quantisation and scaling'.

Next the quantised values must be encoded into bit patterns suitable for sending to the display module. Note that the order of the LEDs as far as the display module is concerned might not be the natural one for our purposes, and so the bits may need to be suitably rearranged. This is the job of the fourth block in Figure 1.

Finally we need to decide how often, within the audio loop, we want to transfer data to the display module. The simplest approach

is to use an integer multiple of the audio sample period, as this timebase is already available. In principle it would be possible to use the DSP's powerful timer facility, but that would be overkill in this case.

Subroutines in the audio loop

Figure 4 shows the audio loop subroutines, their parameters and the signals that flow between them. There is a total of four subroutines, of which only two require parameters.

The subroutine `PeakValueRectifier` implements the ballistic rectifier on two channels. It reads the signals `InL/R` as its inputs and outputs the signals `RectifiedL/R`. At the beginning of

this subroutine there is a correction for the gain of the rectifier that depends on the two time constants, and then the absolute value of the signal is calculated: this corresponds to full-wave rectification. The gain correction factor lies between 1 and 2 and is therefore implemented using a `mac` instruction. A conditional branch is then used to select between the two modes ('attack' and 'decay') of operation. The branch is conditional on whether the difference value at the output of the first summer (here used as a subtractor) is negative or not. The two paths through the code correspond to the two modes, and hence also to the charging or discharging of the 'digital capacitor'. The parameters `Alpha` and `Beta` (α and β) determine the time constants. The parameter `AlphaBeta` is computed from the values of α and β and provides the gain correction factor. Figure 5 shows the output of the peak value rectifier for a sinewave burst input: the burst is too short to allow the rectifier to settle fully. The subroutine parameters are listed in Table 1.

The subroutine `LogConverter` computes the level of the rectified signal with suitable scaling. It reads in the signals `RectifiedL/R` and outputs the signals `LevelL/R`, which represent the levels. The dynamic range covered by a 24-bit value is 138 dB, and this range needs to be scaled to a value between -1 and 0 to represent the result. We approximate the base 2 logarithm function over the interval [0.5, 1] using the same polynomial method as we used for the sine function in the previous instalment in this series. The calculation is simplified by using the DSP instructions `clb` and `normf`, which are designed to facilitate floating-point arithmetic. The subroutine as it stands can cope with an input range from -110 dB to 0 dB. If a greater range is

Table 1. Subroutine parameters for the peak value rectifier			
Fractional DSP program parameters and their default values			
Parameters and default values	DSP program parameters	Calculation of DSP program parameters	Default values for the DSP program parameters
$t_A=10$ ms, $t_R=100$ ms, $A=480$, $R=4800$	Alpha Beta AlphaBeta	$\alpha=0.4234^{(1/R)}-0.649844^{(1/A)}$ $\beta=1-0.4234^{(1/R)}$ $\alpha_\beta=\beta/\alpha$	0.00036018 0.00023982 0.6658

required then a switchable gain stage must be added: this requires just a couple more lines of assembler code. The subroutine `LogConverter` requires no parameters.

The subroutine `LogMapper` converts the scaled decibel value from subroutine `LogConverter` into a form suitable for the LED display. It reads in the signals `LevelL/R` and outputs the signals `LedL/R`. As shown in **Table 2**, the parameter `scale` is used to switch between the two output scales. The subroutine `LogMapper` needs to know the number of LEDs in each row of the display and the dynamic range that the display is to cover. We will assume that the difference in level between successive LEDs is constant across the display. The parameters `dBScale40` and `dBScale80` correspond to a 40 dB scale with 1 dB steps between successive LEDs, and an 80 dB scale with 2 dB steps. The subroutine reads the level values and generates two bytes with values from \$000000 to \$000028, giving the number of LEDs that are to be lit in each row. A more detailed description of the calculation involved is given in the accompanying text box.

With modifications to this code it is possible to create non-linear decibel scales, allowing you to zoom in on part of the range. For example, you could cover the important range between -19 dB and 0 dB in steps of 1 dB using 20 LEDs, with the other 20 LEDs covering the range from -77 dB to -20 dB in steps of 3 dB.

The subroutine `Decoder` drives the LED display on the basis of its inputs `LedL/R`. Its output `ByteBuff` is not a signal in the conventional sense since the display module is driven in groups of 16 bits rather than 24: a total of 80 bits is needed for two rows of forty LEDs and 160 bits for two rows of eighty LEDs. So to drive two rows of forty LEDs we need periodically to send 80 bits to the display module over the DSP's SPI port. These 80 bits are divided among five 16-bit shift registers. As a consequence of the relatively high total LED currents it happens to simplify the printed circuit board layout to divide each of these shift registers into two

Table 2. The parameter `scale` in the `LogMapper` subroutine

Parameter	Range	Data type	Word length	Alignment
<code>scale</code>	[0,1]	integer	24	right-aligned

Table 3. Data layout in the `ByteBuff` array (40-LED rows)

Address in DSP-RAM	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
<code>ByteBuff</code>	L, LEDs 39 to 32	R, LEDs 39 to 32	Not used
<code>ByteBuff+1</code>	L, LEDs 31 to 24	R, LEDs 31 to 24	Not used
<code>ByteBuff+2</code>	L, LEDs 23 to 16	R, LEDs 23 to 16	Not used
<code>ByteBuff+3</code>	L, LEDs 15 to 8	R, LEDs 15 to 8	Not used
<code>ByteBuff+4</code>	L, LEDs 7 to 0	R, LEDs 7 to 0	Not used

Table 4. Program files used for the digital peak level meter

<code>PeakLvm.asm</code>	Main program
<code>LogCoef.tab</code>	Polynomial coefficients for logarithm approximation
<code>src4392.tab</code>	Byte sequence for configuring the SRC
<code>ivt.asm</code>	Interrupt vector table entries, audio interrupts
<code>esai4r2t.asm</code>	Audio ISR: four input channels, two output channels
<code>Mioequ.asm</code>	Handy names for DSP I/O addresses

8-bit halves, one for the left channel and one for the right. Subroutine `Decoder` is thus responsible for converting the byte values received from subroutine `LogMapper` into five 16-bit words for transmission over the SPI port. The routine can be extended in a fairly mechanical fashion to 160-bit or 240-bit output; it uses no parameters. **Table 3** shows the data layout in the array `ByteBuff` for the case of a display with two rows of forty LEDs.

DSP program and ideas for expansion

Table 4 lists the software components required for this project (available for free download at www.elektor.com/110391). In addition to the files used in the other projects and test programs in this series we need the file `PeakLvm.asm`, which is the main program, and the file `LogCoef.tab`, which contains the polynomial coefficients used to approximate the logarithm function.

We have three ideas for possible expansion of this project. The first, already mentioned above, is to add a second or even a third display module. This entails expanding the `Decoder` subroutine and transferring more data to the display. Another idea is to replace the peak level rectifier with an RMS rectifier, and this can be accompanied by suitable changes to the ballistics. It would be possible to combine the display of peak and RMS levels, displaying one as a bar and the other as a single illuminated LED. The third idea is to add an A-weighting filter in the signal processing chain before the rectifier to allow measurements in dBA or dB SPL to be carried out. More information on A-weighting filters can be found in Wikipedia and elsewhere.

Display refresh and data transfer

Inside the audio loop a sample counter is maintained in register R7, taking advantage of the modulo-N function provided by the DSP's AGU. Whenever this counter resets

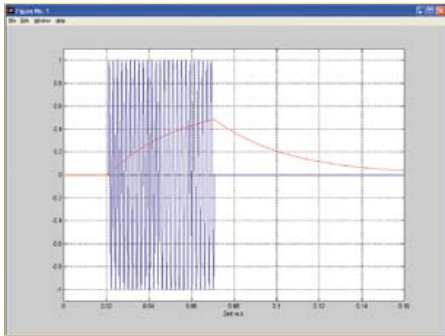


Figure 5. Output signal of the peak value rectifier (red) for a sine wave burst input (blue).

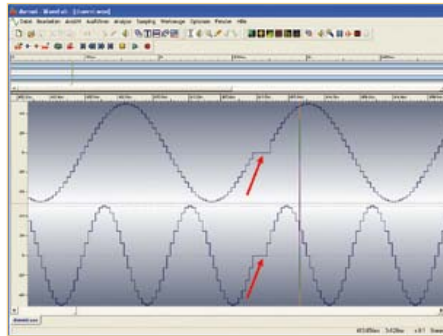


Figure 6. Signal distortion effect when the SPI clock is too slow.

to zero the DSP program writes 80 bits to the display module. The refresh rate of the display is therefore given by $f = f_s/N$ where $f_s = 48$ kHz is the audio sample rate. If we set N to 192 then the display will be refreshed every 4 ms.

The data bits are transferred to the display in one operation. If, for example, the SPI port is configured to run with a 1 MHz clock, it will take 80 μ s to transfer 80 bits. With an audio sample rate of 48 kHz this is about four sample times. Audio samples arriving during this period will be read from the interface register by the interrupt code, but will not be processed by the DSP program. This is unlikely to be a problem in this application with ordinary analogue audio sources. However, if the input is digital and sample-accurate peak values are wanted (or if more than forty LEDs per row are to be driven) then the situation is less satisfactory. It will be necessary to increase the SPI clock rate to 10 Mbit/s, 20 Mbit/s or even higher, and to ensure that the wiring between the DSP board and the display is electrically suitable for carrying data at this speed. An alternative strategy is to spread out the write operations, sending one 16-bit value per audio sample period. In this case the audio sample rate and the SPI clock rate are independent of one another. Since we already have a modulo- N sample counter, implementing this scheme is relatively straightforward. In our prototype we used an SPI clock frequency of about 10 MHz,

which, with forty LEDs per row, resulted in no dropped audio samples even when transferring all the display data in one operation. The setting required in the DSP to achieve this is $x : HCKR = \$000034$ (noise reduction filter disabled, prescaler bypassed, divide by $2 * (6+1)$, SHI clock 10.5326 MHz). We obtained reliable operation in the prototype with a clock frequency of 18.432 MHz (achieved by setting $x : HCKR = \$000024$). If the display is unstable or incorrect, this is a sign that the data rate is too high. It can be reduced by increasing the division ratio HDM. **Figure 6** illustrates how sample values are lost when the SPI clock frequency is set to approximately 1 MHz.

Display module circuit

There are in principle two ways to drive an LED bar graph display. The first is to multiplex the LEDs using narrow high-current pulses. The LEDs are arranged in a matrix and a clock generator and accompanying logic (counters, comparators and decoders) generate the necessary waveforms. The multiplexed arrangement has the disadvantage that if the clock should stop for any reason the unfortunate LED that is lit at the time will probably be destroyed. There is also the complexity of the logic involved, which would probably be best implemented in a programmable logic device such as a CPLD rather than in discrete devices. The alternative approach is a long shift register with individual drivers to provide the low (but continuous) current required by each

LED. In our case the shift register has eighty stages, and so eighty drivers are needed. This design is ‘low risk’ compared to the multiplex arrangement, but there is considerably more wiring. For example, selecting one of 100 LEDs in a multiplex arrangement needs just a seven-bit value, whereas in the shift register approach each LED is wired individually.

Certain semiconductor manufacturers, notably Texas Instruments and Catalyst Semiconductor (now ON Semiconductor), offer low-cost LED driver ICs which include not only a serially writable and readable shift register but also current sources for driving the LEDs, allowing simple brightness control using a single trimmer potentiometer. We have selected the Texas Instruments TLC5926, which can drive 16 LEDs. The shift registers in these devices can easily be cascaded to form chains of any desired length. We have designed a separate display module consisting of two boards (one for the driver and one for the display itself) specially for this DSP course. The two boards can be mounted together to form a compact unit: see **Figure 7**. We will now look briefly at the circuit of the module (**Figure 8**) and how it is used.

To drive eighty LEDs we need five driver ICs. These are arranged in a chain starting with IC5 and finishing with IC1. To simplify the layout of the board each driver IC is responsible for eight LEDs from each row. The ICs require two supply voltages: 3.3 V for the SPI port (which in the case of the first IC in the chain is driven from the DSP) and a supply for the LEDs delivered via K5. The voltage of this supply should be as low as possible in order to minimise power dissipation in the ICs. We chose to use 3.0 V, generated from a 5 V supply using voltage regulators IC6 and IC7. Each regulator supplies forty LEDs. The supply on K5 can be higher than 5 V, but then the power dissipation in the regulators will increase.

The DSP can drive its SPI port, which is connected to the display module, in 8-bit, 16-bit or 24-bit mode. We use 16-bit mode because 16 is the largest available factor of 80. To create an image on the LEDs the

DSP sends a sequence of five 16-bit words to the module. The module is designed so that multiple units can be cascaded in a chain: these units will normally be mounted side-by-side, and the cascade is realised by connecting K3 of one board to K2 of the next. When several units are cascaded the number of 16-bit words the DSP must send is correspondingly increased.

The DSP only communicates directly with one of the display modules, even if several are being used in a cascade. The connection is made using a ten-way ribbon cable fitted with two standard headers, linking K7 on the DSP board with K1 on the display driver module. The 3.3 V supply for the SPI port can be obtained from an external source or can be taken directly from the DSP board. In the latter case a jumper should be fitted to link pins 1 and 2 of JP1 (labelled 'DSP'), and connector K4 is not needed. JP2 is used to enable the display: with the jumper

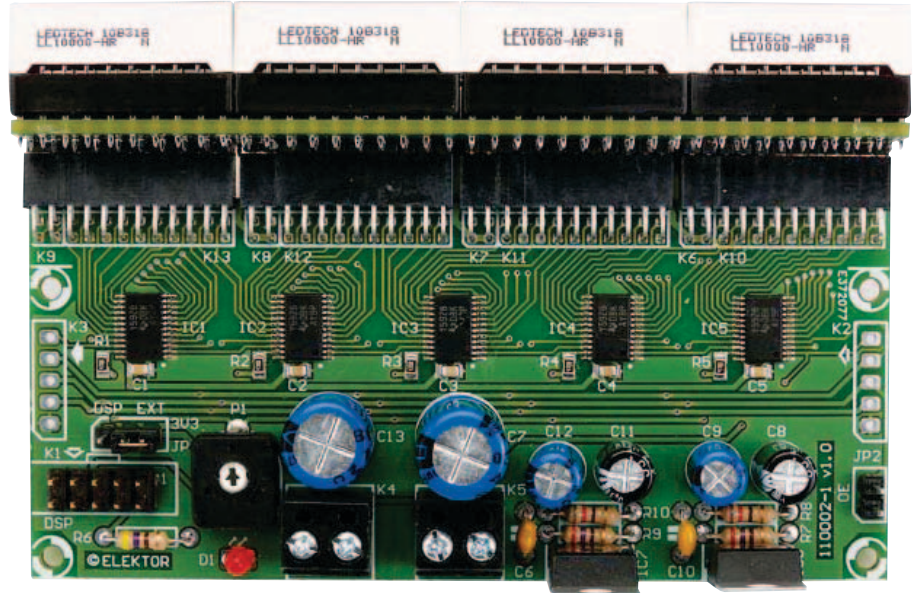


Figure 7. Construction of the separate display module.

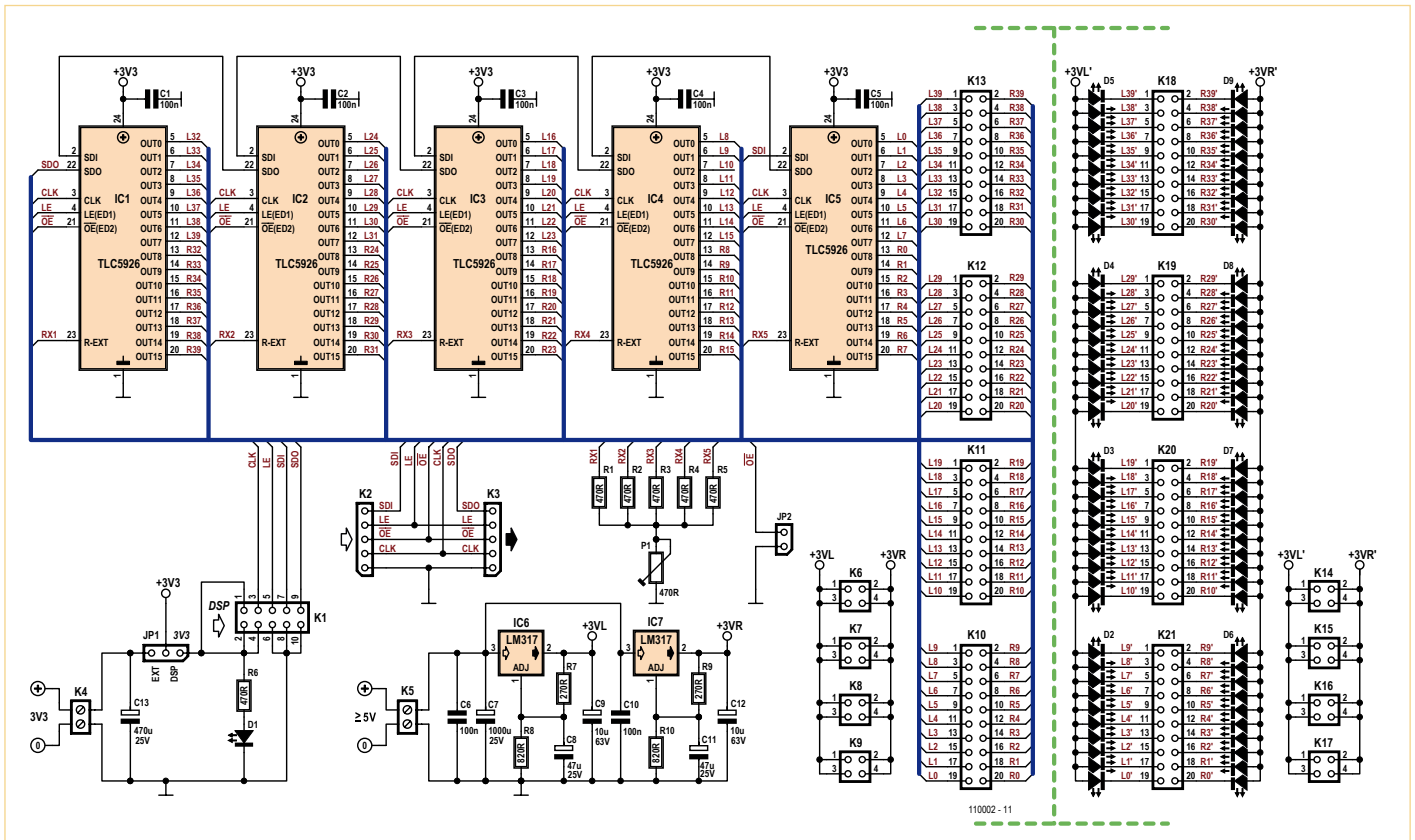


Figure 8. Circuit diagram of driver and display.

removed the display remains dark. Trimmer P1 is used to adjust the brightness of the display. It is important to bear in mind that if each LED draws a current of 10 mA then the total current for the board will be up to 800 mA if all LEDs are lit. In this case the voltage regulator will need to be fitted with a small heatsink.

The display proper has its own board, designed to accept the 10- or 20-LED bar graph arrays that are available in a range of colours from several different manufacturers. We recommend using high-efficiency red LEDs as these can be operated at lower currents. The display board is mounted perpendicular to the driver board: this requires a certain amount of dexterity with the large number of connectors. The simplest approach is to lay the display board horizontally on the bench and carefully push the driver board vertically down onto it.

The DSP is capable of driving its SPI port at over 10 Mbit/s, and the LED driver IC can accept data at up to 30 MHz. This means that whether we have one, two or several display modules in a cascade, the display can still be updated far faster than the human eye can see.

We have also created a test program for the display module, called `tst_led.asm`. It includes two of the components of the peak level meter program: the decoder routine and the data transfer routine. The test causes 23 of the LEDs in one row to light and 27 in the other. The file `tst_led.asm` is the only one required for this program.
(110391)

Display board

The two-part printed circuit board for the display module is available with the SMDs already fitted. The order code is 110002-71 (details at www.elektor.com/110391).

Calculating the number of active LEDs from scaled level values

The decibel level P_x corresponding to a sample value x , in this case an output sample from the rectifier, is given by

$$P_x = 20 \log_{10}(x), x > 0$$

where the useful range is $-110 \leq P_x \leq 0$. However, because of the fractional arithmetic used in the DSP, the logarithm routine in our program calculates a scaled level

$$P_{sx} = \log_2(x) / 32 = (\ln(10) / \ln(2)) \log_{10}(x) / 32$$

so that $-0.570956 \leq P_{sx} \leq 0$. By $\ln(\cdot)$ we mean the natural logarithm function: this function (along with logarithms to the base 10) is available on most scientific pocket calculators. Rearranging and simplifying, we have

$$P_x = 20 (\ln(2) / \ln(10)) \times 32 \times P_{sx} = 192.6592 P_{sx}$$

and conversely

$$P_{sx} = (\ln(10) / \ln(2)) P_x / (20 \times 32) = 0.005190513 P_x.$$

We would like to use a right-aligned integer value to control the LEDs. The integer value \$000001 corresponds to 2^{-23} in the DSP's fractional representation, or 1.1921×10^{-7} . For example, suppose we want to have the range $-60 \text{ dB} \leq P_x \leq 0 \text{ dB}$ correspond to between 0 and 80 LEDs being lit. To turn on 80 LEDs we want to generate the value

$$\text{LED}_{80} = \$000050 = 9.536743 \times 10^{-6}.$$

Now, considering the two extremes of the range, and writing N_{LED} for the number of active LEDs, we have

$$P_x = -60 \text{ dB}, P_{sx} = -60 \times 0.005190513 = -0.31143076 \text{ and } N_{\text{LED}} = \text{LED}_0 = 0$$

and

$$P_x = 0 \text{ dB}, P_{sx} = 0 \text{ and } N_{\text{LED}} = \text{LED}_{80} = 9.536743 \times 10^{-6}.$$

The DSP must therefore calculate the number of active LEDs from the scaled level using

$$N_{\text{LED}} = P_{sx} \times 80 \times 2^{-23} / 0.31143076 + 80 \times 2^{-23} = 3.0622354 \times 10^{-5} \times P_{sx} + 9.536743 \times 10^{-6}.$$

The scaling in this example is illustrated in **Figure A**.

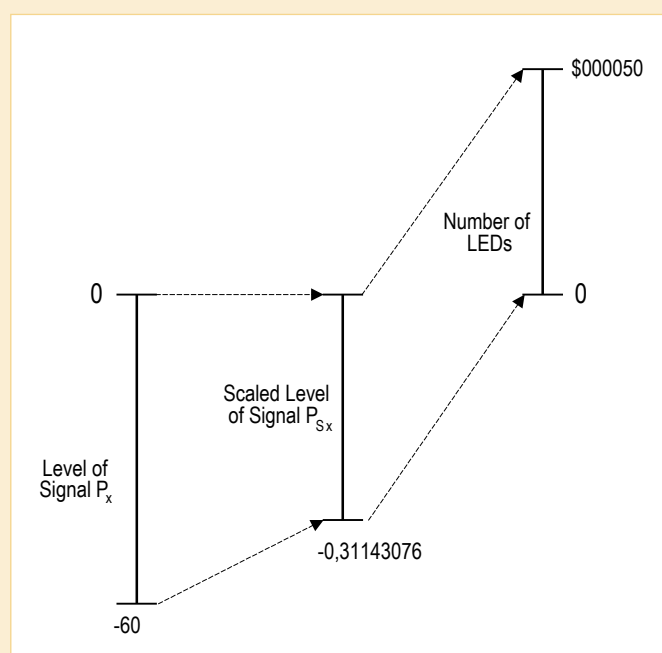


Figure A. Signal levels and the range covered by the LED display.

Elektor Academy Webinars in partnership with element14

Elektor Academy and element14 have teamed up to bring you a series of five 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.

Webinar Programme:

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.

The Making of an Improved Radiation Meter

NEW!

Date: Thursday February 16, 2012

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

Presenter: Thijs Beckers (Elektor)

This webinar covers the design history and ins and outs of Elektor's highly successful Improved Radiation Meter published in the November 2011 edition. This low-cost and all do-it-yourself instrument is able to detect different types of radiation like alpha, beta and gamma using 'sensors' you never thought of as suitable for this kind of application. You can look forward to a lively and highly topical webinar on a guaranteed 'experimentalicious' circuit!

Webinar Archive:

Now available to view on demand at www.element14.com/webinars

Let's Build a Chaos Generator

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!

E-Blocks, Twitter and the Sailing Club

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.

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.



element14

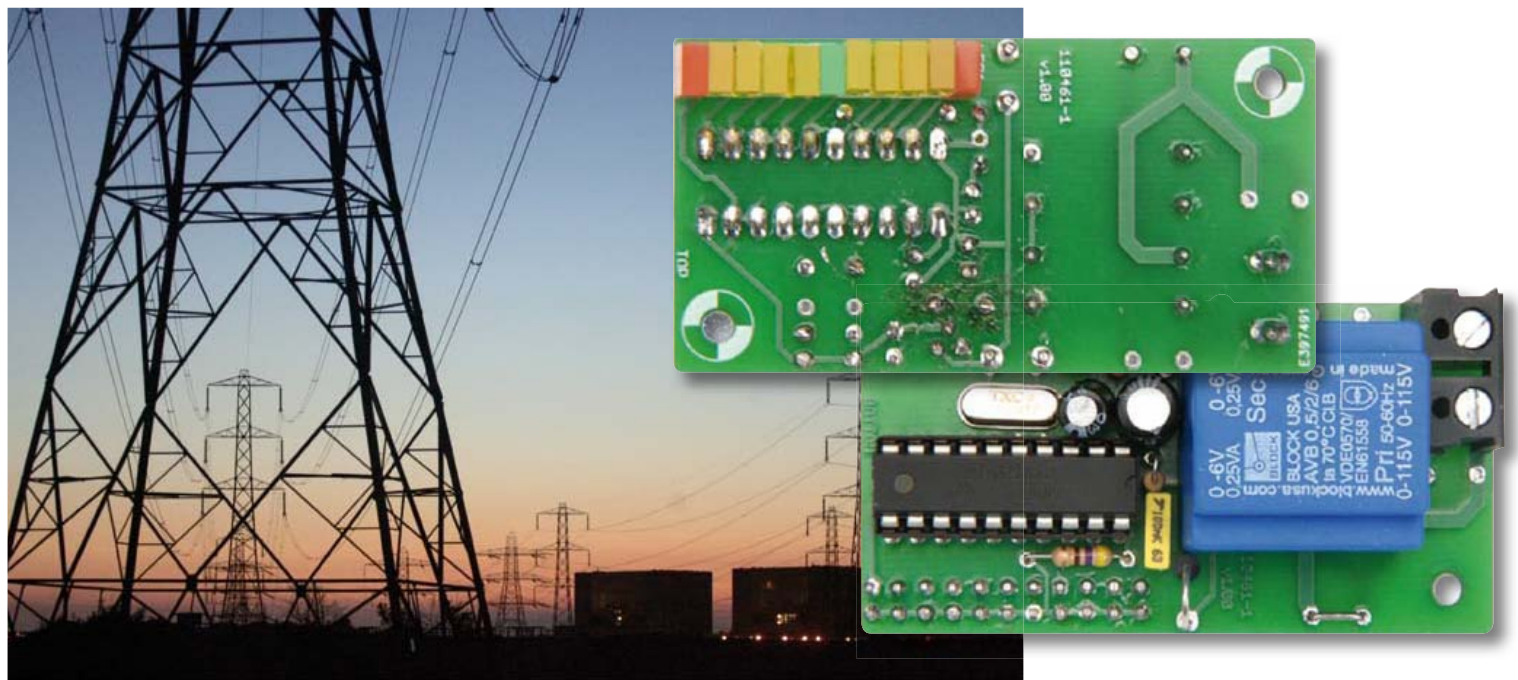
www.element14.com

Places are limited. REGISTER NOW at www.elektor.com/webinars!



Grid Frequency Monitor

Tiny fluctuations evidenced by LEDs



Measuring the AC powerline frequency is a proxy for determining the match between the supply of and demand for power in the national grid and therefore an important indicator of the status of the AC line supply. The monitoring device described here detects tiny deviations in frequency with a range of just ± 0.2 Hz, allowing you to keep an eye on the load on the grid from any convenient socket.

By Dieter Laues (Germany)

The AC line frequency in Europe is nominally 50 Hz; in the United States and certain other countries it is 60 Hz. By measuring the deviation of the frequency of the AC supply from this nominal value it is possible to get an idea of how well (or how badly) the grid is keeping up with the demand placed on it. If available supply outstrips demand the frequency will rise, while if demand outstrips supply

the frequency will fall. Normally these deviations are of less than 0.2 Hz (50 Hz ± 0.4 %), and this is the range covered by our instrument.

A balancing act

The grid has only a very limited capacity for storing electrical energy. This means that in practice supply and demand of power must be kept very accurately balanced. Even short-term imbalances in supply and demand can lead to variations in the AC line frequency and voltage. The national (or state) grid

therefore needs careful control to maintain frequency stability and power balance. [1].

The AC grid frequency [2] is determined by the rotational speed of the (synchronous) generators that supply it. In most types of power station the drive power of the generators can only be varied within fairly narrow limits, and so excessive changes in load lead directly to variations in rotational speed and hence output frequency. If the load is too high, the generator slows down and the output frequency falls; if the load is

Elektor Products & Services

- Printed circuit board # 110461-1
- Ready-programmed microcontroller, 50 Hz firmware: # 110461-41
- Ready-programmed microcontroller, 60 Hz firmware: # 110461-42

- Free download of circuit board layout (PDF), file # 110461-1.zip
- Free firmware download, file # 110461-11.zip

Downloads and product information available at www.elektor.com/110461

Features

- Display range: 49.8 Hz to 50.2 Hz (50 Hz firmware version)
- Display range: 59.8 Hz to 60.2 Hz (60 Hz firmware version)
- Accuracy: ± 0.025 Hz
- LED readout with 11 LEDs
- No calibration required
- No SMD components
- Low component count
- Suitable for 230 V and 115 V operation

too low, the generator speeds up and the frequency rises.

Millihertz, not megahertz

Day-to-day encounters with PCs, mobile and RF devices have inured us to megahertz (MHz) and gigahertz (GHz), but in the area of power generation technology we more often deal in millihertz (mHz). In the ‘continental synchronous area’, which includes most of mainland Europe, the maximum allowed frequency deviation under normal circumstances is ± 50 mHz ($\pm 0.1\%$ of 50 Hz); in the UK the national grid normally operates within ± 200 mHz of the nominal frequency. (The ‘interconnector’ that links France and the UK uses DC.) All large power stations include control equipment that automatically try to stabilise the frequency under varying load conditions.

There is a series of measures in place if there should be a large deviation in national grid frequency. Reserve services with different

response times are available to fill short-falls in supply that would otherwise lead to unacceptable drops in frequency. Also, many generation units, including wind turbines, are designed to disconnect themselves completely from the grid for safety reasons if the frequency falls too low. Reasons for such events include unpredicted spikes in demand or the failure of a generation unit or an interconnector. If the reserve services are not sufficient, it is possible to drop the AC line voltage slightly to reduce power demand, and, as a last resort, start to disconnect customers from the distribution network so that the rest of the network can continue to function as well as it can.

Blackouts

The national (or state) grid is a much more dynamic and complex network than might

appear at first glance. A serious fault or the loss of a main transmission line can, if circumstances so conspire, lead to a chain reaction that shuts down a significant part of the country or state. Reports on blackouts that have occurred make fascinating reading and give a good feel for the complexity of the system. The engineering effort and technological detail involved in providing a reliable AC power supply is astonishing.

All sockets are created equal

All generators and inverters connected to the national grid run synchronously with each other. This means that the frequency measured at any point on the grid will be the same: anywhere in the country, any socket in any house can be used to make measurements. Although the phase and

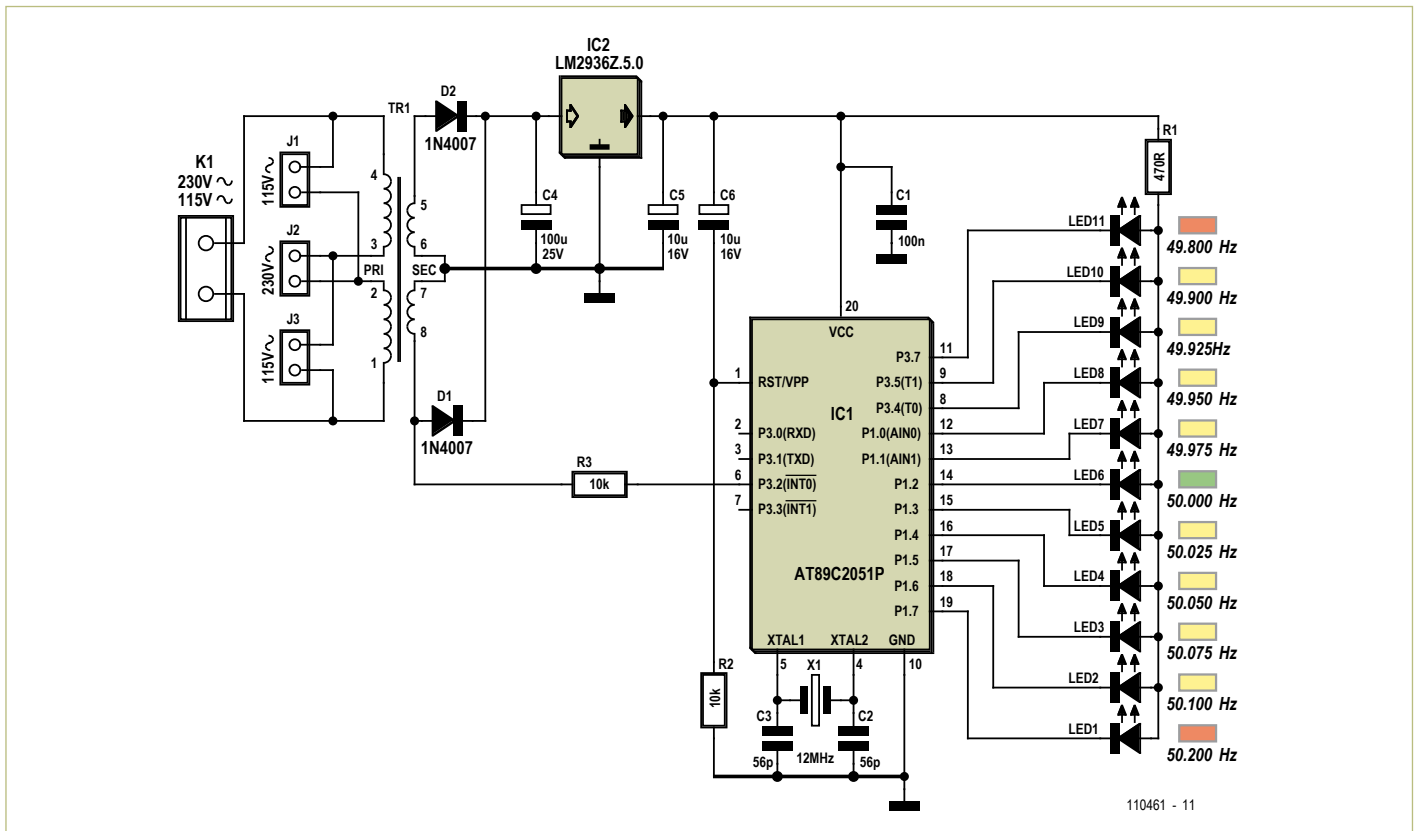


Figure 1. The AC grid frequency meter consists of little more than a power supply, a microcontroller, and a few LEDs.

COMPONENT LIST

Resistors

R1 = 470Ω
R2,R3 = 10kΩ

Capacitors

C1 = 100nF MKT, 5mm pitch
C2,C3 = 56pF ceramic, 2.5mm pitch
C4 = 100µF 25V, radial, 2.5mm pitch
C5,C6 = 10µF 16V, radial, 2.5mm pitch

Semiconductors

D1,D2 = 1N4007
IC1 = AT89C2051-24PU (Atmel), programmed, Elektor # 110461-41 (50Hz) or 110461-42 (60Hz)
IC2 = LM2936Z-5.0
LED1, LED11 = LED, red, low-current, 5x2 mm

LED2,LED3,LED4,LED5,LED7,LED8,LED9,
LED10 = LED, yellow, low-current, 5x2 mm
LED6 = LED, green, low-current, 5x2 mm

Miscellaneous

K1 = 2-way PCB terminal block, 7.5mm pitch
J1,J2,J3 = 2-pin pinheader, 5mm pitch, with jumper (or wire link), see text
TR1 = AC power transformer, 0.5VA, 2x115V primary, 2x6V secondary, e.g. Block type AVB0,5/2/6)
X1 = 12MHz quartz crystal, HC-49S case
PCB # 110461-1
AC adapter case, ABS, approved, to suit local power outlet.

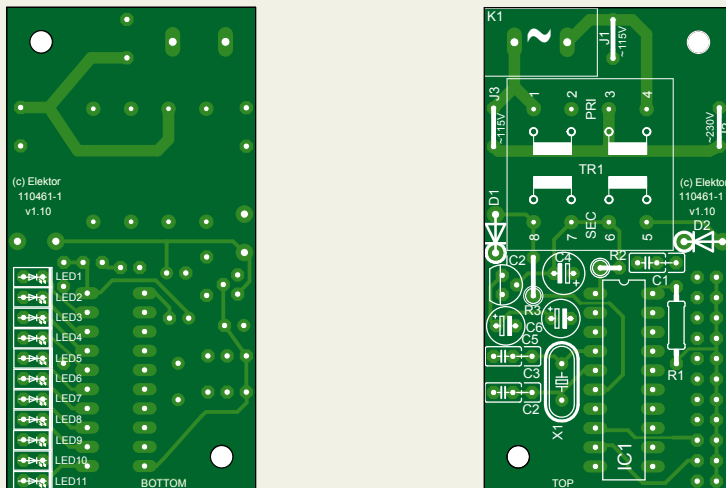


Figure 2. The printed circuit board avoids the use of SMD components and requires no adjustments.

voltage might differ between sockets in St Ives and Stromness, the frequencies will be identical. The same is true of any two ordinary sockets within the European continental synchronous area, but this area is not synchronised with the UK.

Display

The problem, then, is to measure and display small deviations in frequency, in the millihertz range. In theory it is possible to measure the AC grid frequency with a suitable multimeter or frequency counter; however, in the case of cheaper units at any rate, the digits we are interested in fall beyond the right-hand edge of the display. Professional devices capable of greater precision carry three-figure price tags.

The circuit described here shows the instantaneous frequency value between 49.8 Hz

and 50.2 Hz using a row of LEDs. The electronic unit is a replacement for the mechanical vibrating reed frequency meter, but with one hundred times better resolution and at one hundredth of the price.

The hardware

As usual these days, the circuit (Figure 1) primarily consists of a microcontroller. In this case we use the familiar Atmel AT89C2051 to carry out the frequency measurement and to drive the eleven LEDs. The microcontroller firmware is compiled using BASCOM, and two versions are available: one designed for 50 Hz operation and one for 60 Hz.

Also on the AC line side, the circuit has been designed for use either at 230 V/50 Hz or 115 V/60 Hz. To this end we use a small (0.5 VA) power transformer with two 115 V primary windings. Jumper J2 puts the two

windings in series for 230 V operation and J1 and J3 put the windings in parallel for 115 V operation. Having two 6 V secondary windings means that we can make a full-wave rectifier using just two diodes (D1 and D2), the centre tap of the two windings forming the ground point. Voltage regulator IC2 delivers the 5 V supply for the microcontroller (IC1). The microcontroller also receives the 6 V AC signal from one of the transformer's secondary windings on port pin P3.2 (pin 6) via R3, which it uses to measure the AC grid supply frequency. In conjunction with the internal protection diodes on the port pin this 10 kΩ resistor limits the current into or out of the pin to less than 1 mA. According to Atmel this is a reliable as well as a simple method of obtaining a square wave from an alternating voltage: see [3].

The microcontroller derives its clock from the 12 MHz crystal, while R2 and C6 provide a power-on reset pulse. Port pins P1.0 to P1.7, P3.4, P3.5 and P3.7 directly drive the LEDs. Since only one LED lights at a time, a single common series current limiting resistor (R1) suffices. The brightness of the LED can be altered by changing the value of this resistor.

You can of course use whatever colours of LEDs you like. In the interests of clarity we populated our prototype as shown in the circuit diagram: a green LED for the exact nominal value of 50 Hz, yellow for deviations of up to ±0.2 Hz and red at the extremities of the displayed range. The scaling used for the display includes a magnifying effect: the resolution for the nine innermost LEDs (green and yellow) is 0.025 Hz while the frequency step to the outer pair of red LEDs is 0.1 Hz (see Table 1).

The software

When power is applied each LED is turned on briefly: this provides a quick check that the microcontroller is running properly and that all the LEDs work. The firmware then drops into the measurement routine. The microcontroller carries out the frequency measurement by measuring the time interval between successive negative-going edges on the INT0 input. At the nominal 50 Hz frequency this period will be

Table 1. The LED scale

LED colour	red	yellow	yellow	yellow	yellow	green	yellow	yellow	yellow	yellow	red
Deviation from 50 Hz in mHz	<-200	-100	-75	-50	-25	0	+25	+50	+75	+100	>+200

exactly 20 000 μ s. With a crystal frequency of 12 MHz and reading the timer 'on the fly' we can obtain a period resolution of ± 1 μ s, which corresponds to 2.5 mHz at 50 Hz.

In order to avoid the effect of possible interference on the AC powerline input we use two filters implemented in software. Frequency readings of less than 45 Hz or more than 55 Hz are ignored; and furthermore, an average is taken over fifty consecutive readings. This approach practically eliminates errors due to jitter.

The LED display is updated once a second, and a brief pulse is emitted on port pin P3.3 each time the display is updated. If desired an LED can be attached to this pin also: it will flash once every 50 readings, that is, once per second.

Construction and test

The printed circuit board with component mounting plan shown in **Figure 2** is free both of SMD components (hurray!) and of adjustments (cheers 4 that). Also, the total component cost has been kept low. To keep things simple, the microcontroller is available from the Elektor Shop as a ready-programmed device. It is best to use an IC socket for the microcontroller and to test the circuit first with the device not fitted. It is of course essential to observe correct polarity when fitting the diodes and electrolytic capacitors. Using a board-mounted transformer keeps the AC power wiring simple.

Before applying power to the circuit check it over carefully for mistakes. Check also the jumper positions: for 115 V operation fit both J1 and J3 **but not J2** (otherwise a short circuit will be introduced), and for 230 V operation fit **only J2**. When using the circuit with a 115 V/60 Hz input it is also necessary to program the microcontroller with the 60 Hz version of the firmware.

When everything has been checked power can be applied. Since AC line voltages are present at the input to the board, this

should only be attempted by a suitably-qualified technician.

With AC power applied to the board a DC voltage of about 8.5 V should appear across C4, while the voltage across C5 should be very close to 5 V. Assuming this is the case, disconnect the circuit from the AC socket and fit the (programmed) microcontroller. When power is again applied the LEDs should light briefly and then the middle (green) LED should light, indicating an AC power frequency of 50 Hz (or 60 Hz in the case of the 60 Hz firmware version).

If everything seems to be working the unit can be fitted into a fully-insulated plastic enclosure with a moulded-in AC line plug, of the type used for plug-in power supplies. It must be impossible to touch any part of the circuit from the outside. The result is a device that is both electrically safe and easy to use: simply plug it into an AC power outlet and observe the reading on the LEDs.

The unit in practice

In normal situations it will be observed that the power grid frequency remains very stable, deviating from its nominal value by only a few tens of millihertz. The green LED will therefore be on for a lot of the time. In general there will be certain regular fluctuations observed, recurring every morning and afternoon.

As the proportion of energy generated from weather-dependent sources such as sun and wind increases, unpredictable shortfalls or excesses of supply are expected to occur more frequently. These in turn will lead to greater frequency fluctuations.

As an interesting aside, it was the case in Germany until recently that solar panel installations taking advantage of a grid 'feed-in' tariff had to be designed to disconnect themselves if they detected that the grid frequency was above a certain threshold. This gave rise to the so-called '50.2 Hz problem' [4]: the sun comes out, several

gigawatts of solar power are dumped into the grid, the frequency rises... and suddenly all those gigawatts of power simultaneously disconnect themselves from the grid! The back-up services cannot cope with such a large sudden loss of generation, and so the frequency falls sharply. The solar installations observe this fall, reconnect themselves (typically almost simultaneously) and the cycle repeats. The problem is being solved by randomising the frequency thresholds for such generators, with a longer-term move towards having the units smoothly reduce their feed-in power as frequency rises.

It is possible to guess at the degree of load fluctuation on the grid by observing frequency changes. You may also be able to observe sudden drops in supply, for example when a fault detected in a power station trips its connection to the grid. Real-time demand graphs can be found at [5], along with information about power transfers on the interconnectors between Northern Ireland and Britain, between France and Britain, and (due for completion in late 2011) between the Netherlands and Britain. There is also a real-time frequency graph available that you can compare against the results from your meter.

(110461)

Internet Links

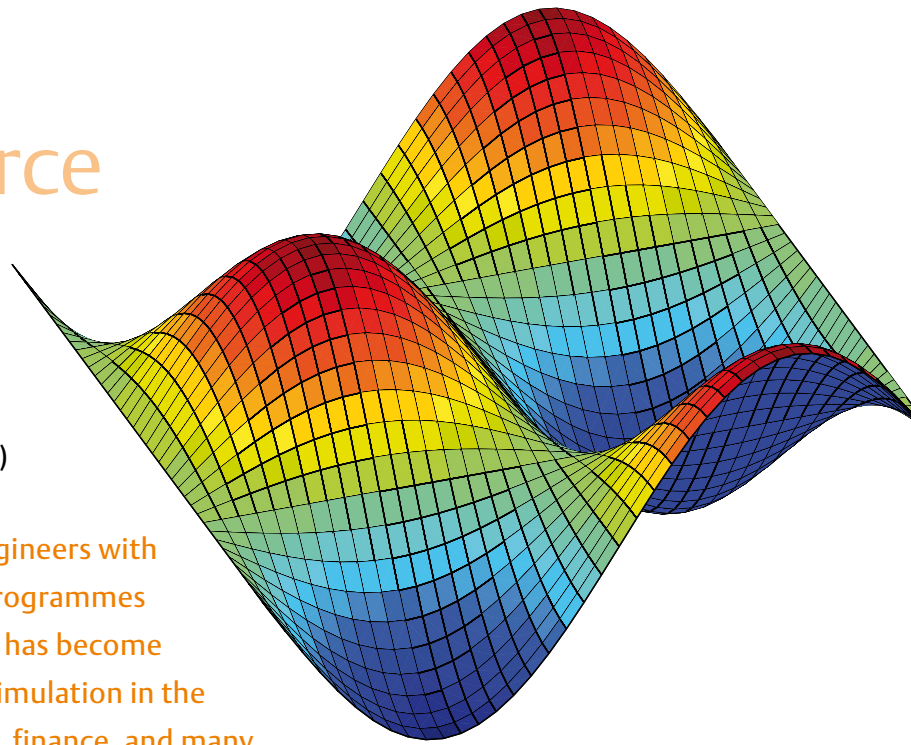
- [1] http://en.wikipedia.org/wiki/Operating_reserve
- [2] http://en.wikipedia.org/wiki/Mains_frequency
- [3] www.atmel.com/dyn/resources/prod_documents/doc2508.pdf
- [4] www.vde.com/en/fnn/pages/50-2-hz-study.aspx
- [5] www.nationalgrid.com/uk/Electricity/Data/Realtime/

Scilab

#1 for Open Source Numerical Calculation

By Vincent Couvert, Bruno Jofret, and Julie Paul (France)

Numerical calculation software provides engineers with a collection of design and study tools and programmes for simulation and modelling. This software has become indispensable for industry, which employs simulation in the automotive, aeronautics, energy, chemistry, finance, and many other fields. It enables industry to reduce, indeed sometimes even avoid, expensive testing – which is moreover often difficult to perform in a real situation.



In order to remain in the race for ever-greater competitiveness and respond to the widespread and growing drive to reduce costs, the industrial world is taking increasing interest in this software, which is proving a very serious competitor for the proprietary software they have been in the habit of using. Today, opting for open source software is no longer something that concerns only the general public for Internet browsing or word processing; as an example, OpenOffice.org has been adopted by numerous French public administrative bodies, and now software for the world of industry is involved as well.

Supported by a consortium of industrial users and already in use around the world, Scilab [1] now represents a credible alternative to Matlab [2]. Apart from the not-insignificant cost benefit, the access to and complete control over the source code often constitute a decisive argument in its being adopted by numerous users, including the strategic sectors of defence and aeronautics. The academic world and education have already widely adopted Scilab, which in June 2011 was recognized as being of high educational value by the French Ministry of Education.

The Scilab software in practice

Scilab is a piece of freeware distributed under a CeCILL licence (GPL compatible). It is available for the commonest operating systems (Windows, Mac, and Linux) and can be downloaded free from [1]. Scilab is a complete environment, open and expandable, for calculation and visualisation. Based on matrix calculation, the software incorporates hundreds of mathematical functions and a powerful programming language. It offers possibilities for connection with

other programs written in various languages (C, C++, Java). Scilab's syntax is comparable to that of Matlab on numerous points, though it's not 100 % compatible. Its functional spectrum is also very large, and users can add numerous external modules to it for simulation, graphic visualization, optimization, statistics, R&D'ing systems and controls, signal processing, etc.

Like much free software, Scilab is interoperable. Users can adapt it to their needs and augment its native functions. For example, the gateway between National Instruments' LabVIEW [3] and Scilab allows comprehensive, high-performance data processing. Users write their Scilab scripts directly in LabVIEW and call Scilab for analysing and visualizing their data.

An integrated environment

Scilab offers an ergonomic, integrated work environment that is easy to get to grips with and use. Its console makes for easy use, displaying command lines and results. Its text editor and advanced functions allow you to save your programs and then recover your working environment next time it is run. It also provides total interaction with the console, for example, allowing users to execute all or part of a file while it is being edited.

Advanced visualization functions

2D and 3D graphics functions are included for visualizing, annotating, and exporting data. These offer numerous ways to create and customize various types of plots and diagrams. In this way, the results calculated by the user in the Scilab console can be visualized interactively.

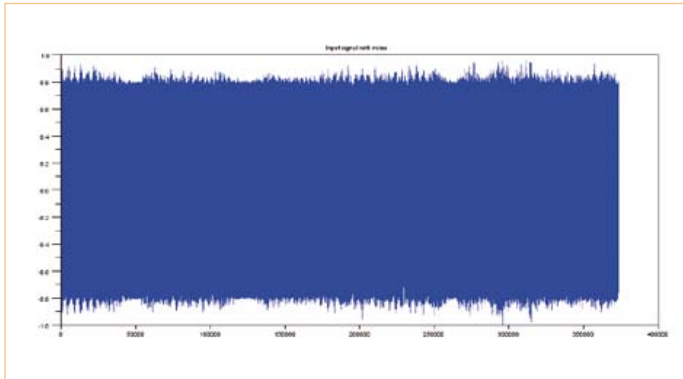


Figure 1. The input signal.

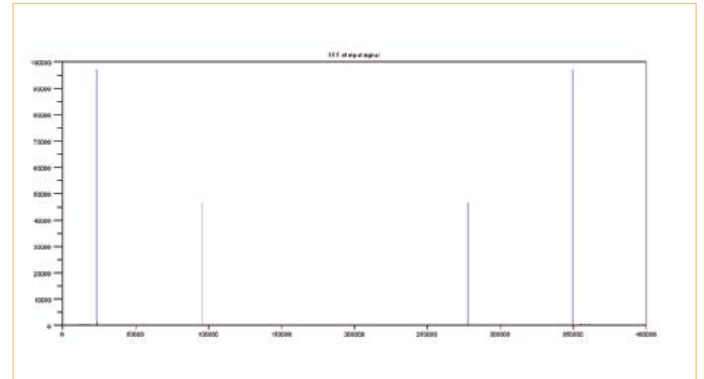


Figure 2. The Fourier transform of the input signal shows that it contains noise at two distinct frequencies.

To give a more detailed view of the software's functions, we're going to show you two examples of using it in the field of signal processing.

Example 1: Digital filtering using Scilab

In this example, Scilab is used to filter a signal coming from a WAV sound file. All the tools required to perform this task are included within Scilab: file read/write, fast Fourier transform calculation, calculation and application of digital filters.

The sound file, which we've placed in Scilab's current directory (consultable and modifiable by calling Scilab's `cd` function) and called "NoisySignal.wav" [4], is loaded using the function `loadwave` which returns the sound data in the form of a vector, together with some information about the file (sampling frequency, etc.)

In the text editor, we enter the following lines and then execute them ('Run' menu — as far as the cursor with echo):

```
stacksize("max"); // Increase memory for Scilab
[noisySignal, noisySignalInfo] =
loadwave("NoisySignal.wav");
sampleFrequency = infosSignal(3)
samples = infosSignal($)
```

Which displays in the console:

```
-->sampleFrequency = noisySignalInfo(3)
sampleFrequency =

    22050.

-->samples = noisySignalInfo($)
samples =

    373380.
```

The results obtained tell us that the signal has been sampled at a frequency of 22,050 Hz and that our signal contains 373,380 samples (values). The `plot` function lets us display this signal (Figure 1):

```
plot(noisySignal)
xtitle("Input signal with noise"); // Title of the
graph
```

In order to find out the frequencies that are masking the wanted signal, we calculate the Fourier Transform of the signal using the `fft` function, then its modulus using the `abs` function:

```
FftOfNoisySignal = abs(fft(noisySignal));
scf(); // Open a new graphics window
plot(FftOfNoisySignal);
xtitle("FFT of input signal")
```

Using the `plot` function, we obtain the graph in Figure 2. This shows that this symmetrical signal has noise at two distinct frequencies. Hence it is going to take two successive filterings to recover the wanted signal. Precise identification of the noise frequencies is made easier in Scilab with its numerous matrix calculation features: mathematical operations, identification of the maximum value and of its position within the data...

Executing the following code from the text editor lets us define the frequency corresponding to the peak:

```
// The FFT is symmetrical, we only keep the first
half of the samples
frequencies = sampleFrequency*(0:(samples/2))/
samples;

// Find the highest peak in the FFT to filter it out
[peakValue, peakValueIndex] = max(FftOfNoisySignal(1:
size(frequencies, "*")));
peakValueIndex
```

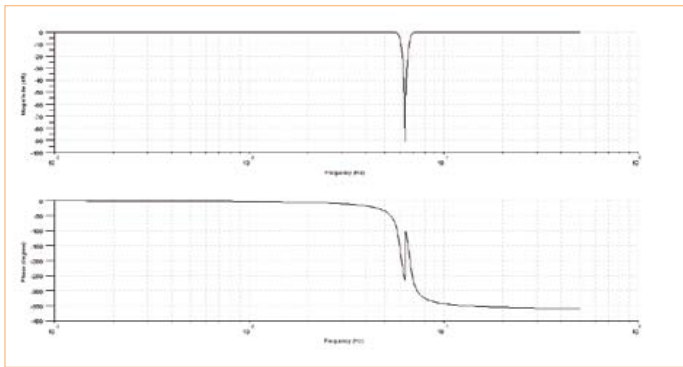


Figure 3. The Bode diagram of the IR filter which will filter out the 1397 Hz signal.

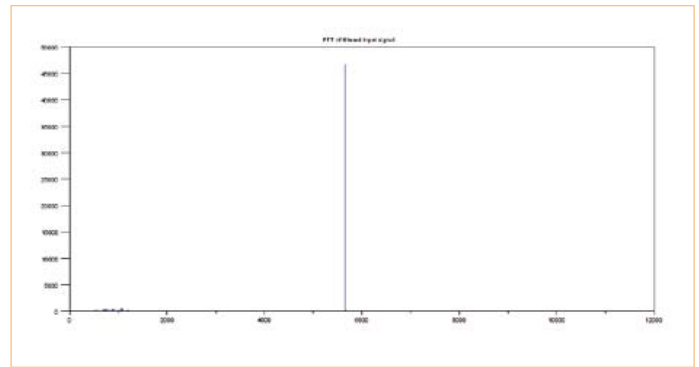


Figure 4. After filtering, the input signal now only has one interference signal.

```
frequencies(peakValueIndex)
```

Now let's verify the results in the console:

```
-->peakValueIndex
peakValueIndex =

    23657.

-->frequencies(peakValueIndex)
ans =

    1397.0079
```

The first peak represents noise at a frequency of 1397.0079 Hz. A 3rd-order Butterworth band-stop filter with a bandwidth of 200 Hz and centred on this frequency makes it possible to eliminate this noise. The transfer function of this filter is obtained using the function `iir`:

```
hz = iir(3, "sb", "butt", [frequencies(peakValueIndex)-100 frequencies(peakValueIndex)+100]/sampleFrequency, [0 0]);
```

In order to display the filter's Bode diagram simply, we use the function `bode` (Figure 3). Now all we have to do is apply this filter to the noisy signal using a single line of code:

```
// Filtering input signal
filteredSignal = filter(hz.num, hz.den, filteredSignal);
```

The signal obtained still carries noise at the second frequency, as we see from the Fourier transform in Figure 4 (calculated using the same method as before). This graph confirms that the first peak has indeed been eliminated. Now we need to filter the second peak with the help of another band-stop filter (for example, a 3rd-order type I Chebyshev filter) centred on the frequency corresponding to this peak, using the same technique as before:

```
// Find the second peak in the FFT to filter it out
[peakValue, peakValueIndex] = max(FftOfNoisySignal(1:size(frequencies, "*")));
```

```
peakValueIndex
frequencies(peakValueIndex);
// Calculate corresponding Chebyshev filter
hzFiltre2 = iir(3, "sb", "cheb1", [frequencies(peakValueIndex)-100 frequencies(peakValueIndex)+100]/sampleFrequency, [0.01 0]);
// Filter the filtered input signal
filteredSignal = filter(hz.num, hz.den, filteredSignal);
```

We display the result (Figure 5):

```
scf();
plot(filteredSignal)
xtitle("Input signal filtered twice")
```

The original signal is now completely filtered and a final calculation of the Fourier transform of the signal shows that both noise frequencies that were present in our original signal have indeed disappeared (Figure 6):

```
FftOfFilteredSignal = abs(fft(filteredSignal));
scf();
plot(frequencies, FftOfFilteredSignal(1:size(frequencies, "*")));
xtitle("FFT of input signal filtered twice")
```

We save our result in a WAV file:

```
savewave("SoundFiltered.wav", filteredSignal)
```

Example 2 – Edge detection

In the field of image processing, and more broadly computer vision, edge detection is one of the first steps required upstream of the more complex algorithms used for detecting objects, detecting faces (face tracking), and so on. All we have to do is represent the image in the form of a matrix of values (usually the grey levels) and to apply the desired processing to this matrix. Here, Scilab plays a dual role, as it lets us both display the image and the changes to it during the processing, as well as manipulate the data using the matrix calculation capabilities available.

This example applies a first-order method (gradient) implemented using Prewitt, Sobel, and Scharr filters.

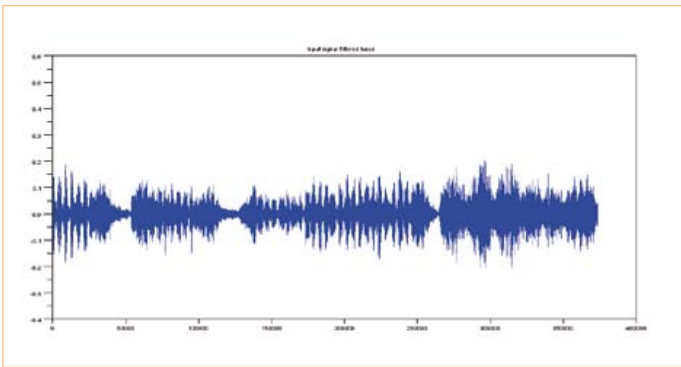


Figure 5. The filtering has made it possible to eliminate both the interfering signals.

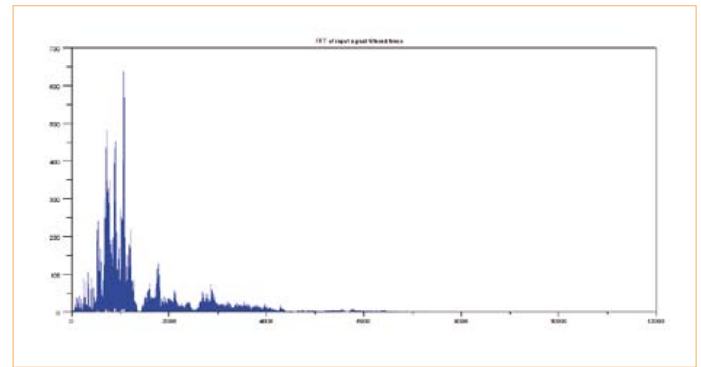


Figure 6. The Fourier transform of the filtered input signal.

The image in the PGM format ('portable graymap', a greyscale image) which we've named "Scilab.pgm" [4] is loaded using the function `readimage` [4]. This function will open the file, read the size of the image and then the grey levels associated with each pixel (values between 0 and 255), returning the image in the form of a matrix.

```
stacksize("max"); // Increase memory allocated to
Scilab
gray_m = readimage("Scilab.pgm");
```

For simplicity, in the rest of this example, we're going to use a function `showImage` to display all the future transformations in levels of grey. This function is based on one of Scilab's native functions, `Matplot` [5], which allows a matrix to be displayed. By using the corresponding greyscale table, we can display the image:

```
function []=showImage(imageMatrix)
    f = Figure();
    f.color_map = graycolormap(255);
    f.background = -2;
    Matplot(imageMatrix);
endfunction
```

```
showImage(gray_m);
```

The image loaded into Scilab is displayed in a graphics window (Figure 7).

Edge detection on a greyscale image is performed using calculations on a 2D matrix. It is possible to work on N -dimension matrices in order to represent colour images (RGB, RGBA, CMYK, HSV, etc.) The processing to be applied will be applied to each channel, or indeed combined between them to obtain other results.

The next step in detecting edges consists in applying smoothing to reduce the noise in the original image and obtain better results in the final detection. Scilab lets us define the smoothing function by using a convolution product and a Gaussian 2D matrix. This takes as a parameter the matrix representing the image and returns another matrix representing the smoothed image.

Two rows of pixels are artificially added around the edge of the image so as to be able to implement the convolution product with the help of the function `sum` and a product of matrices. These func-

tions are optimized in Scilab and perform better than if each of the operations were performed element by element.

```
function N=blurr(P)
    N = zeros(P);
    P2 = [zeros(1,size(P,"c")); zeros(1,size(P,"c"))];
    P; zeros(1,size(P,"c")); zeros(1,size(P,"c"))]
    P2 = [zeros(size(P2,"r"), 1), zeros(size(P2,"r"), 1),
    P2, zeros(size(P2,"r"), 1), zeros(size(P2,"r"), 1)]
    K = 1/159 * [2 4 5 4 2
    4 9 12 9 4
    5 12 15 12 5
    4 9 12 9 4
    2 4 5 4 2];

    for x=3:(size(P2,"r") - 2)
        for y=3:(size(P2,"c") - 2)
            r = 0;
            N(x-2,y-2) = sum(K .* P2(x-2:x+2,
            y-2:y+2));
        end
    end
endfunction
```

A more naïve implementation of the convolution product, such as exists in the literature, would make it possible to obtain numerically equivalent results, but with much longer calculation times.

```
function N=dummy_blurr(P)
    N = zeros(P);
    K = 1/159 * [2 4 5 4 2
    4 9 12 9 4
    5 12 15 12 5
    4 9 12 9 4
    2 4 5 4 2];
    for x=1:size(P,"r")
        for y=1:size(P,"c")
            r = 0;
            for i = -2:2
                for j = -2:2
                    if (x + i > 0 & x + i <= size(P,"r") & y + j > 0 & y + j <= size(P,"c"))
```




Figure 7. The original greyscale image.



Figure 8. The original image smoothed to make it easier to detect the edges.

```

r = r + K(i+3, j+3) * P(x+i, y+j)
    end
end
end
N(x,y) = r;
end
end
endfunction
    
```

The smoothed image thus obtained (**Figure 8**), which to the naked eye appears blurred, will be used for the gradient calculations which will bring out the edges in this image.

For these gradient calculations, we are using Sobel (**Figure 9**) and Prewitt (**Figure 10**) filters. Both these filters are achieved by way of a convolution product used for calculating a gradient.

```

function N=convol2d(K, P)
    N = zeros(P);
    P2 = [zeros(1,size(P,"c")); P;
zeros(1,size(P,"c"))]
    P2 = [zeros(size(P2, "r"), 1), P2, zeros(size(P2,
"r"), 1)]
    for x=2:(size(P2, "r") - 1)
        for y=2:(size(P2, "c") - 1)
            r = 0;
            N(x-1,y-1) = sum(K .* P2(x-1:x+1,
y-1:y+1));
        end
    end
endfunction
    
```

Scilab is not the only free, open source numerical calculation program

Here are three other examples that deserve a mention:

– **Octave** is a structured interpreted programming language like C that accepts numerous constructions from the standard C library. It can be extended to accept Unix functions and system calls, and functions written in C++ can be used. For the majority of commands, the syntax is the same as for MATLAB and careful programming makes it possible to run scripts under both Octave and MATLAB. (source: Wikipedia)

www.gnu.org/software/octave/

– **FreeMat** is a computerized calculation environment and a programming language, in the form of free software, relatively compati-

ble at source level with Matlab and Octave. It can be easily interfaced with external code in C, C++, and Fortran, it offers the possibility of developing parallel distributed algorithms, and it has some capabilities for volume rendering and 3D visualization. The current version 4.0 dates from October 2009. (source: Wikipedia)

<http://freemat.sourceforge.net>

– **JMathLib** is presented as a Java clone of Scilab, Octave, FreeMat, and Matlab. Like FreeMat, this project seems to be less active than Scilab and Octave. The current version 0.9.4 dates from February 2009.

www.jmathlib.de/

For Sobel:

```
GX = convol2d([-1 0 1 ; -2 0 2 ; -1 0 1], gray_m);
GY = convol2d([-1 -2 -1 ; 0 0 0 ; 1 2 1], gray_m);
contourSobel = sqrt(GX.^2+GY.^2);
showImage(contourSobel);
```

For Prewitt:

```
GX = convol2d([-1 0 1 ; -1 0 1 ; -1 0 1], gray_m);
GY = convol2d([-1 -1 -1 ; 0 0 0 ; 1 1 1], gray_m);
contourPrewitt = sqrt(GX.^2+GY.^2);
showImage(contourPrewitt);
```

Other formulæ give more satisfactory results by using a different convolution model: Costella, Robert Cross, and Scharr. It's easy to implement these in Scilab thanks to a language that is simple and fairly close to mathematics (no variable declaration, no typing, no memory allocation, etc.) For example, the Scharr filter (Figure 11) is implemented via the following lines:

```
GX = convol2d([3 0 -3 ; 10 0 -10 ; 3 0 -3], gray_m);
GY = convol2d([3 10 3 ; 0 0 0 ; -3 -10 -3], gray_m);
contourScharr = sqrt(GX.^2+GY.^2);
showImage(contourScharr);
```

This matrix may be used as the basis for other shape detection algorithms. This is just one initial step in processing the image. We can go further by performing additional filtering on threshold values. The most pronounced edges corresponding to the lightest areas would be retained and the others eliminated. The matrix thus obtained would contain only 0s and 1s for each pixel recognized as being part of an edge.

There are countless possible examples of the use of Scilab across very diverse fields of application. A forthcoming issue of Elektor will be carrying an article on Xcos, Scilab's tool for modelling and simulating hybrid dynamic systems – the equivalent of Matlab's Simulink.

(110491)

Internet Links

- [1] Scilab: www.scilab.org
- [2] Matlab: www.mathworks.com/matlab
- [3] LabVIEW: www.ni.com/labview
- [4] Files for the examples shown: www.elektor.com/110491
- [5] Help with the Matplot function:
http://help.scilab.org/docs/current/fr_FR/Matplot.html



Figure 9. The edges as revealed by the Sobel filter.



Figure 10. The edges found by the Prewitt filter.

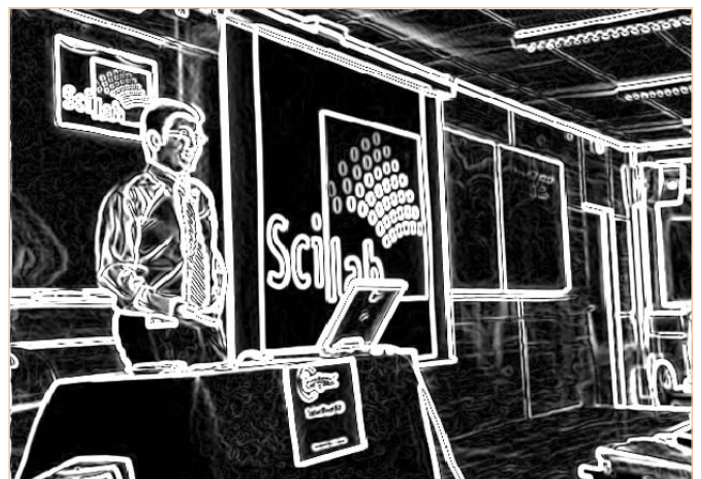


Figure 11. The edges according to the Scharr filter.

• **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 today!



audioXpress

Bat—batter—best!

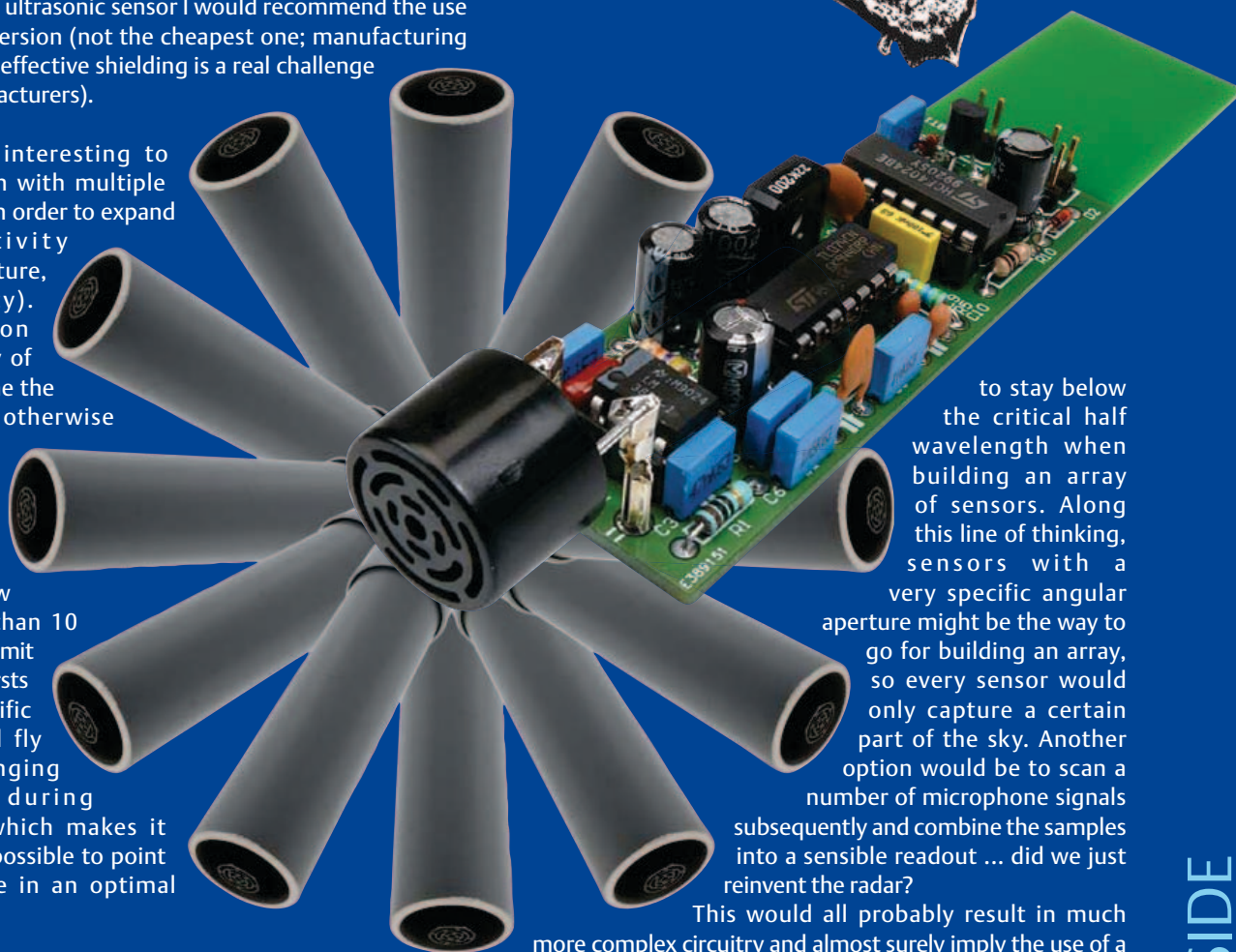
By Thijs Beckers (Elektor UK Editorial/ Labs)

Reader Matthias Schwarzwald (Germany) kindly sent us an interesting remark on the Simple Bat Detector published in the November 2011 edition [1]. Although he really enjoyed the SMT-free circuit — many bat fans are not particularly educated in electronics — he feels some components should be of higher quality.

“For instance, the 40 kHz ultrasonic sensor is only capable of detecting about half of the population of bats living in my country. There are many species ‘emitting’ at other frequencies, some as ‘low’ as 20 kHz. In this particular instance a suitable electret microphone capsule would be a better choice. When using an ultrasonic sensor I would recommend the use of a shielded version (not the cheapest one; manufacturing and testing an effective shielding is a real challenge to most manufacturers).

It would be interesting to have a version with multiple microphones in order to expand the sensitivity (angular aperture, or directivity). Depending on the directivity of the microphone the range would otherwise be reduced from the theoretical 30 meters (100 feet) to just a few meters (less than 10 feet). Bats transmit their sound bursts in a very specific direction and fly an ever-changing trajectory during their hunt, which makes it practically impossible to point a microphone in an optimal direction.”

We think Matthias has a point. During the prototyping phase fellow lab worker Ton Giesberts, who has been manoeuvring this project through our lab, had already thought about adding a resistor to the circuit to optionally supply an electret microphone with a bias voltage. Also, the use of multiple sensors had come to mind. The problem with using a ‘sensor array’ though, would be summing the signal correctly — without introducing phase problems. The wavelength of a 40 kHz signal is about 8.25 mm, so it is practically impossible



to stay below the critical half wavelength when building an array of sensors. Along this line of thinking, sensors with a very specific angular aperture might be the way to go for building an array, so every sensor would only capture a certain part of the sky. Another option would be to scan a number of microphone signals subsequently and combine the samples into a sensible readout ... did we just reinvent the radar?

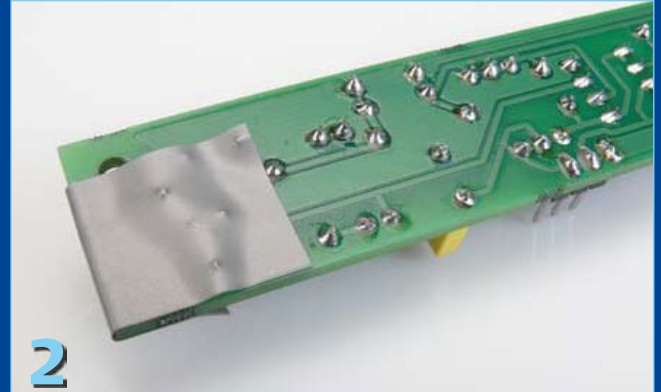
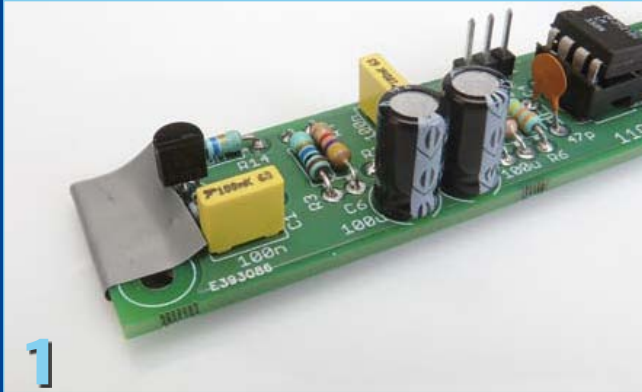
This would all probably result in much more complex circuitry and almost surely imply the use of a microcontroller — defeating the simplicity of the project. So these ideas were quickly abandoned. That doesn't mean it wouldn't be interesting to engage in such a complicated design challenge. So if there is somebody out there who's dying for something to put his teeth — sorry, soldering iron — in, let us know!

(120021)

Internet Link

[1] www.elektor.com/110550

Radiation meter:



By Thijs Beckers (Elektor UK Editorial / Labs)

The article about the Improved Radiation Meter (Elektor November 2011, [1]) contains a basic description of how the sensor — type BPW34 — has to be shielded from ambient light. Having noticed some confusion among our readers we will explain it again with the aid of some photos. The **BPW34**, which is suitable for the measurement of **gamma radiation**, can be completely enclosed; gamma radiation is strong enough to pass through various materials and still be measurable afterwards. The shielding from ambient light can be done in several ways. We will show two here:

1. Shielding with aluminium foil.

The sensor is covered with a piece of kitchen aluminium foil. To prevent a short-circuit, we first put a piece of insulation tape on the circuit board at the location of the sensor (**Photos 1 and 2**). This has the added benefit of blocking the light that would otherwise pass through the board and onto the back of the sensor. Now we fit the sensor (**Photo 3**). Make sure you get the polarity correct (this is also indicated on the circuit board in the kit). The cathode on the BPW34 is indicated by a small tab on the pin, see **Photo 4**.

You can now cover the sensor with a piece of aluminium foil. Use a piece of foil that's large enough for it to be connected to

the ground terminal on the circuit board. Make sure that the foil has no minuscule holes and that the foil is 100% light-proof. Also ensure that the space between the foil and the sensor is as small as possible. Otherwise the sensor will also react to loud sounds, and we want to avoid that of course (**Photo 5**). Finally you can cover the entire circuit board with aluminium foil, so you keep external influences out of the circuit as much as possible (**Photo 6**). When doing so, make sure that you are not creating a short-circuit somewhere! If you've worked carefully the sensor is ready for making measurements.

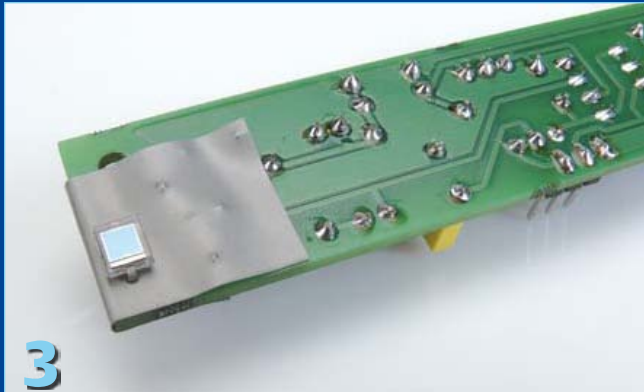
2. Shielding in a tin.

In this case we place the entire amplifier board within a shielding and light-proof can (**Photo 7**). There are several options for this, consider a biscuit tin or the blue tin in which the hand lotion of a famous brand is sold. 'Old-fashioned' spice or tea and coffee tins are also a good choice. It is best to use a tin *without* a hinge, since these hinges will often allow light to enter the tin. Mount the circuit board in the tin and ensure that there is a good connection between ground on the circuit board and the tin.

The wires connecting the amplifier board to the counter can exit between the top of the tin and under the lid. In this case there is no need to shield the sensor separately as well. In our experiments this option gave the best results and the least noise and interference.



mounting the sensor



Alpha radiation is blocked much easier than gamma radiation; a sheet of paper is already a big obstacle for alpha radiation. As already discussed in the above-mentioned article, the BPW34 is not really suitable for measuring this type of radiation. The plastic package will (for the most part) already block the radiation. Wrapping it in aluminium foil is therefore entirely out of the question. A good alternative sensor for measuring alpha radiation is the type **BPX61 (Photo 8)**. Although this photo diode is more expensive, it is supplied in a TO-39 package, which has a glass window on the light-sensitive side.

If we now carefully remove this glass window, without damaging the sensor, then the light-sensitive layer is completely exposed and the radiation can reach the sensor surface completely unimpeded (**Photo 9**).

As already mentioned, alpha radiation is easily blocked. So the radioactive material that you are using in the test must not be too far from the sensor. Tests with this diode carried out at the University of Namen (Belgium) indicated that when using a ^{239}Pu -test radiator (about 5 MeV of alpha radiation) resulted in pulses of about 200 mV at the output of the measuring amplifier.

In this way we have a very sensitive sensor in our hands that's capable of measuring alpha radiation readily. There is no need to change either the amplifier nor the pulse counter. They both will operate with either sensor. A few 'everyday' sources of

radioactive radiation are an 'old' watch with illuminated hands (mainly alpha radiation, so the glass has to be removed), a gas mantle intended for gas lamps, potassium chloride, WT20 welding electrodes, a discarded smoke detector (when it indicates that ^{241}Am is being used) and pitchblende (of uraninite). Sending radioactive materials by mail is absolutely forbidden, but there is of course no problem if you take your radiation meter to a location where there are radioactive materials. For example, in a hospital. If you meet the right staff there then they will almost certainly be interested in the 'sensor' we use in this circuit, and perhaps can be persuaded to do a test calibration of your circuit.

Incidentally, we are still experimenting ourselves with other 'sensors', such as the famous 2N3055 (**Photo 10**) and also the author of the article, Burkhard Kainka, has carried out a substantial number of experiments [2]. We will of course not withhold any good results and these may be published in a future article.

(110709-I)

Internet Links

- [1] www.elektor.com/110538
- [2] www.elektronik-labor.de/Projekte/Projekte.html



Debugging the debugger

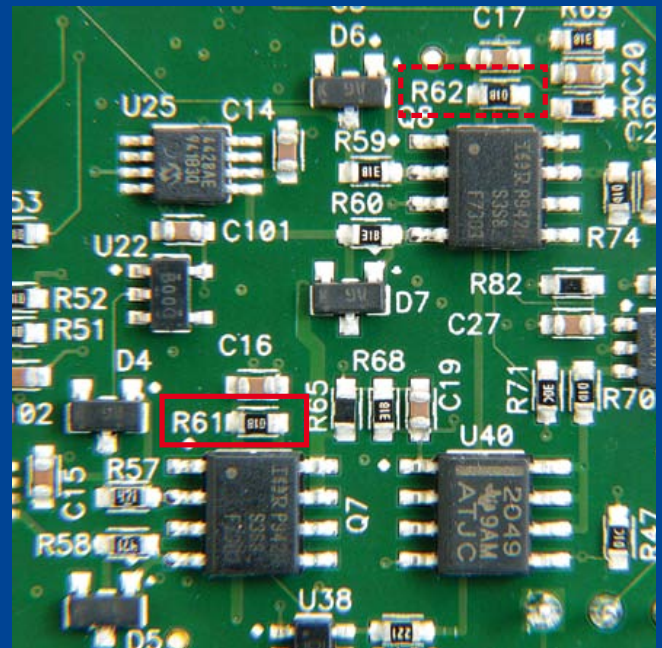
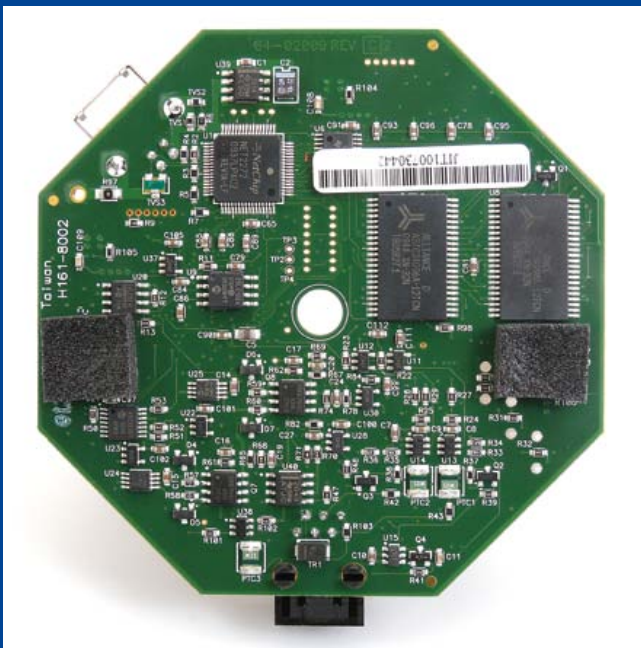
By Luc Lemmens (Elektor Labs)

During the development of the USB-stick logger published in the September 2011 edition of Elektor we couldn't manage to get the communications going to the microcontroller via Microchip's In Circuit Debugger 3 (ICD 3). We did manage to program the flash memory of the chip in the circuit, a PIC24FJ64, using the predecessor of the debugger, the ICD 2. However, as soon as the logger was connected to the newer version the development environment of Microchip stated that the target processor didn't correspond with the type that was configured. It has to be said that an ID of '0000' (the value returned in MPLAB) gave rise to the suspicion that there were communication problems.

A quick search on Microchip's website told us that there may be problems with the ICD 3 when the pull-up resistor connected to MCLR of the system is too strong. Our prototype indeed

to the bottom of the case with sticky feet; in our ICD 3 we could pry them off easily, without causing any damage. Resistor R61 (1 k Ω) should be replaced with one of 100 Ω . This is easier said than done, since we have to deal with SMDs and the resistor has an outline of 0402, which is very small. Despite that, we don't really need specialist tools for this single component. With a bit of courage and care it's possible to do this with an 'ordinary' soldering iron.

Desoldering the resistor turns out to be the more difficult task. Place the tip of the soldering iron on top of the resistor, preferably such that both metal ends of the component are heated at the same time. Tinning the tip of the soldering iron beforehand will improve the thermal contact with the resistor. Eventually it will come loose and it should stick to the tip of the soldering iron.



did have a value of just 1 k Ω , whereas Microchip warned that problems could occur with a pull-up resistor of 4.7 k Ω . The simplest solution to this problem was to increase the value of this resistor to 5.6 k Ω .

In the associated Engineering Technical Note (ETN#29, [1]) it's mentioned that a resistor can be changed in the hardware of the ICD 3, which will prevent this problem from occurring in the future. A quick description of the procedure follows (also see the photo).

Opening the ICD is very straightforward. The circuit board can then be removed from the case to replace the resistor, which is mounted on the underside of the board. The board itself is fixed

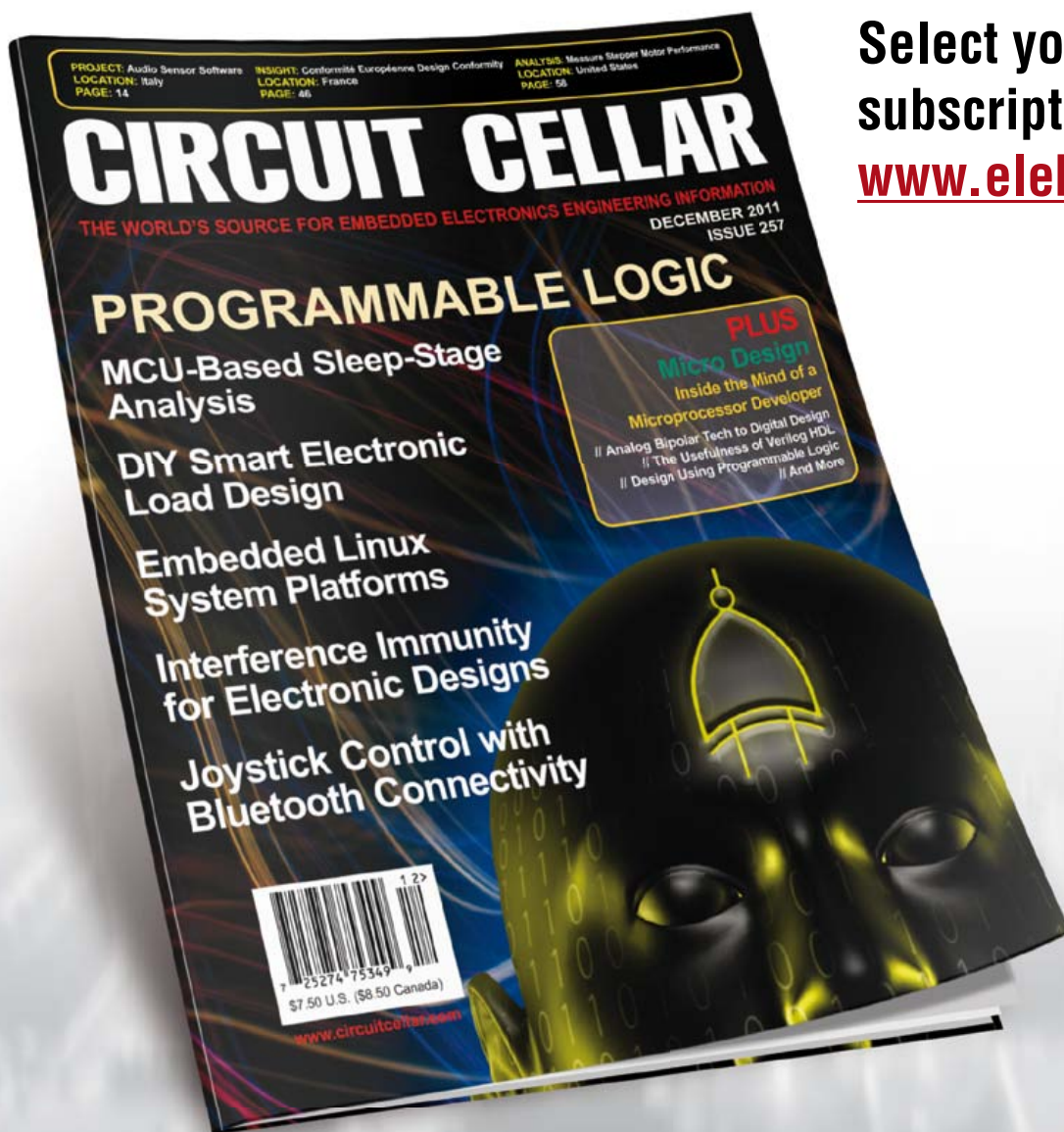
A small pair of tweezers will come in very useful to keep the new resistor in place when soldering it to the board. Once the resistor is positioned properly it doesn't take much effort to solder both ends to the board. When the operation has been carried out successfully you should do the same for R62, since the V_{DD} connection of the ICD 3 can suffer from the same (potential) problem.

(120023)

Internet Link

[1] http://ww1.microchip.com/downloads/en/DeviceDoc/ETN29_MPLAB_ICD%20%20VPP_CURRENT_SINK.pdf

Subscribe now to the leading US-based computer applications magazine specializing in embedded systems and design!



Select your personal subscription at www.elektor.com/cc-subs

12 editions per year for just

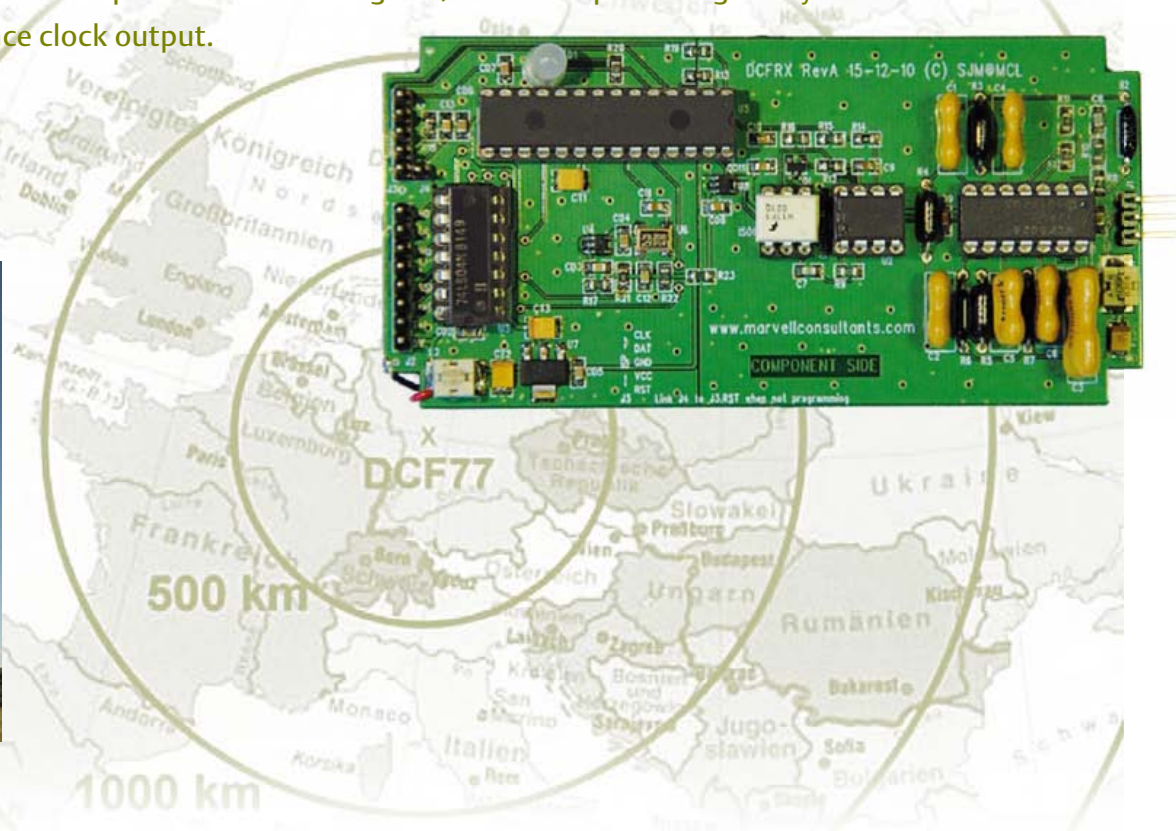
Digital: \$38
Print: \$63
Digital + Print: \$90

**CIRCUIT
CELLAR®**
THE MAGAZINE FOR COMPUTER APPLICATIONS

Ultra-accurate DSP-based DCF77 Timecode Receiver

Beats commercially available demodulators hands down

To extract the highest possible accuracy from the German DCF 77.5 kHz timecode broadcast this project uses DSP algorithms running on a low-cost dsPIC33 microcontroller to filter and demodulate both the AM and phase modulated signals, while also producing a very stable 10 Hz carrier-locked reference clock output.



By Steve Marchant (United Kingdom)

Whereas commercially available receivers / demodulators (e.g. from Galleon Systems or Conrad Electronics) work well and produce reliable time-code pulses, the exact timing of these pulses (with respect to a stable 1 Hz clock) have large amounts of jitter — of the order of tens of milliseconds. The reason for this is that they all rely on crystal filters to extract the carrier frequency — the ultra-low bandwidth of these filters seems ideal for the job but brings with it inherent timing issues for the demodulation stage. These low-cost commercial receivers also do not provide a carrier-locked reference frequency output nor do they decode DCF's pseudo-random phase

modulation scheme, which can provide an order of magnitude improvement in timing accuracy and operate much more robustly on low signal strengths. With good quality, low-jitter second markers and a carrier-locked reference frequency it is possible to build a clock with sub-millisecond accuracy.

Hardware

The system consists of a receiver board and an active antenna board, which can be remotely located at the end of a length of coax cable — proper siting of the antenna well away from interference sources can greatly improve signal quality. The antenna is a commercially available ferrite coil and tuning capacitor assembly pre-tuned to

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.

77.5 kHz. This is combined with an op-amp gain stage shown in **Figure 1** to form the line-powered active antenna capable of driving a reasonable length of coax cable to the receiver board. Looking now at the receiver/processor schematic in **Figure 2**, the first op-amp stage provides some more AC gain which drives into three further stages comprising a 6th order low-pass anti-aliasing filter with 1 dB flat pass-band to 78 KHz and -50dB stop-band from 232.5 kHz. Note that the signal is subsequently sampled at 310 kHz ($f_s = 4f_c$) for input to a DSP notch filter at 77.5 kHz so the first poten-

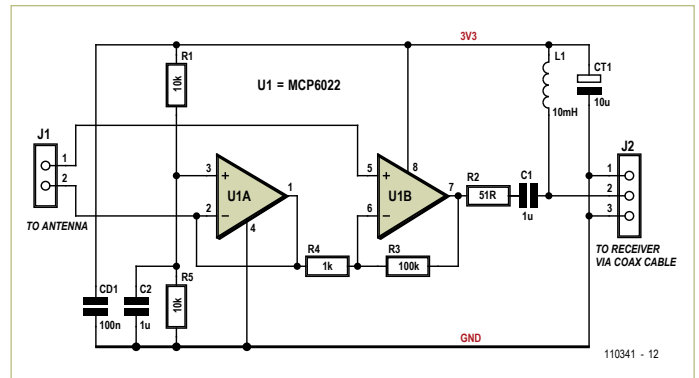


Figure 1. DCF77 antenna preamplifier circuit. The 3.3 V supply voltage arrives via the download coax cable.

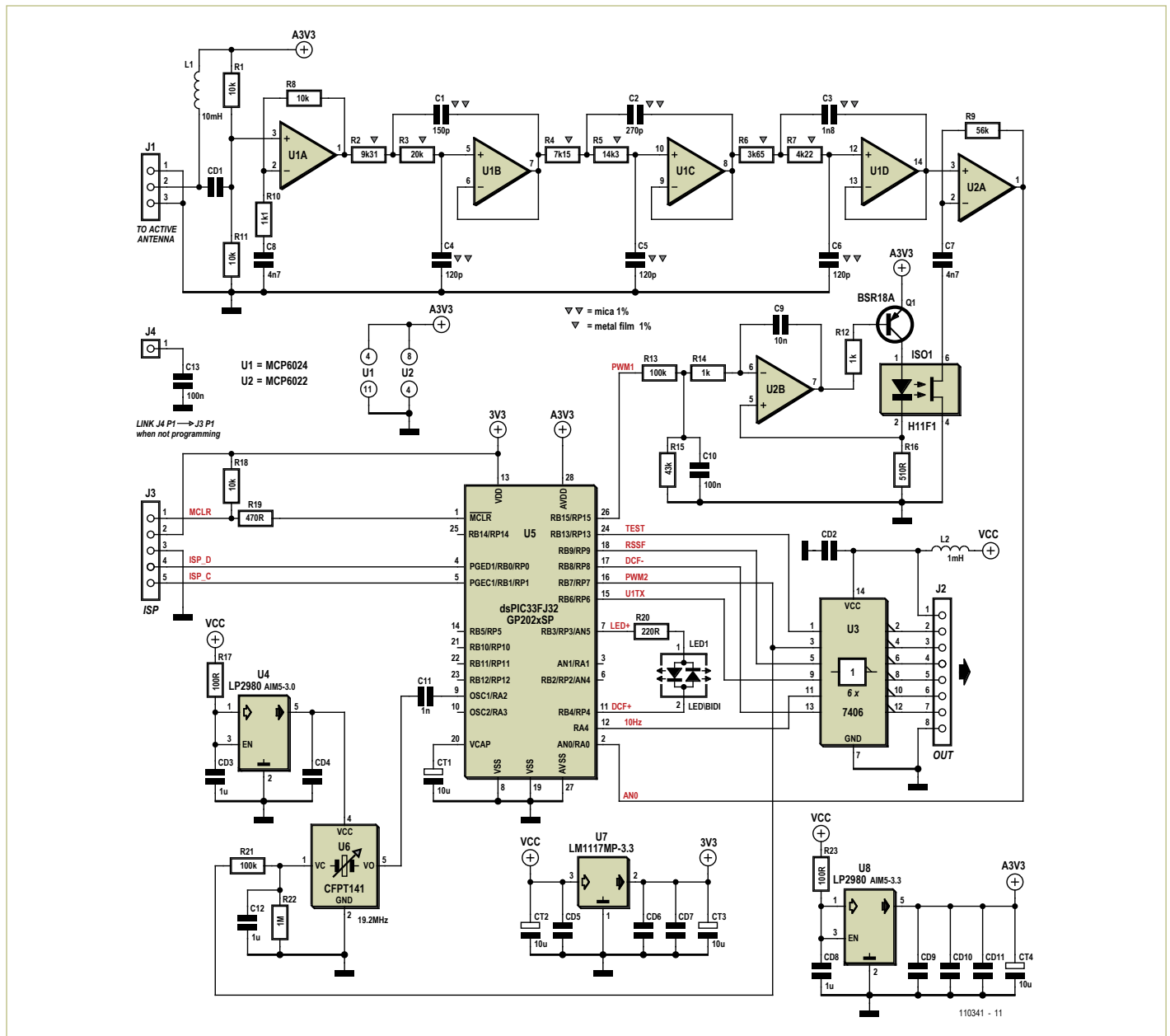


Figure 2. The circuit diagram of the DCF signal processor is a mix of analogue electronics interfaced to a microcontroller.

READERS' PROJECTS

I/O Connectors	
J1: Antenna header (3-pin)	
J1.1	GND
J1.2	RF signal
J1.3	GND
J2: Power and output signals header (8-pin)	
J2.1	5 V in, approx. 100 mA
J2.2	TEST output toggles whenever the main loop exceeds its allocated time
J2.3	The VCXO PWM signal to the master oscillator
J2.4	n/u
J2.5	9600 baud serial out (TTL polarity, not RS232 polarity)
J2.6	10 Hz reference output
J2.7	DCF pulse output, clean & accurate leading negative edge
J2.8	GND
<i>Note all outputs are open collector if a 74LS06 buffer is used at U3.</i>	
<i>Replace U3 with a 74HCT04 if totem pole outputs are preferred.</i>	
J3: PIC in-system programming header (5-pin)	
J3.1	MCLR
J3.2	3v3
J3.3	GND
J3.4	ISP Data / GPS reference pulse input
J3.5	ISP Clock

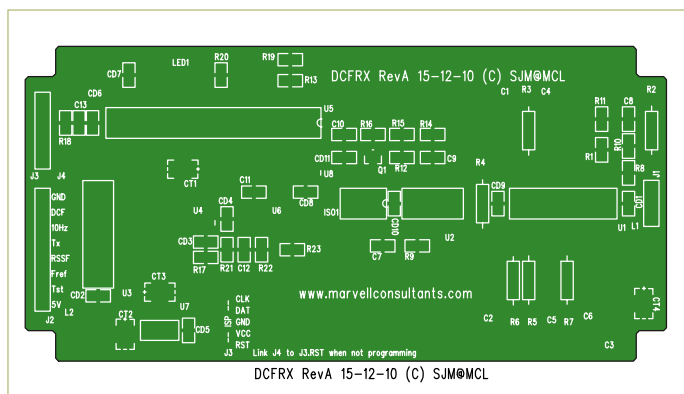


Figure 3. Silk screen overlay (80% of real size) showing component placement on the receiver/processor board. The complete PCB design files are available free from [1].

tial alias we need to worry about is at 232.5 kHz. The analogue filter was designed using Microchip's free FilterLab software and should be constructed with 1% components throughout as indicated. A fifth op-amp stage provides additional programmable AC voltage gain. All the op-amp stages have unity gain at DC and so the mid-rail bias voltage applied at the first stage simply propagates through to the last. The output of the fifth stage feeds directly into an ADC pin on the dsPIC33 where the signal is sampled at a rate of 310 kS/s. Note that the gain of the last stage of the circuit is optimised by software control to provide a roughly 3 V_{pp} signal for the ADC. After demodulation by software (see below) the DCF time-code signal is buffered by an inverting, open-collector output driver. The pro-

grammable gain stage makes use of an H11F1 opto-coupled bi-lateral FET — effectively a current-controlled isolated variable resistor — in the feedback path of an op-amp gain stage. The higher the current through the LED, the lower the resistance of the FET and the higher the gain of the op-amp stage. The H11F1's LED is driven by a voltage to current converter (implemented with a spare op-amp and a PNP transistor) driven by a filtered PWM output from the PIC, enabling software control of the overall RF gain. To provide the carrier frequency-locked clock source the master crystal oscillator circuit of the PIC is implemented with a voltage-controlled crystal oscillator to allow its precise tuning under software control. This is done using another RC filtered PWM output from the PIC.

Software

The dsPIC33 provides a 40 MIPS 16-bit DSP-capable core with RAM, Flash and a host of peripherals, most relevant of which is the 500 KS/s 12-bit ADC used to sample our RF signal at 310 kHz. The brilliantly designed ADC sub-system has a built-in buffer that can hold two pages of up to eight samples and flips from one to the other automatically setting a flag to tell the software when new samples are ready. The software processes four samples at a time in an endless loop that must complete each pass in less than 13 μs — the cycle time of the carrier frequency. In fact 40 MIPS is not adequate for this application and we have to over-clock the PIC to obtain enough CPU power.

After obtaining our four samples (one cycle's worth) the first job is to cross-correlate (multiply) them with a sine and a cosine waveform of the frequency we want to extract, f_c . In the digital world that simply consists of {0, 1, 0, -1} and {1, 0, -1, 0} respectively, the multiplications are trivial. Each of the four sine results are added together, as are each of the four cosine results, these totals then update two 120-entry ring buffers and a running total of each buffer is maintained. These running totals can be vector-added using

$$\sqrt{(\sin^2 + \cos^2)}$$

to produce an amplitude result. The amplitude signal is filtered and buffered over a two second period to obtain maximum and minimum values from which are computed an upper and a lower threshold with which to demodulate the amplitude signal into binary. This binary signal is of course the raw time code signal and could be used as input to a suitable clock — but we can do much better, more of which later.

In order to get this far though there are a few more things to consider, firstly the matter of the AGC. The software checks the four raw ADC samples to see if any are approaching either the upper or lower limit of the ADC's input range, a simple control loop strives to keep just a handful of samples at the outer limits of the ADC's range. If none or too few samples are pushing the limits the gain control is increased. If too many, the gain is decreased. The RF gain is programmable in hardware and is controlled by a PWM output from the PIC, the software simply changes the PWM register to obtain a corresponding change in RF gain.

Debugging

The receiver was developed in West Yorkshire UK, some 850 km from the transmitter, consequently the signal strength was fairly low and significant RF gain was required, it may be that your receiver could benefit from some reduction of gain for use very much closer to Mainflingen, to this end reduce R8 or increase R10. Antenna orientation and location is also fairly critical in low signal strength areas, the ferrite rod should be broad-side on towards the direction of Mainflingen (50N, 9E) and away from low frequency interference sources such as computer monitors, switch-mode power supplies etc. Cheap LV lighting transformers are a particular problem and, rather inconveniently, having an ICD2 debugger connected also

seriously compromises received signal quality. The serial debug output will not start up until the phase decoder is in lock, from power-up this can only happen once the master clock PLL locks and the LED has changed from red to green (this takes at least 10 seconds) then a clean AM signal is briefly required. The LED will clearly show if the AM signal is noisy. Once the serial data has started up there's plenty of debug info available, signal strength should be >40 or so for reliable reception the AGC PWM value should not be stuck at either the zero or 1023 extreme, the master clock PLL PWM signal should be relatively stable, not wildly fluctuating and not stuck at an extreme.

Next is the matter of fine tuning the master oscillator, this is required in order to produce the phase-locked reference output and to make decoding the phase modulated signal much easier. Given the sine and cosine cross correlation data it's a simple matter to calculate the relative phase of the signal with respect to the sample rate; once you have a measure of the phase you can construct a phase-locked loop. Either the sin or the cos data, divided by the amplitude is a measure of phase; this is used both directly and via a software filter to control another PWM output from the PIC. After a little RC filtering in hardware this is fed to the voltage control input of the crystal oscillator, closing the control loop. With the phase-locked loop in operation the master oscillator becomes locked to a multiple of the carrier frequency and from this can be derived a 10 Hz reference output. The software must also detect the state of lock — should the loop go out of lock the 10 Hz signal is squelched until lock has once more been achieved.

Finally we are in a position to consider decoding DCF's pseudo-random sequence phase modulation. This is a sequence of 512 bits, each bit lasting 120 carrier cycles, used to modulate the phase of the carrier by ± 13 degrees. The modulation starts 200 ms after the leading edge of the AM time code, i.e. the second marker, and continues for approximately 793 ms. There are as many zeros as there are ones in the sequence and so the overall phase of the carrier is unaffected. The software already has a measure of carrier phase (stabilised by the action of the PLL) and you now see why a buffer length of 120 cycles was chosen above. The code implements another cross-correlator, a reference bit sequence (stored in code memory) is multiplied by the carrier phase measurement every 120 cycles and summed over 512 bits. The result is a measure of the correlation between reference and transmitted bit sequences and depends strongly on how well aligned the two sequences are.

If we can get the reference sequences optimally aligned then we have a strong handle on the second marker timing. In fact DCF encodes one bit of time-code data on the bit sequence using either the true or complemented sense of the sequence accordingly. Consequently we get either a positive or negative correlation result depending on the data bit encoded. But how do we achieve good alignment? In a perfect world, with unlimited CPU power, we'd store all the phase data we get over the one second period (77,500 samples) and hunt for the peak correlation by re-sampling against our reference at different starting points within the data. In the real world we have only enough CPU power to do one set of correlations per second, so we have to pick a starting point and run with

it. We need some way of knowing at the end whether we started too early or too late so that we can make an adjustment ready for the next second.

To obtain this information we use yet another cross correlator running half a bit-time out of phase with the first and using a differential bit sequence derived from the main sequence. The result is a signal that (when corrected for encoded data polarity) is zero when the timing is perfectly aligned, negative when the starting point was too early and positive when the starting point was too late. We use this to constantly adjust the next starting point and to home in on the correct alignment, once good alignment is achieved we can start outputting second marker pulses with very much more accurate leading edges. One problem is choosing an initial starting point for the cross correlator, the correlation results explained above only work as long as we are within ± 1 bit time i.e. 120 carrier cycles or 1.5 ms of the optimum. To obtain our initial starting point we have to rely on the AM signal — the software looks for a clean negative edge and uses that as a reference point, under good signal conditions that's usually ok but if the phase decoder fails to lock it has to go back for another try. However once the phase decoder is in lock it's very robust and can survive periods of signal strength so low that the AM decoder fails to produce useful data.

The source and object code files for the project are available free of charge from [1].

Serial data output

The software makes the phase encoded time-code data bit, plus a collection of other debug data, available via a 9600 baud serial output — a packet of data is transmitted every second on the second with the first start-bit of the first character accurately aligned to the second marker. Note that serial output is only active when the phase decoder is running. The packet is of one of two formats:

```
"!LDsseeeeddddpppaaayy" or
":LDsseeeeddddpppaaayyoocrrrrrrzzzz"
```

where

!= the phase decoder is transitioning between locked and unlocked.

: = the phase decoder is stable.

L = locked status of master clock PLL. 0 or 1.

D = phase decoder output i.e. time code bit from the previous second.

ss = signal strength; <40 is poor, >100 is very good, max 160.

eeee = main pseudo-random phase correlator output.*

dddd = differential pseudo-random phase correlator output.*

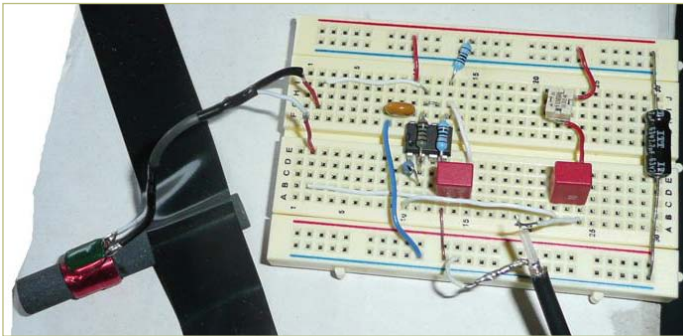


Figure 4. The active antenna was built in experimental fashion on a breadboard.

ppp = filtered PWM value to VCXO, the PLL control voltage 0 – 1023.
 aaa = current AGC PWM value, the RF gain control voltage 0 – 1023.
 yy = signal amplitude at the end of phase modulation. 0 – 160.
 oo = phase correlator happiness factor; –1 fail, 0 poor, 60 max. *
 cc = last adjustment made to the phase correlator starting point. *
 rrrrr = absolute phase of GPS reference input if present. 0 – 77499.
 zzzzz = absolute phase of second marker. 0 – 77499.

All fields are expressed in hexadecimal notation. * means that the value can be negative, if its MSB is set then compliment the data, add one and interpret it as a negative number.

The DCF output from the board is just the AM receiver pulse but with a phase decoder-disciplined leading edge. The clean leading edge lasts for 50 ms, after that the AM pulse shape takes over until 200 ms have elapsed, the section of pulse from 50 ms to 200 ms may therefore contain noise under low signal conditions. Also, in the case of second marker 59, which is omitted by DCF in AM, the 50 ms pulse is still output. If the pseudo-random phase decoder is not locked then no pulses are output. Note that the phase encoded data for seconds 59 to 9 are currently transmitted as all ones (this is not the case for the corresponding AM data bits) this can be used as a framing sequence.

The LED (if it's in the right way round) shows the raw AM pulse in green when the master clock PLL is locked or red at start-up and when the PLL is not locked.

Construction

The receiver/processor board is double-sided and contains a mix of through-hole and SMD parts. The component placement is shown in **Figure 3**; the full PCB artwork (top side and underside copper track layouts) as designed by the author is a free download from [1]. The antenna preamplifier was built experimentally like in **Figure 4**. No PCB was designed for it; if you can help other readers, let them know through the Elektor forum.

Signal quality

The DCF77 radio signal from Mainflingen has two routes to your receiver – the ground wave should provide useable signal levels up to 1000 km (approx. 600 miles) or so from Mainflingen. Then there's the 'sky wave', or propagated wave, which is a reflection of the transmitted signal via the ionosphere. This signal component is dependent on the state of the ionosphere at any given time and it varies hugely with both time of day and time of year. Usually it's stronger at night and in the winter months. Unfortunately, since it has travelled a significantly longer distance to get to your receiver, its relative phase will vary with respect to that of the ground wave, this causes all kinds of drop-outs and signal phase-shifts. Thus with increasing distance from Mainflingen it becomes increasingly difficult to obtain reliably accurate timing from the signal during the night. During the day however, 10 am to 2 pm in winter for example the signal is good for reproducibility in the region of $\pm 250 \mu\text{s}$; a figure supported by monitoring over a period of many months, see below.

Accuracy

To check the prototype receiver's accuracy, provision was made to input a GPS timing pulse (positive going 3.3 V pulse on J3.4) and to report the phase of this pulse and that of the decoded second marker as part of the debug serial output stream. In my case for example, on a good day, I see an absolute GPS phase of typically 32547 and a decoded second marker phase of 32768; see the oscilloscope screendump in **Figure 5**. The units are in carrier cycles so that's a 221 cycle offset, corresponding neatly to the propagation delay due to the receiver's 858 km distance from Mainflingen which works out at 2.86 ms or 221.8 cycles. Long-term GPS-referenced monitoring results from a prototype receiver are available on-line, see **Resources** below.

For optimum absolute accuracy, a calibration is required, in either the clock or the receiver, to compensate for the ground wave propa-

Further software development

Those wishing to dive in at the source code level or make a propagation delay adjustment are welcome to download the code and experiment with it as they wish but please make bug-fixes and improvements available to all. You'll need Microchip's MPLab IDE development software and C30 compiler, all free to download from the C30 web page.

To program or re-program the dsPIC33 you'll need a suitable in-circuit programmer, the cheapest unit supported by the MPLab environment is the PICKit3, I used an ICD2. You'll have to make up a lead to use the board's Molex 5-pin programming port J3. When a programmer/debugger is connected, expect to see severe loss of signal quality!

gation delay. You'll find a link below to daftlogic.com — a site that'll allow you to work out your distance from the Mainflingen transmitter from which you can then calculate your propagation delay using the speed of light expressed as 300 km/ms or 3.33 $\mu\text{s}/\text{km}$ or 0.2585 carrier cycles/km. The result in carrier cycles, rounded to the nearest integer, if less than 378 can be plugged into the source code — see the 'Propagation delay adjustment' section in the code. Note that a value greater than 377 cannot be used in this way as it will partly or completely stop the generation of seconds marker pulses. Once re-compiled the new firmware will output seconds markers tuned exactly for your location and absolute accuracies in the region of $\pm 250 \mu\text{s}$ will be available during the day — see **Figure 6**. The standard issue firmware contains no propagation delay correction.

(110341)

Author Contact

Steve Marchant, steve@marvellconsultants.com

Internet Link

[1] www.elektor.com/110341

Resources

1. Author's project website:
www.marvellconsultants.com/DCF
2. The software design for the cross-correlating digital notch filter was inspired by THE SCIENTIST & ENGINEER'S GUIDE TO DIGITAL SIGNAL PROCESSING by Steven W. Smith available from analog.com: <http://bit.ly/faQrb2>.
The original is at: www.analog.com/en/embedded-processing-dsp/adsp-21xx/processors/design-handbooks/scientist_engineers_guide/resources/fca.html
3. Developing in C30 for the dsPIC processor range:
www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=81
4. In-system programming PIC microcontrollers:
www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2519¶m=en534451&page=wwwdevMPLABEmulatorDebugger
5. Microchip's filterlab software:
www.microchip.com/filterlab
6. DCF77 specifics:
www.ptb.de/en/org/4/44/442/pcf77_1_e.htm
http://hw-server.com/design/pcf8574/komponenta_pcf8574.html
<http://electronic-engineering.ch/microchip/datasheets/pcf77/pcf77.html>
7. Googlemaps distance calculator:
www.daftlogic.com/projects-google-maps-distance-calculator.htm

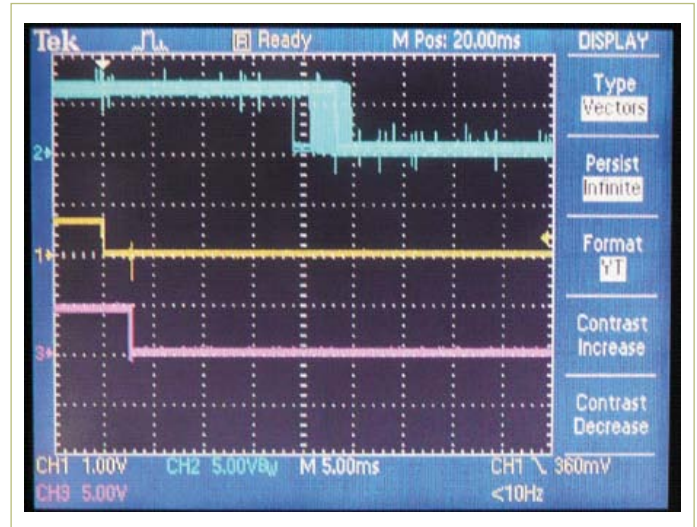


Figure 5. The 'scope will tell the difference! Top trace (blue) = Galleon MSF Rxvr; Middle trace (yellow) = GPS reference; Bottom trace (pink) = DCF Rx. Two hour, 7200 s 'exposure' 14:30 – 16:30 on 24 May 2011. At HX7 UK: DCF propagation time: 2.86 ms; MSF propagation time: 0.3 ms.

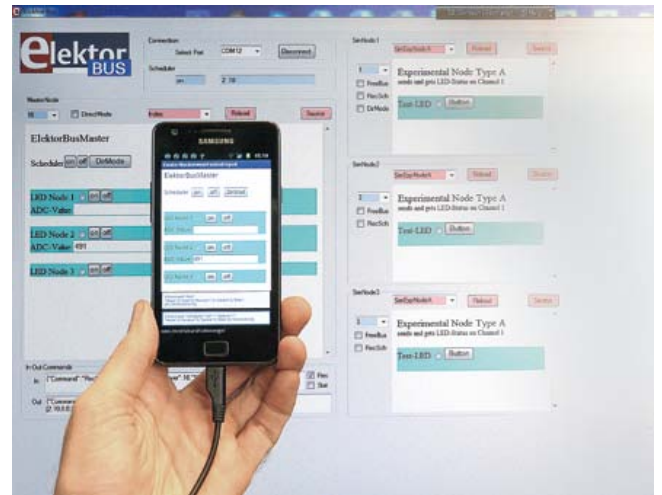


Figure 6. DCF77 receiver typical short-term (24 hour) performance. Red dots show average timing accuracy over a 200 s period with grey lines indicating stanrad deviation over that period. Vertical axis is 100 μs per division. Ideal performance would be a flat line across the centre. The yellow line indicates signal strength. Notice the contrast between day-time (10 am - 2 pm) and night-time performance.

Here comes the Bus! (11)

Simulation beats cogitation

While developing ElektorBus applications it started to get a bit tedious to have to set up all the hardware every time. This new PC-based software simulates up to three freely-programmable bus nodes, which can be even used in conjunction with real-world bus nodes and with another master, perhaps running on a smartphone. This allows development and testing with or without actual bus hardware, and allows much greater insight into the operation of the bus.



By Jens Nickel (*Elektor* Germany editorial)

After taking a slight diversion in the last instalment in this series, our bus is now back on track. New hardware, C code and applications are being developed, and so my *Elektor* labs colleague Raymond Vermeulen was charged with designing a new compact circuit board for the project. As soon as this is ready we will be able to take the bus design from prototype to final form, covering greater distances (up to tens of metres) within the small castle that forms the hub of the *Elektor* publishing operation.

In the course of developing the bus a certain amount of hardware has accumulated on my desk. When embarking on a new application everything has to be gathered together, connected up and reprogrammed: this unwieldy collection of equipment hardly makes for the most flexible or agile development.

However, since the bus communications protocols are all implemented in software, it seemed to me that it should easily be possible to simulate the hardware. I would then

be able to develop and test applications on any computer, with the help of the HTML-and-Javascript control interface that has been previously described.

Easy to program

After a fair quantity of midnight oil had been burned here in the *Elektor* castle, a considerably extended version of the ElektorBus browser was produced: as usual, this can be downloaded from the project web pages as VB.NET source code and as a .EXE file. The structure of the software is shown in **Figure 1**. As before a 'host', or master, controls the bus. This can be programmed, even by beginners, using a little HTML and Javascript. It is written in the form of one or more .htm files, as we have briefly set out previously [1]. For example, when the user presses a button in the HTML interface, the included JSBus Javascript library assembles a small message element, called a 'part', encodes this into a message and sends the message 'out' to the host, which ensures that it ends up being transmitted on the bus. It works in the same way in the opposite direction: a received message is passed

from the host to the Javascript library, which decodes the message into its constituent parts, which can, for example, be readings or threshold alarms. The application developer need only ensure that the readings are suitably displayed in the HTML user interface, which requires just a couple of lines of Javascript. The protocol stack is shown again for reference in **Figure 2**.

Since we have already obliged bus application developers to become acquainted with simple HTML and Javascript, it seems sensible to use the same approach to implement the simulated nodes. For each node we just need to create a suitable HTML file which contains the user interface of the node (such as buttons and LEDs) as well as the 'intelligence' required of a bus participant. It is of course also possible for a single HTML file to simulate two or three nodes, as long as they are all to exhibit the same behaviour. Since different applications are likely to share at least some functionality, we hope that developers will be able to take advantage of the efforts of others. Here we offer a starting-point, with three different types of node (see below) available for

Elektor Products and Services

- Experimental nodes: PCB # 110258-1 or set of three boards # 110258-1C3
 - USB-to-RS-485 converter (ready built and tested): # 110258-91
 - Free software download, file #110708-11.zip
- Products and downloads available via www.elektor.com/110708

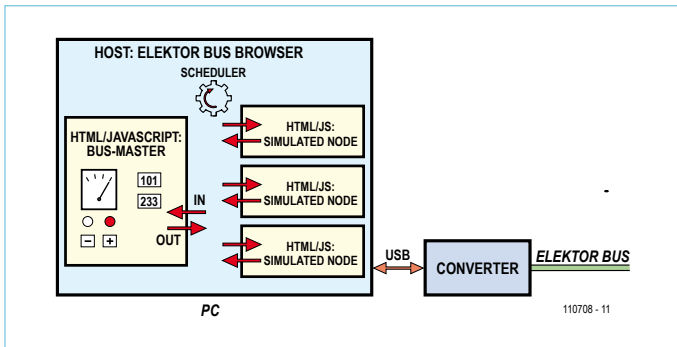
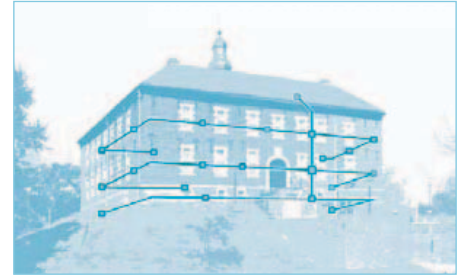


Figure 1. As usual the host program is responsible only for the basic bus functions. The ‘intelligence’ in the master and the simulated nodes is provided by a little Javascript.

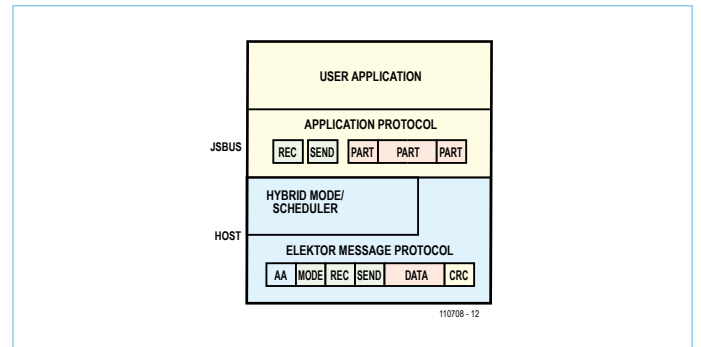


Figure 2. The protocol stack: the Javascript library divides the bus messages into parts (such as sensor readings) that can then easily be processed individually [1].

download [2].

Another advantage of using Javascript is that its syntax is broadly similar to that of C. In the future a C version of our bus library will offer similar functions to those offered by the Javascript JSBus library, and so it will be possible to port code from a virtual node to a real-life node or vice versa without too much effort.

Choose your address

The HTML files for the simulation should be placed, along with the files ‘Index.htm’ and ‘JSBus.txt’ in a folder called ‘UIBus’ on the desktop. The first .htm file in the folder which contains the characters ‘Sim’ in its filename will be used by default for all the simulated nodes (see Figure 3). Of course it only makes sense if each simulated node is given its own address: this is done using the blue combo box. In the last two instalments of this series we set the node address explicitly by hard-coding it in the Javascript, and so the file could not be used simultaneously for more than one node. To allow dynamic address selection we have extended the Javascript library to include a new variable `ownAddress`. When the combo box is used to change the address to a new value it is passed on to the Javascript

and the variable `ownAddress` is suitably modified. The variable can then be used in the node code. For example, a (type A) node would send the status of its test LED using the following code:

```
var parts = InitParts();
parts = TransmitValue(parts,
    ownAddress, 10, 1, 0,
    LedStatus);
SendParts(parts, true);
```

Just beneath the address selection combo box is the option to select for each simulated node whether it should send messages only when invited to do so by the scheduler or also during a ‘free bus phase’ (see [3]). In any case the host will only send a message from the node when there is something to send: for example, when the test LED on a type A node is toggled by pressing the HTML button. The Javascript code conveys the message out to the host, storing the message until it is that node’s turn.

Bus development environment

For an initial test, download the zip file accompanying this article from the *Elektor* website [2]. The folder ‘UIBus’ contained within it should be copied to the desktop. When the new ElektorBus browser is started it should display the file ‘Index.htm’ in the large window on the left: this contains the bus control proper. The node address is preset to 10:

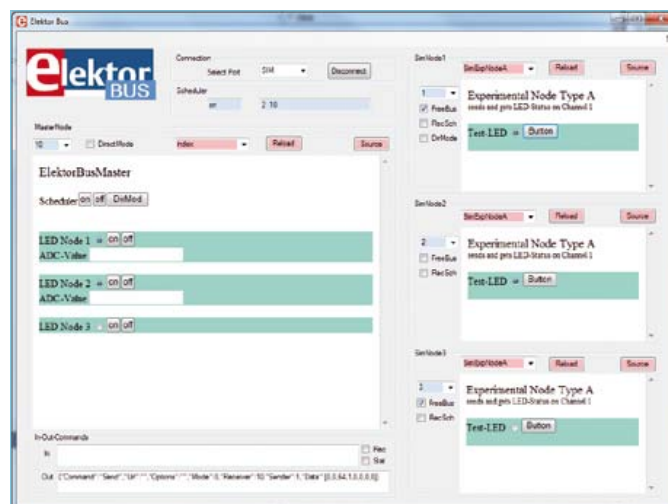


Figure 3. The new ElektorBus browser. On the left is the master user interface panel, while on the right are the simulated nodes, in this case all of type A.

see Figure 3. On the right there are three ‘type A’ simulated nodes: these are the simplest kind, consisting of an HTML button and a ‘test LED’. When the button is pressed the state of the LED is toggled and its new state is sent to the master over the

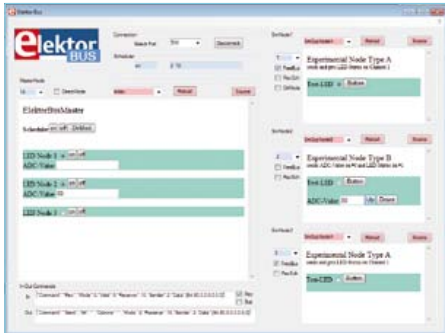


Figure 4. Node type B sends simulated ‘readings’ that can be adjusted using the up and down buttons.

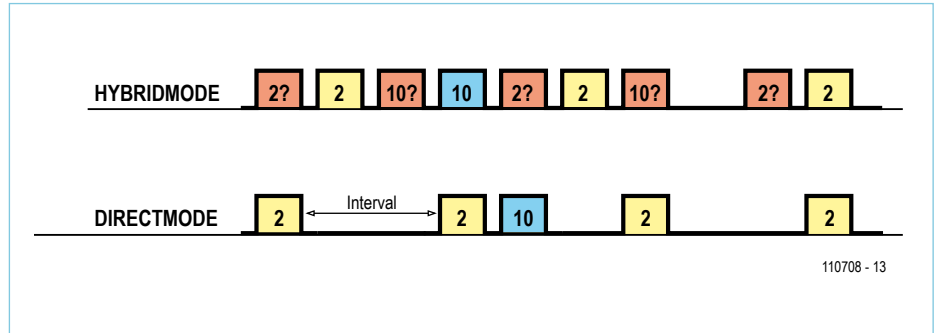


Figure 5. When there are just two bus participants, there is no need for scheduler messages (red). In ‘direct mode’ the slave (typically a sensor node) transmits at predefined intervals, with any control messages from the master following immediately afterwards.

bus on channel 1. Likewise, the node is also capable of receiving a command to switch its LED on or off.

Pressing the button labelled ‘Source’ will reveal the contents of the HTML file. A new window will open to display the code. As described in [1] it consists of a reference to the JQuery Javascript library, the Javascript application code itself and a little HTML to implement the user interface. You can edit the code in the large text box and save it using the ‘Save’ command. If you click on ‘Reload’ in the main window, the HTML page will be reloaded with your changes: this makes for a quick development cycle. Let us take a quick look at the source code for the type A node. There is a function called `ToggleLED`, which calls the function `TransmitValue`. This latter function has been reimplemented in JQuery. It is comparable in operation to `SetValue`, which will be familiar from part 9 [1], except that here the master does not send a value: instead a node sends a value to the master.

Now we need to consider whether bus hardware is connected or not. First the situation with an USB-to-RS-485 converter attached: since this device always receives all the messages it sends, the VB routine `ShowMessage` described previously will always be called each time. This routine will analyse and process the message, and if the address matches that of the master or of a simulated node the message will be passed on to the appropriate part of the Javascript code. If, on the other hand, no converter is attached,

we need to simulate the RS-485 ‘echo’ function ourselves. In the VB code the transmit routine `SendMessage` calls the routine `EchoMessage`, which in turn (after a small delay) passes the message on to `ShowMessage`. To enable this the user must have selected ‘SIM’ rather than one of the serial ports in the combo box at the top of the screen and, as usual, clicked on ‘Connect’.

LEDs and analogue values

Whether or not the converter is connected, once the scheduler has been launched it should be possible to control the test LEDs on each of the three nodes from the master. Conversely, when the buttons on the nodes are pressed, it should be observed that on the master only the LED on node 2 changes in synchrony. This is because only node 2 is ‘scheduled’. If the ‘free bus’ check box is ticked for the other two nodes then it will be possible to see in the simulation that the LEDs corresponding to these nodes update more quickly in the master user interface. The reason behind this is that (ignoring the possibility of message collisions during the free bus phase) the scheduler allows a free bus phase twice as often as it polls node 2. Now switch the middle node to type B (‘SimExpNodeB’ in the combo box: see **Figure 4**). You can use the ‘Up’ and ‘Down’ buttons to simulate a potentiometer connected to one of the analogue inputs on an experimental node. However, the simulation is not exact: the Javascript code only sends a message with the new ‘reading’

when we click on one of the HTML buttons. In real-world sensor nodes we have so far arranged for a message containing an updated value to be emitted whenever the node is polled by the scheduler, with the result that the reading shown by the master jitters a little to and fro.

So far the reaction to scheduler messages has been handled entirely by the host code, without involvement from the node code. This changes if you tick the check box ‘RecSch’ (for ‘receive scheduled messages’). Now the host will also pass scheduler messages on to the Javascript code for handling. The JQuery library has been extended so that in these cases the function `ProcessPart` (which must be present in all HTML files) in the node code will also be called. The argument passed will be `null`, a special signalling value in Javascript indicating ‘nothing’. Scheduling can now be handled in the code for a type B node (see **Listing**). To provide reassuring jitter in the display of readings, we add a little ‘noise’ to the values.

Adding real-world nodes to the mix

The master will work perfectly well in conjunction with real bus hardware such as the experimental node from part 8 of this series [4] which included a light-sensitive resistor (firmware ‘ElektorBus9.bas’). Unless the contents of the EEPROM in the microcontroller are changed, the node will have address 2. Any nodes we simulate at the same time will need to be assigned differ-

ent addresses to ensure that readings are received correctly from the sensor, because the messages from the simulated nodes will always also appear on the real bus.

The system even continues to work if we add another master to the bus. The prototype of the 'Andropod' Android interface board has been sitting on my desk for a while, and an ElektorBus browser for Android is also ready to go: we are waiting only for the printed circuit board design, which I hope to bring to our readers in the next issue. If the files 'Index.htm' and 'JSBus.txt' are copied to the smartphone and the 'ElektorBusBrowserForAndropod' is launched, the user interface presented is similar to that on the PC. The reading from the light sensor can be monitored on the small screen just as it is on a large monitor. It is important to take care that only one of the ElektorBus browsers runs the scheduler. Control messages, however, can be issued equally well from the PC as from the smartphone. In each case the master can be under the control of an internal or an external scheduler, and in principle this

scheduler could run on one of the ATmega microcontrollers. Readers who have been following this series closely will realise that the earlier versions of the PC software will not work with an external scheduler, as the master is called directly from the loop of the (internal) scheduler. In the new version of the ElektorBus browser the two functions are separated. The routine `ShowMessage`, which is called when a complete message has been received, now deals with the scheduler messages. And as mentioned above, thanks to the local echo function it does not matter whether these originate from within the same program or were sent by another transmitter.

The direct approach

The function `ShowMessage` has also been written so that it will work when there are just two bus participants operating without the assistance of a scheduler. 'Direct mode' is used when a bus participant (typically a sensor node) sends messages at predetermined time intervals. The other bus participants then use these messages as a time-

base. For example, if a controller wishes to send a message to the sensor, it can do so immediately after it sees the periodic message from that sensor: see **Figure 5**.

Another possibility that is not yet implemented is the (more conventional) reverse of the above: the master generates the timebase and the slave replies. The master can send control commands as part of this exchange with the sensor node or can ask for particular readings. The text box shows how these request commands are encoded. The thermometer application from the previous instalment of this series [5] works much better in direct mode. Suitable BAS-COM firmware (file name 'ElektorBusDirect-Mode.bas') is available at [2]. If the temperature sensor node is programmed with the corresponding hex file then it will without further ado send a new temperature value approximately every 500 ms to the master. In the other direction messages are sent to the sensor to set the measurement units and scaling, which now works much more reliably. You can try it out by loading the file 'IndexTemp' into the new ElektorBus

Listing 1: The node code reacts to a scheduler message (identified by `part==null`) by sending a message (excerpt from 'SimExpNodeB.htm').

```
function ProcessPart(part)
{
  if (part==null)
  {
    if (Noise==1) {Noise=-1} else {Noise=1};
    SensorValue = SensorValue + Noise;
    TextboxSetvalue("SValue", SensorValue);

    var parts = InitParts();
    parts = TransmitValue(parts, ownAddress, 10, 0, 0, SensorValue);
    parts = TransmitValue(parts, ownAddress, 10, 1, 0, LedStatus);
    SendParts(parts, true);
  }
  else
  {
    if (part.Channel==1)
    {
      LedStatus = part.Numvalue;
      RadioButtonSetvalue("LED", LedStatus);
    }
  }
}
```

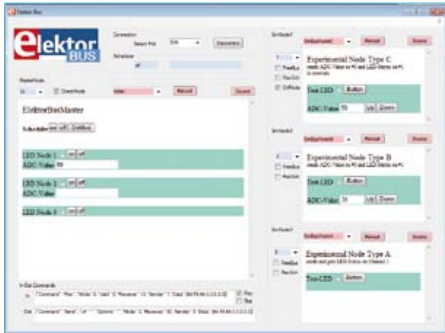



Figure 6. Direct mode in simulation. The simulated node above sends out readings periodically.

browser in the master node. Before the master can send messages in direct mode the eponymous check box must be ticked (see Figure 6). Don't forget that none of the simulated nodes should be assigned address 2! On the smartphone version there is of course somewhat less room for these control elements. For this reason we also allow the possibility of enabling direct mode from the HTML user interface: see the button labelled 'DirMod' in Figure 6.

Repeat function

It would of course be desirable to be able to simulate the whole system at once. To this end node type C is capable of sending messages at preset intervals. For a test, load the code 'SimExpNodeC' into the first simulated node and tick the check box labelled 'DirectMode' as shown in Figure 6. The line in the Javascript code responsible for periodically sending a message is this:

```
var sendinterval =
setInterval ("SendValues ()",
500);
```

The first parameter expected by the function `setInterval` is the name of another function, which is to be called regularly. The second parameter is the interval between these calls in milliseconds. The return value is a variable that uniquely identifies this repeating action: the value can be reused later to stop the repeating action by calling `clearInterval (sendinterval)`. This

is a good example of how simple it can be to write Javascript code.

Here is a small exercise for our more sophisticated readers: extend the control and sensor node code so that the interval can be switched between 250 ms and 125 ms under control of the master (see [5]). The new function in JSBus should be called `SetIntervalValue (parts, sender, receiver, channel, mode, interval, numvalue)` and the type of the part should be `PARTTYPE_INTERVAL`. A couple of additional constants such as `INTERVAL_MILLISECONDS` have been defined: see the file 'JSBus.txt'. Our solution to this exercise can be found in the files 'IndexInterval.htm' and 'SimExpNodeC2.htm'.

The ElektorBus will continue its journey, but in a different livery. In the February 2012 edition we hope to present the 'Andropod' Android interface board, which includes the ability to add an RS-485 port. This board will

let you use Android smartphones or tablets to control the bus. In the March 2012 magazine we plan to describe the RS-485 expansion itself. ElektorBus aficionados will in future be able to keep up with developments in the 'E-Labs Inside' pages in the centre of each edition.

(110708)

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

Internet Links

- [1] www.elektor.com/110517
- [2] www.elektor.com/110708
- [3] www.elektor.com/110258
- [4] www.elektor.com/110428
- [5] www.elektor.com/110610

Requesting readings

It is possible to use the application protocol to set a target value on a node from a controller. Sensor nodes can also report current readings [4]. Until now it has however not been possible to prompt a particular sensor or actuator node to send these values: the scheduler does divide up the transmit time slots, but does not carry out polling in the strict sense of the word. We also need a format to transmit absolute threshold values, as so far we have only been able to use the current reading as the setting for an upper or lower threshold.

We now fill these lacunae:

Reading request	Byte 1	Byte 2
Request reading	104 + CH	240 (F0 _{hex})
Request lower threshold	104 + CH	241 (F1 _{hex})
Request upper threshold	104 + CH	242 (F2 _{hex})

Set absolute threshold	Byte 1	Byte 2	Subsequent bytes
Set lower threshold	104 + CH	217 (D9 _{hex})	Value (2 or 4 Bytes)
Set upper threshold	104 + CH	218 (DA _{hex})	Value (2 or 4 Bytes)

Report absolute threshold value from sensor: byte 1 = 72 + CH

In the above, 'CH' represents the channel number from 0 to 7.

Elektor Print

Confident 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
cut-rate PLUS subscription!**

Subscribe now or upgrade: www.elektor.com/subs

Electronics for Starters (1)

Diodes and LEDs

Electronic devices are becoming more and more complex, with simple circuits and discrete transistors practically a thing of the past. This makes it increasingly difficult for beginners to get up to speed. In this series we therefore want to get back to basics, and in electronics the basics are analogue. However, we realise that many beginners are interested in digital technology, so a microcontroller circuit is also included in the course material.

By Burkhard Kainka (Germany)

One way to approach a basics course would be to start with basic theory and concepts, including current, voltage and power, Ohm's law, parallel and series circuits and so on — in other words everything taught in 'Electronics 101', which you actually already know or should know. But that's deadly dull, so it's better to start with real circuits in small practical examples and projects.

You may be wondering what we aim to achieve with this course. Ideally it should help to bring new readers of Elektor up to the usual Elektor level. Some of our readers who follow this course may be the sons or daughters of long-term Elektor readers, who are looking for a chance to emulate their parents. It would certainly be helpful

for experienced electronics enthusiasts and beginners to follow the course together. We have also established a forum for this course series at www.elektor.com/starters-forum. It would be nice if old hands could contribute something from their knowledge and experience in this forum.

There may also be a few Elektor readers who have already built lots of projects but never really understood exactly how they work. Of course, you can't expect miracles from a course such as this, but it should help clear away a few cobwebs.

The basics lie largely the realm of 'old-fashioned' analogue electronics, but the fact that so much of modern electronic is digital does not mean that the basics are no longer relevant. Even for people who are interested in microcontrollers, there's no getting around analogue technology. This can be demonstrated using a few simple

Semiconductors and depletion layers

The electrical conductivity of a typical semiconductor material, such as silicon, generally increases with rising temperature, but it is very low at room temperature. This is because all four outer electrons of the individual atoms are bound in the crystal lattice (**Figure 6**). However, they can be freed by the addition of a small amount of energy.

Devices made from semiconductor materials, such as transistors and diodes, are commonly called semiconductors. They are made by intentionally adding foreign atoms to a material such as silicon (which is called doping the material) to obtain a defined conductivity. Doping with a Group V substance, such as phosphorus, produces free electrons and therefore n-type conductivity (**Figure 7**). Doping with a Group III substance produces electron holes, leading to p-type conductivity. These electron holes migrate through the crystal lattice, as though they were positive charge carriers, when the holes are filled by neighbouring electrons that leave behind new holes (**Figure 8**).

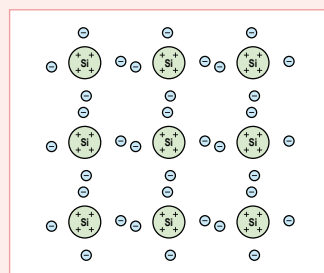


Figure 6. Crystal lattice of silicon.

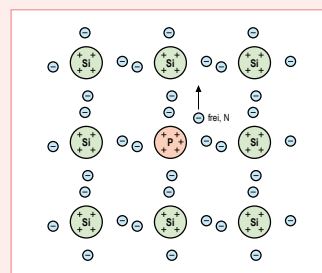


Figure 7. Silicon doped with phosphorus (n-type).

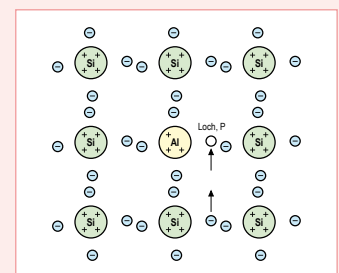


Figure 8. Silicon doped with aluminium (p-type).

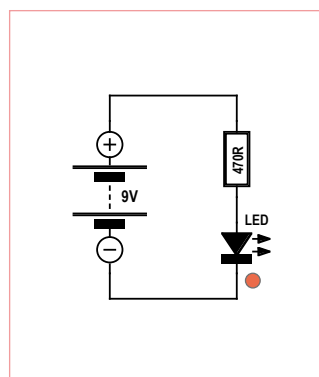
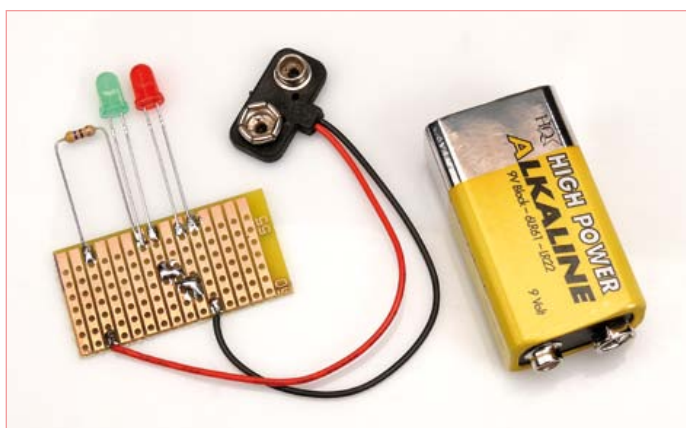


Figure 1. LED circuit.

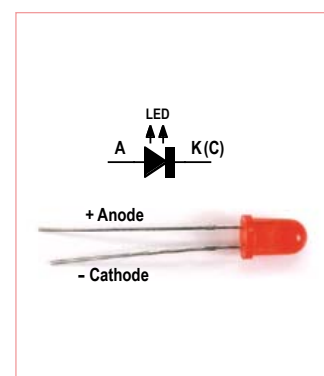


Figure 2. An LED.

examples from the world of embedded applications. For example, microcontrollers are used to measure analogue quantities, among other things. This means that the course is unquestionably suitable for people who are getting their feet wet in the pool of pint-sized computers.

LED with series resistor

Let's start off by putting together the circuit shown in **Figure 1**, with a LED, a resistor (470 Ω) and a battery. You can assemble it any way you wish — by simply soldering the components together on your desktop, using alligator clips or using a breadboard — although making a PCB specifically for this circuit would hardly be worth the effort. No matter how you do it, you're bound to get the LED to light up.

With LEDs you always have to pay attention to the polarity. The plus lead is the anode lead. The minus lead, which is normally the shorter lead, is the cathode lead. There is also a flat on the package next to the cathode lead. Inside the LED package you can see a throat-shaped mount for the LED chip, which is usually (but not always) fitted on the cathode side. The anode connection is provided by an extremely thin wire bonded to a contact on the top surface of the chip. If you connect the LED with reverse polarity, it won't light up. It shares this feature with every type of diode: current flows through a diode in only one direction.

LEDs should never be connected directly to a battery. If you look at a plot of the current through a LED versus the voltage on the LED, you can see why. **Figure 3** shows the characteristic curves of various

charge carriers toward the depletion layer. Above a voltage of approximately 0.5 V, the n and p layers start to touch each other and a current starts to flow (**Figure 10**). Good conductivity is achieved at a voltage of approximately 0.7 V. In this state the diode is operating in the forward direction.

If the polarity of the applied voltage is reversed, the opposite effect occurs. The charge carriers are attracted to the outer connections and the depletion layer widens. This makes the depletion layer an even better insulator (**Figure 11**). A typical diode such as a 1N4148 can handle reverse voltages up to 75 V. A diode effectively allows current to pass in only one direction, so it can be used as a rectifier.

In most cases the reverse voltage should not exceed the value recommended by the manufacturer. A reverse current flows if the applied voltage is too high. This results from what is called breakdown (of the insulating layer). With some special types of diodes, such as Zener diodes, this effect is used intentionally. Zener diodes have well defined breakdown voltages and are used as voltage regulators. If you abuse a silicon diode such as a 1N4148 by applying an excessive reverse voltage to it, you will cause what is called second breakdown, which is fatal. This is because the excessive reverse current heats the junction to the point of destruction. This causes the formation of a permanent, non-repairable short circuit.

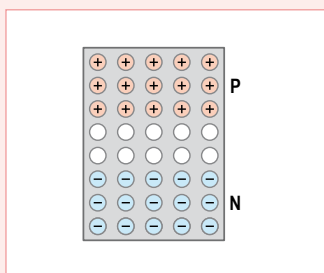


Figure 9. Layer structure of a diode.

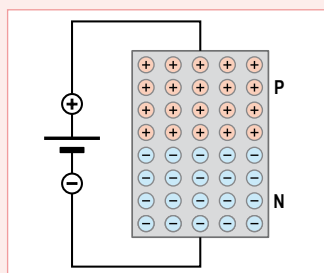


Figure 10. A diode in the forward conduction state.

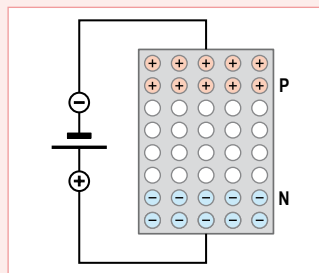


Figure 11. Depletion layer widening under reverse bias.

LEDs, as the name indicates, are diodes and also have p-n junctions. They are made from semiconductor materials such as gallium arsenide. LEDs have higher forward voltages than silicon diodes, and electron-hole recombination in LEDs produces visible light. This effect also occurs in silicon diodes, but they produce only minute amounts of light in the infrared region.

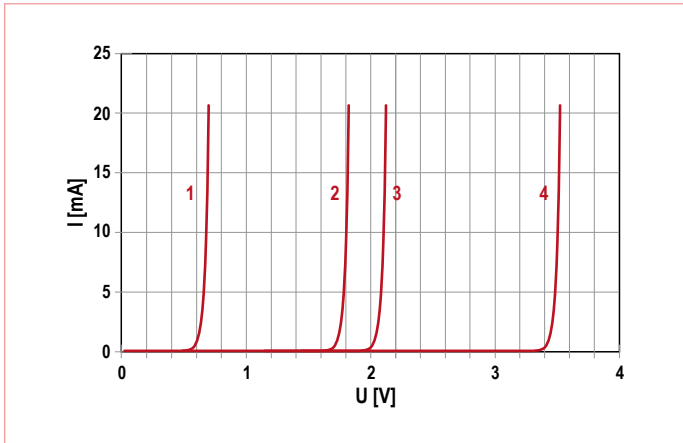


Figure 3. Characteristic curves of a silicon diode (1), a red LED (2), a green LED (3) and a white LED (4).

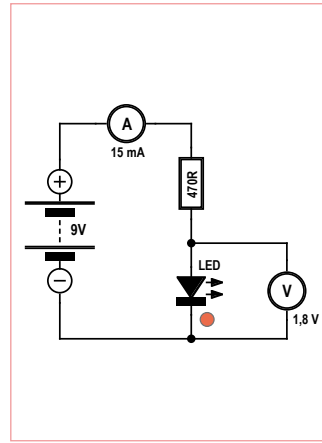


Figure 4. In-circuit measurements.

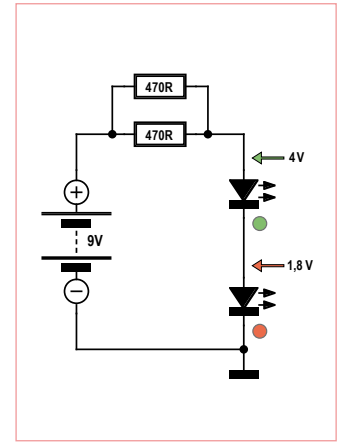


Figure 5. LEDs connected in series.

types of diodes. All of these curves have one thing in common: the current increases exponentially with increasing voltage. If the voltage is below what is called the forward voltage, virtually no current flows through the LED. However, if the voltage is just a bit higher than the forward voltage, the current quickly rises to a very high level and the LED may become overloaded. It's practically impossible to set the voltage to exactly the right value, in part because the curve shifts to the right with rising temperature at approximately 2 mV/K. Nevertheless, it's easy to set the current to a particular level, and all it takes is a single resistor. You simply have to choose the right value, and with this arrangement the right LED voltage is obtained automatically.

The forward voltages of several types of diodes at a typical current level of 20 mA are:

Silicon diode (e.g. 1N4148):	0.7 V
Red LED:	1.8 V
Green LED:	2.1 V
Blue or white LED:	3.5 V

You can check this by measuring the voltages yourself (see **Figure 4**). The exact voltages may vary somewhat. For example, modern high-brightness red LEDs have a slightly higher forward voltage than older types of red LEDs.

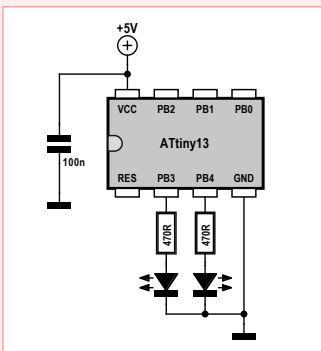


Figure 12. A microcontroller with two LEDs.

Blinking LED

LEDs are often driven by micro-controllers. Here again series resistors are necessary. The circuit depicted in **Figure 12** has two LEDs, each connected to an I/O port pin of the ATtiny13 micro-controller by a 470 Ω series resistor. The associated simple BAS-COM program sets PB3 constantly high and PB4 alternately high and low, causing the connected LED to blink. Try measuring the voltage on the PB3 lead. It will be a bit less

than 5 V (for example 4.9 V) because the switch transistor in the micro-controller also has a small resistance. You can determine the internal

resistance of the port pin from the voltage drop. The next question is how much current is flowing through the LED. You can calculate this easily. Have a look at the ATtiny13 data sheet (www.atmel.com/dyn/resources/prod_documents/doc2535.pdf) to see how much current is allowed to be drawn from an I/O port.

```
,ATTiny13 driving LEDs
$regfile = „attiny13.dat“
$crystal = 1200000
Config Portb = Output

Do
  Portb.3 = 1
  Toggle Portb.4
  Waitms 500
Loop

End
```

Component dimensioning

If you have measured the diode voltage and you know the battery voltage, you don't need to measure the current. You can simply calculate it. This is because the voltage over the resistor is the difference between the battery voltage and the LED voltage (e.g. $9\text{ V} - 1.8\text{ V} = 7.2\text{ V}$). With this information you can use Ohm's law to determine the current:

$$I = V / R$$

$$I = 7.2\text{ V} / 470\ \Omega$$

$$I = 0.0153\text{ A} = 15.3\text{ mA}$$

If you instead want to calculate the value of the series resistor, you must specify the desired current value and know the values of the supply voltage and the LED voltage. For example, suppose you want to have a current of 20 mA flow through a green LED. For practical purposes, the voltage across the LED can be taken as 2.1 V. The battery voltage is 9 V, so the resistor has to produce a voltage drop of 6.9 V ($9\text{ V} - 2.1\text{ V}$). The calculation yields a value of 345 Ω , but this is not a standard resistor value. However, you may be able to find a 330 Ω resistor or a 390 Ω resistor in your parts box. It's a good idea to choose the higher value, since this puts you on the safe side with regard to the amount of current.

$$R = V / I$$

$$R = 6.9\text{ V} / 0.02\text{ A}$$

$$R = 345\ \Omega$$

You should also experiment with this circuit with various resistors having much higher resistance values. In each case, measure the LED voltage and determine the current. Generally speaking, no matter whether you operate the LED at a current of 1 mA, 5 mA or 10 mA, the voltage across the LED is nearly the same. This is due to the exponential shape of the characteristic curve.

Series circuit

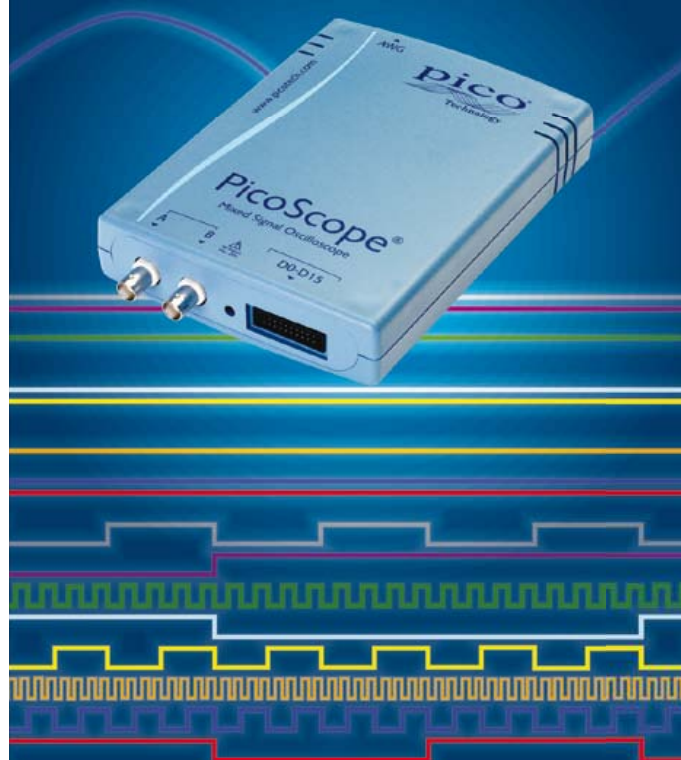
It's often useful to connect two or more LEDs in series with a common series resistor, as shown in **Figure 5**. In this situation the voltage across the series resistor is lower because the voltages across the LEDs add together. This means that the resistance must be reduced in order to obtain the rated 20 mA current through the LEDs. Suppose you are using a red LED with a forward voltage of 1.8 V and a green LED with a forward voltage of 2.2 V. This makes the voltage across the two series LEDs exactly 4 V, so the voltage across the series resistor is only 5 V. With a 470 Ω resistor you will have a current of approximately 10 mA. If you connect two such resistors in parallel, the current doubles. If you check the calculations, you should find that the current is 21 mA.

(120001-1)

THE NEW PICOSCOPE 2205 MSO

MIXED SIGNAL OSCILLOSCOPE

GREAT VALUE, PORTABLE, HIGH END
FEATURES AS STANDARD AND EASY TO USE



Think Logically...

Channels	2 Analog, 16 Digital
Resolution	8 bit
Bandwidth	Analog 25 MHz,
Digital frequency	Digital 100MHz combined
Sampling rate	200MS/s
Trigger modes	Edge, Window, Pulse width, Window pulse width, Dropout, Window dropout, Interval, Runt pulse, Digital, Logic
Price	£349

pico
Technology



www.picoscopemso.com/123

Wavelet Analysis on MikroElektronika's PIC32 platform

By Clemens Valens (Elektor France Editor)

MikroElektronika, the Serbian manufacturer of development tools for microcontrollers, offers everything a PIC32 developer needs: C, PASCAL, and BASIC compilers, debugger, programmer, prototyping boards, and even application boards with colour touch-screen display. One fine morning, I found all this lot stacked on my desk with a Post-It from Father Christmas stuck on it saying "Have fun!" Sometimes, I really like my job!

There were too many items for me to evaluate them all, so I confined myself to just the following products: the *mikroC PRO for PIC32* (v1.80) tool chain with C compiler, the link editor, the debugger/simulator and the PIC32 programmer, the *mikroMMB for PIC32* (v1.01) application board, the *mikroProg* programmer/debugger, and the *Visual TFT* (v2.01) MMI graphic design tool. I 'subcontracted' the evaluation of the *LV32MX* prototyping board to a colleague; we'll be coming back to tell you about that in another issue.

Hardware

The Microchip PIC32 at the heart of the MIPS MK4 is less well known than its counterparts with a Cortex-M3 ARM core, yet they are all playing in the rather crowded field of 32-bit microcontrollers (see also the 'Super Arduino' article in the November 2011 issue [1]). This type of processor and their application boards have become so powerful that programming them is getting closer and closer to pure computing. Soon, software development will be done directly on the boards themselves, without having to use a computer alongside. The *mikroMMB* board (\$ 99) is a little 8 × 6 cm board, one side of which is occupied by a 320 × 240 pixel TFT touch-screen display. On the other side we find a PIC32MX460F512L processor (80 MHz, 512 KB flash + 12 KB boot flash, 32 KB RAM, USB OTG device, 4 × DMA, 2 × SPI, 2

× I²C, 16 × 10-bit ADC @ 1 Msample/s etc.), a WM8731SEDS stereo audio codec, a microSD card connector, an M25P80 8 Mbit serial flash memory, a power supply, and a USB port. On the long sides of the board are rows of 26 holes giving direct access to the microcontroller ports. The *mikroMMB* is a smaller version of the *MMB for PIC32MX7 (MultiMedia Board, \$ 149)* board, which has in addition an Eth-

ernet port, a USB host port, and a number of LEDs and push-buttons. Each object has several modifiable properties that let you customize the object. Most of the objects accept events like 'click' or 'press' (a click is a short press) produced by the MMI engine that manages the touch panel. When you have finished designing the screens, press the Generate Code button to automatically produce the source code for

buttons; four panes of the same size and one button; one large pane with one button. And to make it look attractive, I added a screen with the Elektor logo, which appears only at start-up. Once you've got your ideas clear in your head, designing the screens in Visual TFT (Figure 1) is fast — the tool is quite intuitive, especially if you have a minimum of experience of Visual Basic or Visual Studio. The number of predefined controls is limited to just three types of buttons (rectangular, round, or rectangular with rounded corners), one label, and one image. There are also four graphics objects: rectangle, circle, or rectangle with rounded corners, and line. This isn't very much; if you want to create more complicated objects, you'll have to construct them from these basic building-blocks. It is possible to design in several layers, but before you start, do be aware that they don't exist in the code, they are merely used for organizing

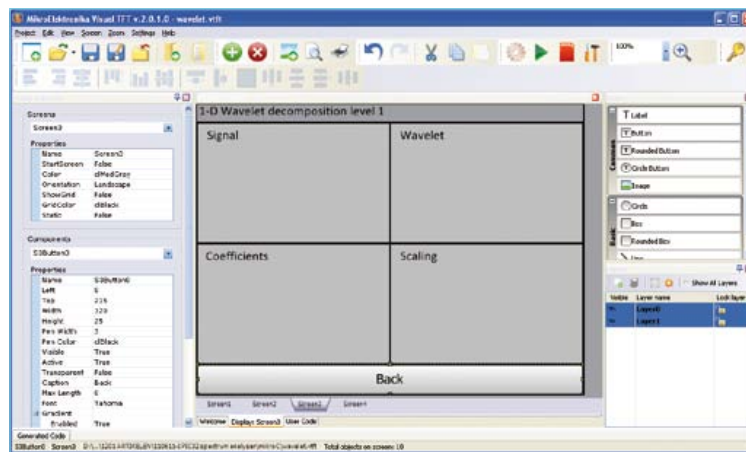
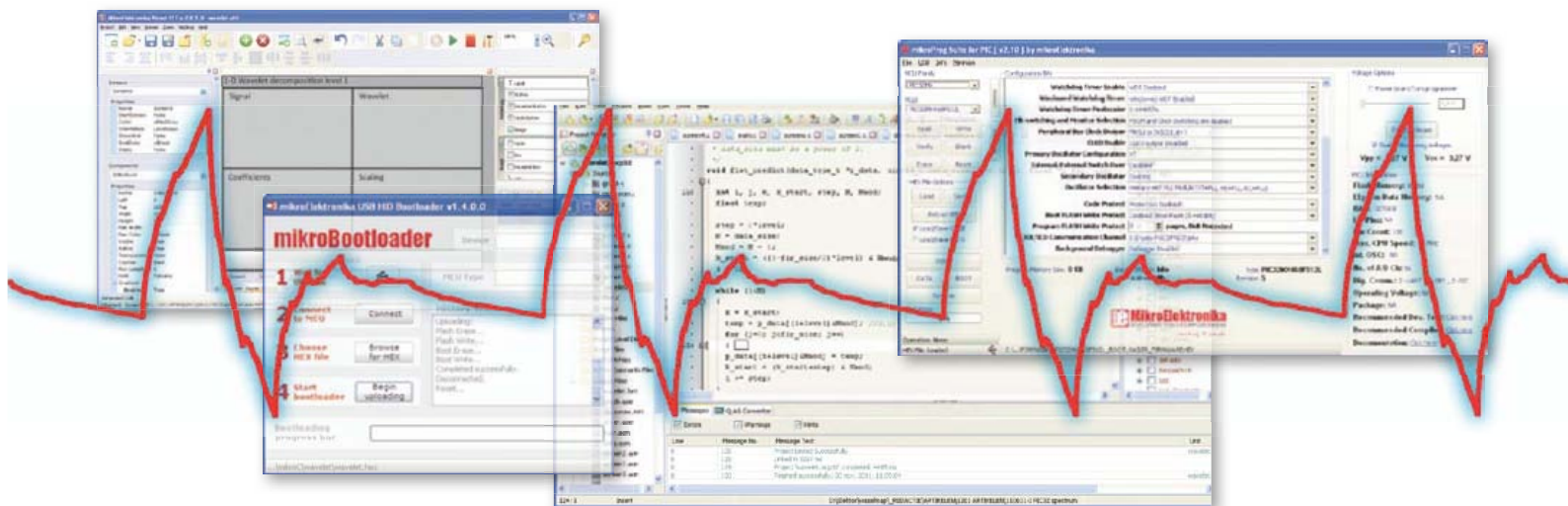


Figure 1. Visual TFT lets you produce a not-too-complicated MMI quickly. Here we see a project with four different screens.

ernet port, a USB host port, and a number of LEDs and push-buttons.

Visual TFT

I started my evaluation with the Visual TFT tool (\$ 99) in order to define the man/machine interface for my test application. After some consideration, I decided to confine myself to three screens: five panes of different sizes plus four



the MMI in C, PASCAL, or BASIC, according to your choice. It is also possible to run the compiler from within Visual TFT.

The code produced and compiled works without modification, i.e. the first screen is displayed. Of course, you can't yet navigate within your application, as the connections haven't yet been put in place, but the initialization of the hardware has been done. What's more, you have the option to include calibration of the touch panel when your software is run, which can be achieved by ticking the right box under Project Settings -> Advanced Settings -> Touch Panel -> Calibration.

I found the quality of the C code produced by Visual TFT a little disappointing. A whole host of global variables are defined (including one named 'i!'), H files are not protected against multiple inclusions, and compilation produces a large number of warnings of the type: "Implicit conversion of int to ptr" – all this is not very tidy! But the biggest drawback is that Visual TFT is unable to handle modifications made directly in the source code. The moment a program is changed outside this tool, synchronization is lost with the Visual TFT project – so you'd best not touch it at all.

Let's end this paragraph on a positive note: it is possible to separate the graphics resources from the source code so as to store them in an external memory such as an SD card. In this way, the appearance of

the MMI can be changed without touching the program.

mikroC PRO

After, or from, Visual TFT, you start the integrated development environment (IDE) for C, PASCAL, or BASIC in order to set about programming the application. For this article, I coded in C in mikroC PRO (\$ 299). The IDE is user-friendly, thoughtfully

designed, and includes much more than a compiler, a link editor, and a debugger. In point of fact, to make life easier for the firmware programmer, several tools are available, such as a terminal or a bitmap editor. The programming editor allows code folding, underlines undefined (or badly-written) variables, attempts to complete your 'phrases' automatically, and offers so-called 'active' comments. This feature is interest-

ing, as it makes it possible to provide good documentation for a project by including documents like data sheets, drawings, and photos. Clicking the active comment opens the file in question, or just hovering the mouse pointer over the link will bring up the illustration. These documents are copied into the project, so the active comments aren't affected if you move or modify the originals.

Despite this well-thought-out appearance, you sense that the IDE is not a real IDE for C. For example, a new file is called 'Unit' as in PASCAL or Delphi (the IDE is written in Delphi) and compilation errors and warnings are sometimes a little odd. However, the greatest inconvenience is the way libraries are handled. In their efforts to simplify programming, the MikroElektronika developers have gone a bit too far in doing away with the notion of standard C libraries. So writing #include <stdio.h> produces a compilation error!

Instead of including a standard library, you have to tick a box in the Library Manager (Figure 2). Simple enough, it's true – but it's also totally incompatible with all C code that's ever been written anywhere in the galaxy. No need to point out that MikroC PRO is not ANSI.

Quite apart from the "genuine C" aspects, the editor is not mature and still has a large number of bugs. The code folding doesn't

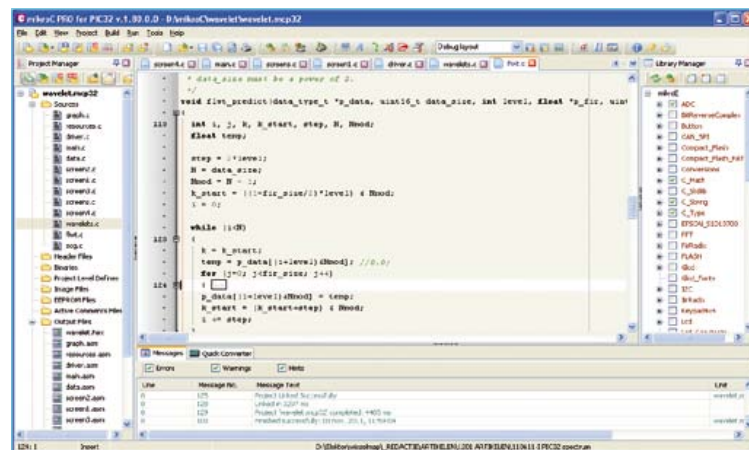


Figure 2. Here's the mikroC PRO IDE. On the right, the library manager which replaces the standard C libraries.

designed, and includes much more than a compiler, a link editor, and a debugger. In point of fact, to make life easier for the firmware programmer, several tools are available, such as a terminal or a bitmap editor. The programming editor allows code folding, underlines undefined (or badly-written) variables, attempts to complete your 'phrases' automatically, and offers so-called 'active' comments. This feature is interest-

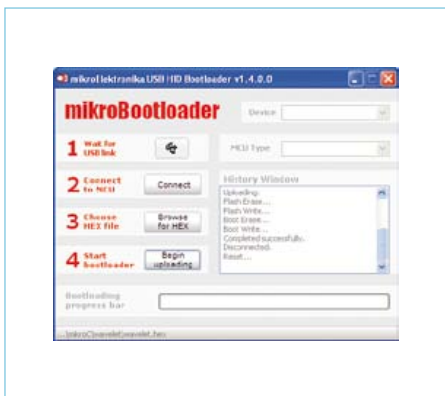


Figure 3. If there is a bootloader present in the microcontroller, the mikroBootloader makes it possible to load a new executable in record time.

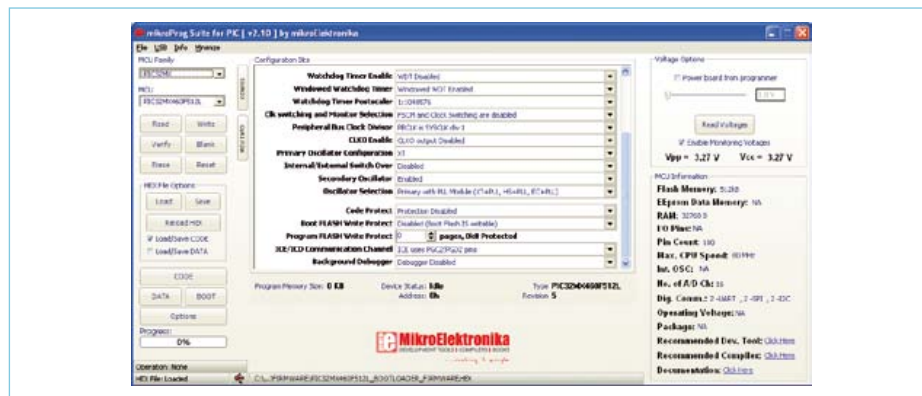


Figure 4. The mikroProg Suite utility lets you program the microcontroller. It offers many more programming options than the bootloader.

always work properly: after folding a function, double-clicking on a word (to select it, for example) places the cursor at the place where the word used to be before folding. The automatic correction can be very irritating, especially since it is not possible to cancel using 'undo'. When you're moving around within the code, the cursor does not go back to the end of the line, despite turning off a well-hidden option (Tools -> Options -> Editor Settings -> click on Advanced editor options -> click on Options tab -> maintain caret column – yes, I did find it all by myself). On my computer (Windows XP, T4200 @ 2 GHz, 4 GB RAM), scrolling using the scroll bar is jerky, especially if you try to go quickly. The automatic completion recognizes structural element references by 'dot' (e.g. element.structure), but not by arrow (element->structure). The underlining for unknown names does not recognize macros defined in another file. And there are certainly bound to be other bugs I haven't found yet.

Of course, I have reported all these problems to MikroElektronika, who promised they will be corrected in future versions. The IDE also lets you debug the code without needing additional hardware, which is very handy. In fact, it uses a simulator, but it's also possible to debug the software *in situ*, i.e. directly on the board. You select the debugger in the Project Settings window, which is not necessarily on the screen. To

display it, don't go via the Project menu – you get to it, surprisingly, via View. Select Software or mikroICD as your debugger, and Release or ICD Debug as Build Type. The debugger seems by default to stay in assembler mode. You can force it into C mode (Run menu or Alt D), but as soon as you stop the debugger in mid-flight, it goes back to assembler mode if you are unlucky enough to interrupt it within a library function. So make sure you memorize the Alt D key combination, you'll be using it a lot. The simulator is especially handy for working on an algorithm that does not use a hardware peripheral, as the simulator is more responsive than the *in situ* debugger. However, if you come across a hardware problem, you'd do better to use the *in situ* debugger so as to be sure of the state of the registers.

Programming using a bootloader

There are two methods available for loading the program into the processor's flash memory: using a bootloader, or via an external programmer. The mikroMMB board is supplied with a factory-installed bootloader, there's a little label stuck on the display to tell you. If your board doesn't have the bootloader, you can load it yourself, it's available free from the MikroElektronika website. On the computer side, there's a little free utility to run called mikroBootloader (Figure 3).

The bootloader uses a USB port and appears as an HID, which has the advantage of not requiring you to install a driver for the board, as Windows takes care of it all on its own. The first time you connect the board, Windows detects it and installs the driver. On my own computer, this didn't happen quite as it was meant to, but that's not a problem. By disconnecting the board and then reconnecting it, mikroBootloader did end up finding it, even though Windows asked me to reboot my computer (which I refused).

Then the procedure is simple: first run the mikroBootloader utility on the computer, then connect the board and press its reset button. As soon as the USB icon alongside '1 Wait for USB link' goes red, click button '2 Connect' just below. Then you can select the file to be loaded using button 3 and start the file transfer by clicking button 4. The transfer is quite fast, ending with a window saying that everything has gone alright. Click OK to end. Repeat this whole process each time you want to update your firmware.

If you know me a bit by now, you'll already have guessed that what bothers me here is that it takes too many clicks. When you're in the middle of developing a piece of software, you often have to reprogram the microcontroller with the same file (but modified, of course), in which case you have to go through the whole sequence of reset

– connect – select file – load – OK, which soon becomes trying. For your convenience, as well as my own, I've suggested to MikroElektronika that they should simplify the procedure, and they've promised to look into my request. Watch this space...

Programming via an external programmer

The bootloader does not allow you to debug the software, so it may be worthwhile using the mikroProg programmer/debugger (\$ 99) instead of the bootloader. This programmer, housed in an attractive white case, is supported by the mikroProg Suite for PIC utility (v2.10, **Figure 4**), which also lets you modify lots of parameters that are not accessible via mikroBootloader. You can use this utility on its own, or run it directly from the IDE. If you click on the Build and Program button, you won't have to do anything else.

To my great surprise, loading my executable into the microcontroller using mikroProg took over five times longer than with the bootloader (32 s instead of 6 s) and I haven't found any options for speeding this up. What's more, by default it overwrites the bootloader, so remember to enable the

Boot FLASH Write Protect option in mikroProg Suite if you want to keep this feature. Obviously, it is possible to reprogram the bootloader using mikroProg, but then you end up overwriting the program...

The programmer can power the mikroMMB board, but only at 3.3 V. In this situation, it works, but the display isn't very bright – it's almost black, in fact. To get better brightness, you can power the board via its USB port.

Conclusion

MikroElektronika's collection of PIC32 tools is very comprehensive. It includes several development boards, an integrated development environment (IDE), a number of software utilities, and an *in situ* hardware programmer/debugger. The IDE include a C, PASCAL, or BASIC compiler and a debugger/simulator. All these tool are available at very affordable prices, and if you already have one MikroElektronika development tool, you're entitled to a reduction on certain others.

MikroElektronika does have the advantage of making PIC32 development relatively simple and user-friendly. I managed to produce the whole of my application without

once having to consult the data sheet for the microcontroller or display. Any imperfections in my application are undoubtedly my own fault.

Despite all the care the manufacturer has taken with them, these products are not perfect, the mikroC PRO editor in particular leaving something to be desired. Other tools like Visual TFT or mikroBootloader still have some way to go before they can become really powerful and useful. These tools still don't come up to professional standards – but fortunately the prices aren't at professional levels either.

Note that the MikroElektronika products are intended for use with Windows only.

(110729)

Internet Links

- [1] Super Arduino : www.elektor.com/110661
- [2] Wavelets: www.polyvalens.com
- [3] Source code: www.elektor.com/110729
- [4] MikroElektronika: www.mikroe.com

Let's produce something original

In order to assess a tool chain properly, we need to put it to work seriously, and the best way is produce a real application. With its audio input, the graphic touch screen, its calculating power, and direct memory access (DMA), the little mikroMMB board seems ideal for building a small oscilloscope, a spectrum analyser, or even both. But that's rather lacking in originality, don't you think? That's why I decided to produce something never before seen (in Elektor): a wavelet analyser... and touch-sensitive to boot (**Figure 5**).

So what are wavelets?

Before we go on, a little warning: the following is an 'extreme simplification' which will probably make experienced mathematicians frown. If you're one of those, then I'm sorry, but I only have a few lines available here...

Faithful Elektor readers will certainly already have heard about Fourier analysis of a signal. Fourier analysis makes it possible to break a signal down into an (infinite) series of sinewave signals

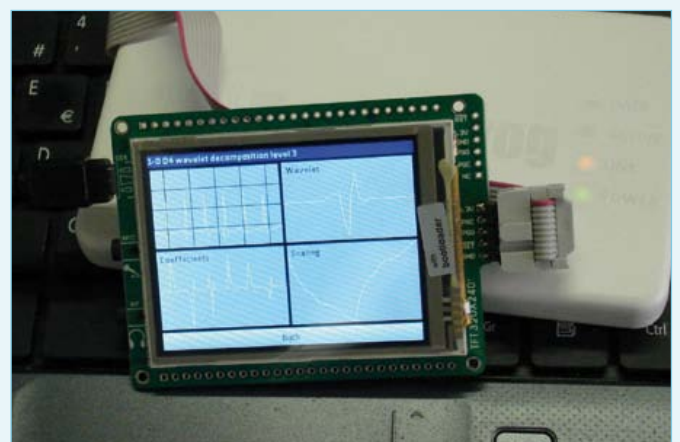


Figure 5. The mikroMMB board connected to the mikroMMB programmer. The display is showing the test application's second screen. Note that the board is also connected to a USB port (on left) to obtain better brightness.

at different frequencies, phases, and amplitudes. This analysis makes it possible, among other things, to determine the spectrum of a signal.

This analysis, also called Fourier transform, is performed on the whole of the signal which we first have to record, but it doesn't tell us at which moment a given frequency was present. It merely allows us to confirm that such or such frequencies were present in the signal.

Several techniques have been developed to improve the temporal resolution of Fourier analysis. The simplest method is to chop the signal up into several pieces and analyse these one by one. This works quite well, but the very act of chopping it up introduces some errors. To limit the damage, various techniques have seen the light of day for cutting the signal up, but they do complicate the analysis. In this case, we speak of short-term Fourier transform.

The underlying reason for the lack of temporal resolution in Fourier analysis is the signal this technique is based upon: the sinewave. (The cosine wave is a sinewave with a 90° phase shift.) Sinewaves have an infinite duration — they are not limited in time. By using another basic signal that is time limited, it's possible to obtain an analysis with temporal resolution. This is where wavelets come in.

“Wavelet theory”, continuous wavelet transform (CWT) to be more exact, was formulated only around 30 years ago. It is very similar to the short-term Fourier transform, where it replaces the “Fourier” sine-wave by another function, the wavelet. This function is not precisely defined, but must satisfy a certain number of criteria. It would take too long to expand here on the maths behind the wavelet transform, but take my word for it, such a function resembles a little oscillation, a little wave or wavelet.

Instead of varying the frequency of the sinewave in order to scan

the whole spectrum of the signal to be analysed, as in Fourier analysis, the wavelets are translated and expanded to scan the duration and spectrum of the signal to be analysed. To make this explanation a little easier to grasp, compare the signal to a building. The wavelet transform now makes it possible to break down this building into bricks of identical shape, but different sizes. This basic shape is the wavelet, the size of the brick corresponds to the expansion, and its position in the building, to the translation.

To make it easier to rapidly calculate CWT by computer, the mathematicians have developed the discrete wavelet transform (DWT). It can be demonstrated – a task I shall spare you – that this transformation corresponds to passing the signal through a bank of constant-Q filters. Remember, the quality factor Q of a filter is the ratio of the filter centre frequency to its bandwidth. Think, for example, of an audio equalizer where the ratio between the setting centre frequencies is an octave.

Figure 6 attempts to sum up the principle graphically. Several algorithms exist for calculating the DWT, but to my way of thinking, one of the most elegant is Lifting. This algorithm makes it possible to use any wavelet without modifying the heart of the transformation algorithm, and in addition, to re-use the same algorithm for the inverse transformation.

Are you finding this all a bit abstract? Well, just remember that the JPEG2000 image compression standard is based on wavelets and the Lifting algorithm. Wavelets are very effective in data compression.

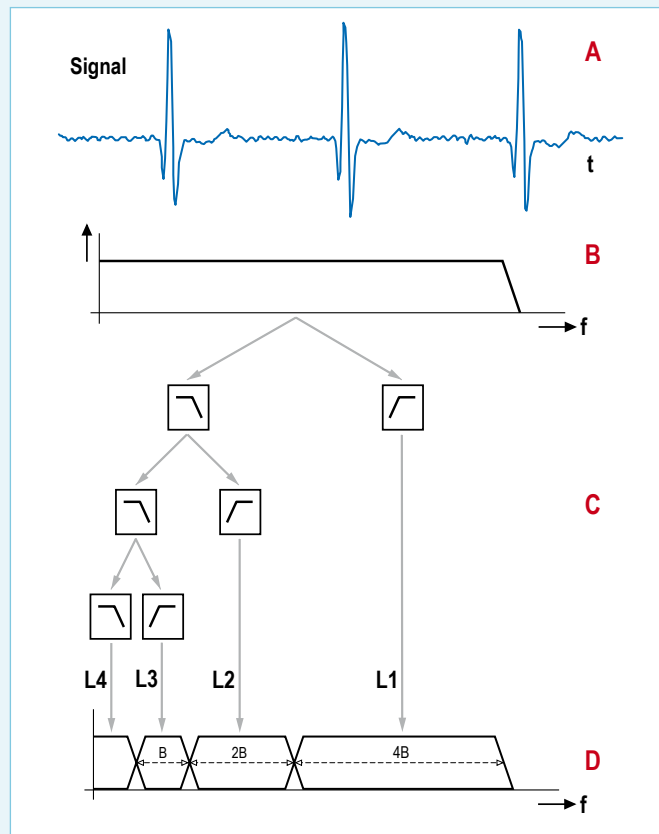


Figure 6. Graphical representation of one-dimensional discrete wavelet transform. The signals (with their labels) Signal and L1–L4 are reproduced on the test application screens. A: input signal; B: spectrum of the input signal; C: discrete wavelet transform visualized as an iterative bank of filters; D: spectrum of the constant-Q filter bank.

The test application

There, now you have a (vague) idea what a wavelet is, let's move right along to the three-screen application that is going to let you find out a bit more about these mysterious functions.

The first screen displays the input signal and the breakdown of it

into wavelets on four levels (see **Figures 6 & 7**). There are also four buttons that let us choose between four different wavelets. The screen title indicates the wavelet being used. When you press on one of the windows, you move on to another screen. Pressing on the Signal window opens the third screen where you can examine the signal more closely. Touching one of the bottom four windows opens the second screen.

This second screen shows four windows. The top left-hand one shows the input signal, the bottom left-hand one displays the contents of the window that brought us to this screen (also shown in the screen title). The right-hand windows show the wavelet (top) and its scale function (bottom). These two graphs are obtained by calculating the inverse of the wavelet transformation of a pulse. As you've understood everything just fine up till now, I'm sure you'll have guessed that we're in fact dealing here with the pulse responses of filters L4 and L3 (or L2 or L1, since they're identical) from **Figure 6**!

Touching one of the four windows makes it possible to study the signal displayed in more detail on the third screen. The Back button takes you back to the first screen.

And lastly, the third screen shows a single signal in close-up; the title gives its name. The Back button takes you back to the first screen.

The input signal is a recording of an electrocardiogram found on

the Internet. This type of signal is often analysed using wavelets. Producing a proper analogue input for capturing your own signals – now there's a good exercise that we're going to leave up to the reader.

Four wavelets are accessible via the four buttons on the first screen. In the source code, you'll find other wavelets, in particular, some variants on the Cohen – Daubechies – Feauveau (CDF) family. It's easy to attach them to the buttons – see how to do this in the screen2.c file. The four wavelets I've chosen as defaults give a good illustration of the variations possible in the forms of the

wavelets. Personally, I'm very keen on the Daubechies-4 (D4) for its irregular form. Note that this wavelet possesses fractal properties.

As indicated above, the algorithm for calculating the wavelet transform is based on the Lifting technique. I don't have enough space here to explain in detail how it works, but you can find fuller details from [2]. The special feature of Lifting is that if it is executed in reverse, it calculates the inverse wavelet transform. This makes it possible very simply to check the form of the wavelet (see above). The algorithm is implemented in

the file fltw.c (Fast Lifting Wavelet Transform), the wavelets are in the wavelets.c file.

The test application source code is available from [3].

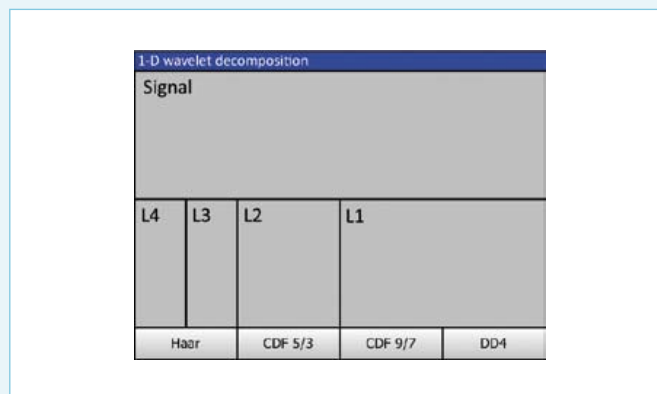


Figure 7. The correspondence between the signals in Figure 6 and the five windows of the first screen (not counting the intro screen).

EURO
CIRCUITS

**The European reference for
PCB prototypes & small series**

www.eurocircuits.com

Time / Interval Meter with ATtiny

Vladimir Mitrovic (Croatia)

This project goes to show what little hardware is sufficient to build a versatile Time and Interval Meter with a user-friendly LCD readout. An AVR microcontroller type ATtiny2313 measures the time interval between two consecutive logic level transitions of pulses applied to input pins PD2 and PD3. In the μs (microseconds) range, time intervals from 10 μs to 30 minutes (!) can be measured with a resolution of 1 μs . In the ms range, the available range is 1 ms to four hours at 1 ms resolution. In the circuit, DIP switch block S2 determines the settings of the instrument, as follows:

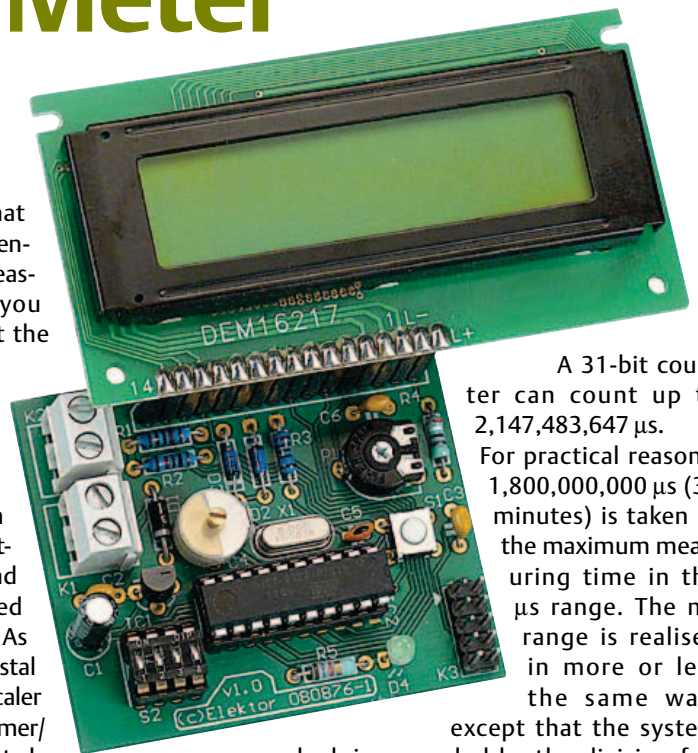
- S2 1-8: selects pulse edge (Rising or Falling) starting the measurement.
- S2 2-7: selects pulse edge (Rising or Falling) halting the measurement
- S2 3-6: measurement unit and resolution (μs or ms)
- S2 4-5: measurement mode (Continuous or Oneshot/Hold)

See **Table 1** for a detailed explanation of the configuration switch functions and options. The measured time is displayed on an LC display with two rows of 16 characters. The first row shows the selected time interval, the measuring unit and the measuring mode, the second, measured time. LED D4 is on during the measurement and off between two measuring intervals, as well as during the Hold period. Its main pur-

pose is to show that 'something is happening' during long measuring periods. If you want you can omit the LED and R5.

In the μs range, the 8-bit Timer/Counter0 inside the ATtiny2313 is set to Normal mode with Compare Match output B activated, and counts the prescaled system clock pulses. As an 8-MHz quartz crystal is used and the prescaler divisor is set to 8, Timer/Counter0 increments by 1 each μs . Output Compare Register B is set to 255 which toggles the Output Compare pin (OC0B) every 256th pulse. OC0B pin is internally connected to the Timer/Counter1 input, and the Timer/Counter1 counts these pulses on the OC0B pin in a 16-bit resolution. This way we get a 25-bit hardware counter (16-bit T/C1 + OC0B bit + 8-bit T/C0).

An additional 6 bits worth of resolution is realised in software. During the measurement the program runs in a loop, waiting for the flag to stop the measurement and constantly pooling the Timer/Counter1 Overflow Flag, TOV1. If TOV1 is set, the program increments the 6-bit SW counter by 1 and clears TOV1. No interrupt is used here because it could delay the recognition of the Stop condition.



A 31-bit counter can count up to 2,147,483,647 μs .

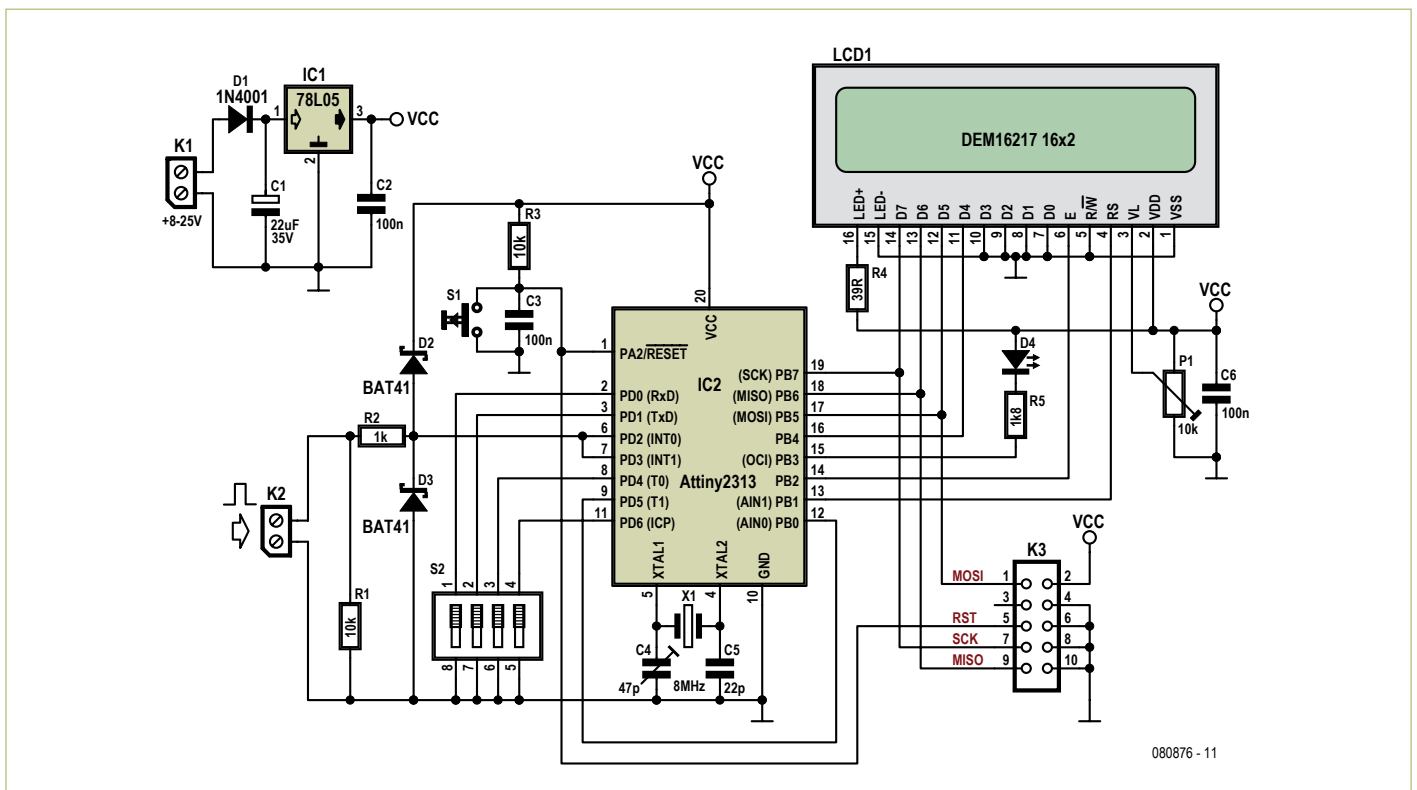
For practical reasons, 1,800,000,000 μs (30 minutes) is taken as the maximum measuring time in the μs range. The ms range is realised in more or less the same way,

except that the system clock is prescaled by the division factor 8. After the measurement the result is divided by 125, giving the maximum value of 17,179,869 ms. For practical reasons, 14,400,000 ms (four hours) is taken as the maximum measuring time in the ms range. The measuring starts when a falling or rising edge is recognized at ATtiny pin PD2 (the triggering slope depends on the setting of S2 1-8), whereupon external interrupt vector INT0 is executed. The measuring stops when a falling or rising edge is recognised at pin PD3 (the triggering slope depending on the setting of S2 2-7) whereupon external interrupt vector INT1 is executed. The program developed for the project is a free download from [1]. It is written in BascomAVR, with embedded assembler code for interrupt and other time-critical routines inserted. In the main loop, the program

Table 1.

S2 1-8	S2 2-7	S2 3-6	S2 4-5	Measured time interval
off	off	x	x	From falling edge to the next falling edge.
off	on	x	x	From falling edge to rising edge (time interval between two positive pulses).
on	off	x	x	From rising edge to falling edge (duration of positive pulse).
on	on	x	x	From rising edge to the next rising edge.
x	x	on	x	Measurement in μs ($t(\text{min})=10\mu\text{s}$, $t(\text{max})=1800\text{s}$ (30m)).
x	x	off	x	Measurement in ms ($t(\text{min})=1\text{ms}$, $t(\text{max})=14400\text{s}$ (4h)).
x	x	x	off	Measures continuously: when one measurement is finished and the result is displayed, a new measurement is started.
x	x	x	on	Just one measurement: when one measurement is finished and the result displayed, the program holds.

(x = don't care)



repeatedly monitors the switches in S2, initiates a measurement and displays the result. If any change in the switch settings compared to the previous reading occurs, the new measuring range, mode and/or starting/stopping slopes are determined and the first line on the display is updated. At the beginning of the measurement routine, Timer/Counter counting registers and the software counting register (for bits 26-31) are cleared, OC0B bit is reset and the external interrupt INTO is enabled. After that, the program loops, waiting for the INTO interrupt to occur which will trigger the INTO interrupt routine. In this routine, the Timer/Counter0 is started, some counters are set to initial values, LED D3 is switched on, INTO interrupt is disabled to disable retriggering and INT1 interrupt is enabled instead. The program continues to loop, waiting for the INT1 interrupt to occur which will trigger the INT1 interrupt routine. In this routine, the Timer/Counter0 is stopped and further external interrupts are disabled.

Timer/Counter1 is started at the beginning of the program and there is no need to stop and restart it during the program execution: as it counts the pulses from the Timer/Counter0, it will be started and stopped simultaneously.

A measurement is finished when the measuring routine senses that both external interrupts are disabled. The result is col-

lected from the hardware (HW) and software (SW) registers, LED D3 is switched off, some calculations are done and the measured time interval is displayed on the lower display line.

On accuracy, the HW part of the counter counts the clock pulses and there is no possibility of false counting. The SW counter extension counts the Timer/Counter1 overflows with a few cycles delay. However, when the timers are stopped, the SW part of the counter is synchronised with its HW part. The INTO and INT1 interrupt routines have been written very carefully, and there is exactly the same delay from the start of the routines to the instant Timer/Counter0 is started in one, and stopped in the other routine. There is a small delay from the instant the external interrupt is triggered to the instant the interrupt routine starts to execute. The duration of this delay depends on the instruction that was executing the moment the interrupt is triggered. Instructions execute in 1-4 clock cycles, so the difference can be up to 3 clock cycles. But, as Timer/Counter0 counts the system clock divided by 8, this will cause a ± 1 error on the last digit only in both ranges (μs and ms). So, the only factor that significantly influences the overall accuracy is the crystal accuracy – consider that when you see 't = 123456789 μs ' on the display.

The second thing to be considered is measuring time, which can last for hours in the

ms range. Therefore, a special SW counter is enabled during the waiting loops (there are two waiting loops during the measuring: wait for the Start signal and wait for the Stop signal). This counter counts up to the defined maximum measuring time and, if it elapses, stops further measuring and displays 't > 14400000'. Initially, the maximum measuring times are set to maximum permissible values, i.e. 30 minutes in the μs range and four hours in the ms range. Like the other switches, S2 4-5 is read before the measuring starts. However, the program acts in a different manner if this switch is closed (and the oneshot/hold mode is activated):

The maximum measuring time counter is blocked while waiting for the Start signal, but it is enabled again while waiting for the Stop signal (allowing indefinite waiting time for a non-repetitive pulse to occur, although the pulse duration is still limited to the max measuring time).

When the result is displayed, the program waits in a loop, constantly monitoring S2 4-5 and continues as soon as the switch is opened.

The program *Tmeter_Elektor.bas* supplied for the project should be programmed into the ATtiny2313 microcontroller before the first use. Make sure to set Flash Fuse bits CKSEL3...0 to the proper value for an external crystal resonator (CKSEL3...1,0 = 1111) because the internal RC Oscillator is selected

COMPONENT LIST

Resistors

R1,R3 = 10kΩ
 R2 = 1kΩ
 R4 = 39Ω
 R5 = 1kΩ8
 P1 = 10kΩ preset

Capacitors

C1 = 22μF 35V radial
 C2,C3,C6 = 100nF
 C4 = 47pF trimmer
 C5 = 22pF

Semiconductors

D1 = 1N4001
 D2,D3 = BAT41

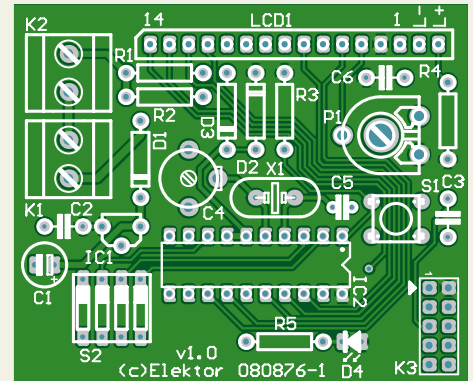
D4 = low current LED

IC1 = 78L05

IC2 = ATtiny2313, programmed, Elektor Shop # 080876-41

Miscellaneous

S1 = tactile switch, 6mm footprint
 S2 = 4-way DIP switch
 K1,K2 = PCB mount terminal block, lead pitch 5mm (0.2")
 K3 = 10-way boxheader
 X1 = 8MHz quartz crystal
 LCD1 = LCD module, 2 lines, 16 characters, e.g. DEM16217
 PCB, ref. 080876-1



by default. It is very important that the crystal's frequency is exactly 8.000 MHz, as this determines overall accuracy. A trimmer C4 is provided to fine tune the crystal frequency. If you are satisfied with the crystal's accuracy, replace C4 with a fixed capacitor. You can also use a precise quartz oscillator to drive the microcontroller; in this case omit C4, C5 and X1 and connect the oscillator output to the XTAL1 input. Adjust P1 for optimum display contrast.

In the program, the maximum measuring times are defined as constants at the beginning:

```
Const Tmax_us_default = 1800
'max measuring time for us
range [s]
Const Tmax_ms_default = 14400
'max measuring time for ms
range [s]
```

Shown values are the recommended maximum measuring times but the user may replace them with appropriate lower values, for example:

```
Const Tmax_us_default = 60
'max measuring time for us
range [s]
Const Tmax_ms_default = 300
'max measuring time for ms
range [s]
```

This will shorten the maximum measuring times to 60 s and 300 s in the μs and ms ranges, respectively. Of course, it is necessary to recompile the program and reprogram the microcontroller after such a change. It is very important to choose the optimal maximum measuring time because the program loops and waits for the maximum measuring time to elapse if the input

level is constant (or it will wait up to 2x maximum measuring time to elapse in the case when level transitions occur too slowly). Another way to change the maximum measuring time is to program appropriate values as 16-bit unsigned binary values in the microcontroller's EEPROM, refer to **Table 2**. These values should be given in seconds.

The program logic is as follows:

- If the EEPROM is empty (FFh), the maximum measuring times defined in the program will be used.
- If the EEPROM value for the μs range is greater than 1800, the maximum measuring time will be used for that range as defined in the program.
- If the EEPROM value for the ms range is greater than 14400, the maximum measuring time will be used for that range as defined in the program
- Any EEPROM value that is within the allowed range will be used instead of the maximum measuring time for the corresponding range as defined in the program.

The configuration switches are read at the beginning of the main program loop. This can be confusing with long-lasting measuring because the old settings and the previous result are displayed until the measurement under way is finished and the result is displayed. You can speed up the process if

you reset the microcontroller right after the new switch setting. To reset the microcontroller, press the Reset switch.

In the Oneshot/Hold mode, you can initiate a new measuring in the following ways:

- open S2 4-5 to restart continuous measuring or
- reset the microcontroller to reinitiate the Oneshot/Hold mode (you cannot open and close S2 4-5 quickly enough to restart the Hold mode this way)

Finally, the input pulses should have a TTL or CMOS compatible logic levels. It is very important that edges are well-defined, i.e. are free from bouncing and ringing.

(o8o876-1)

Internet Link

[1] www.elektor.com/080876

Downloads & Products

PCB: 080876-1 PCB design, download from www.elektor.com/080876

Programmed Controller: 080876-41 ATtiny 2313, ready programmed

Software: 080876-11 Bascom & assembly code program

Table 2.

EEPROM address	EEPROM value	
000 0000	LSB	Max. measuring time for μs range [s]
000 0001	MSB	
000 0010	LSB	Max. measuring time for ms range [s]
000 0011	MSB	

Hexadoku

Puzzle with an electronics touch

Hopefully in time for the Christmas holidays, here’s a fresh Hexadoku puzzle to get you through the odd moment of boredom after all that gift unpacking, lengthy dinners and watching the same childhood movies again on TV. Simply enter the right numbers in the puzzle below. Next, send the ones in the grey 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 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that’s 0-9 and A-F) occur once only in each row, once

in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation. Correct entries received enter a draw for a main prize and three lesser prizes. All you need to do is send us the numbers in the grey boxes.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for one Elektor Shop voucher worth £ 80.00 and three Elektor Shop Vouchers worth £ 40.00 each, which should encourage all Elektor readers to participate.

Participate!

Before February 1, 2012, send your solution (the numbers in the grey boxes) by email, fax or post to
 Elektor Hexadoku – 1000, Great West Road – Brentford TW8 9HH
 United Kingdom.
 Fax (+44) 208 2614447 Email: hexadoku@elektor.com

Prize winners

The solution of the November 2011 Hexadoku is: **40F58**.
 The Elektor £80.00 voucher has been awarded to M. Murphy (Ireland).
 The Elektor £40.00 vouchers have been awarded to Tony Jaques (United Kingdom), Phillip Kelly (USA) and J.L. Kruger (Germany).
 Congratulations everyone!

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

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

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

Philips 'Colette' Portable Radio

Not forgetting Annette, Evette, Babett

By Jan Buiting (Editor, Elektor UK & US editions)

Tubes are great in radios and amplifiers as long as you have plenty of space in your kitchen or living room, not forgetting a nearby AC power outlet. Most people would say that's because of "them tubes' inefficiency, you know, the heater current and all that". True, a medium sized 1950s tabletop radio consumes anything between 25 and 100 watts of AC power for just 1 to 5 watts of audio power to enjoy or bother the neighbours, which makes it a tad difficult to use in a picnic location or on the beach. The trouble was with the large amount of energy required to make the filaments in the tubes heat up to temperatures where sufficient emission is obtained to establish an electron flow. Small signal triode and pentode tubes easily waste more power due to the filament than to anode current.

Not long after WW2 Philips, a leading tube manufacturer in the Netherlands, in their famous Natuurkundig Laboratorium ('NatLab', in English: Physics Laboratory) developed and perfected a series of miniature low-power tubes employing direct heating instead of indirect, specifically for use in portable radios. These tubes identified by the first letter D in their type code (rather than E for 6.3 V indirect heating) operate at a filament (heater) voltage of just 1.4 VDC and 50 mA current consumption (typical), with the cathode doubling as the heater. Four of these small tubes could be connected in series for running off a standard car battery, which was 6.2 volts at that time, not 12 V. The anode ('plate') voltage was typically between 45 and 90 volts, which was considered low at the time and easy to furnish by a rechargeable battery. US equivalents also appeared like the 3S4 (DL92).

Clearly the Miniwatt D series of 'battery receiving tubes' opened the way to portable radio design. Towards the late 1940s Philips started to release its first portable radios, most of these models were housed in rather dull looking Bakelite cases and offered MW, LW and SW reception only. Expensive at the time and for the well to do only, these radios added considerably to the success of the little tubes of the 'cold and dark' variety. In 1951, a piece of equipment based on D tubes was launched for the professional market: the backpack PMR type SDR314 I discussed in a previous Retronics piece (November 2006).

Meanwhile a 72-page book introducing the Miniwatt D series to engineers rather than scientists was published around 1951. It became famous probably because of the solid theory, individual presentation of tubes and nice application examples of radios you could build at home with a complement of these D tubes:

- DK92 self-oscillating heptode mixer;
- DF91 RF pentode;
- DAF91 diode/AF pentode demodulator



- DL92 and DL94 AF power pentodes;
- DM70 and DM71 tuning indicators or 'magic eyes'.

The book is now a rare find, I have pictured it here (second edition 1954).

Moving on in time, towards 1955 the first official VHF FM broadcasts were aired, in Europe, initially in Germany. Not surprisingly, Philips' first portable FM radio featuring the brand new 87-100 MHz FM band got designed there. It was called 'Colette', following a widespread craze started around 1950 in the USA and lasting well into the 1970s to add the suffix "-ette" to product names, like DryCleanette, kitchenette, Sedanette, Echolette, Corvette, Mobylette, Autoette, Wagonette, sandalette, The Ikettes). In this case, the marketing people at Philips Germany did not want or need the 'endearing diminutive' but still rode the wave by using French girls' names for their range of portable radios. Technical staff and radio & TV repair men on the other hand generally stuck to Philips' established type code system. Colette technically was model LD562AB (later changed to L5D62AB) where

- L = radio, portable;
- 5 = price class indicator (0-9);
- D = manufactured in Germany (X =Belgium/Netherlands);
- 6 = year in decade (195x);
- 2 = model;
- A = AC powered;
- B = battery powered.

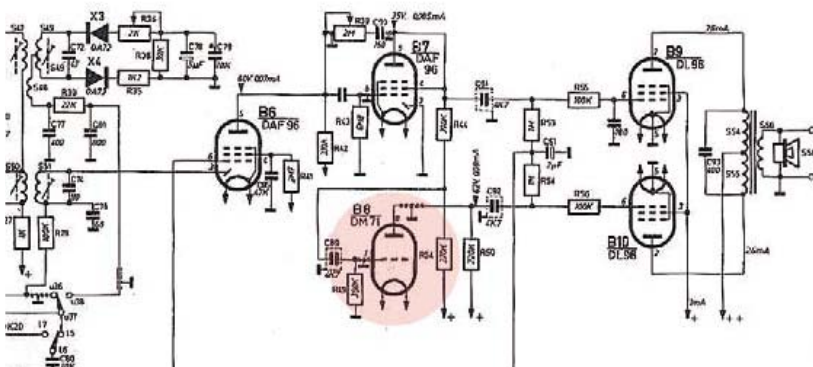
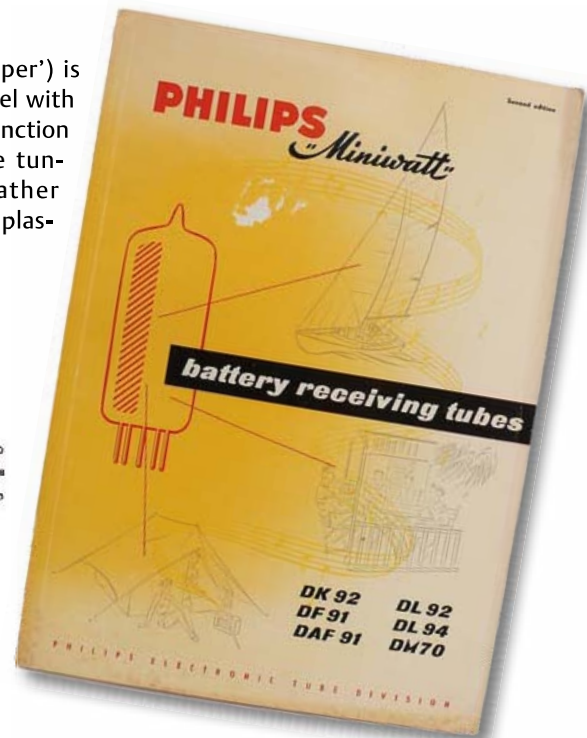
Retronics is a monthly column covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcomed; please send an email to editor@elektor.com

(1956) e and Georgette



Colette with her prestigious FM band coverage proved hard to get, probably due to her price tag of 398 Dutch guilders (roughly two month's wages for a factory worker). Not surprisingly, lesser priced "sisters without FM" were also available called *Annette*, *Babette*, *Evette* and *Georgette*. Not meaning any offence to these young ladies, they were pitched in the '3' and '4' price classes. In Holland,

Colette ('Klipper') is the only model with stylish dual-function knobs on the tuning scale, rather than cheaper plas-



Philips' home country, a number of earlier and almost identical radios got named after small boats like *Jol* (dinghy), *Klipper* (clipper), *Flying Dutchman*, *Regenboog* (Rainbow), *Valk* (a medium size open sailing boat), and *Boeier* (a Frisian vessel). As opposed to the German division the Dutch did not actually put these names on the radio grille — they only appeared in sales brochures.

tic thumbwheel controls beside it. The radio contains ten D tubes, one diode and two selenium rectifier bridges. In portable operation, the heater voltage is supplied by a 1.2 V 'Deac' rechargeable battery with a capacity of 6 Ah. The anode voltage comes from by a 90 V battery. The Deac is a nickel cadmium battery and notorious for its spillage of heavily corrosive substances that



attack the inside of the radio, slowly rendering Klipper a wreck over 25 years of neglect. After opening the back cover of a tube radio like the Colette (Klipper), to look at the toxic remains of a forgotten Deac is a depressing sight. With some drawbacks the Deac can be omitted and its function taken over by one or two D size (LR20) 1.5 V batteries and a large electrolytic capacitor across them. Dry cells were optional originally and a dual holder is provided within the radio case. The anode battery is a different problem and today usually takes the form of a switch-mode step-up voltage inverter with proper screening to prevent radio interference. Various designs float around on the web as well as kits on Ebay, some visually perfected, wrapped in an Eveready carton you can't distinguish from real.

Colette also has an internal AC power supply section for the 90 V plate voltages as well as (primitive) Deac charging, where the Deac doubles as a smoothing device — with directly heated tubes you do not want AC on the cathodes. A complete charging cycle is stated to take 14 hours at 0.6 A. The radio can also charge up while playing. My Colette plays on AC power only, it has no Deac or 90 V battery. The sound, particularly on FM, tends to surprise people pleasantly probably because they associate a portable radio of this age with tinny sound. By contrast, the sound from Colette is warm and quite full, with a good dose of loudness thrown in at lower volume settings. The biggest surprise is the amount of bass you get from the measly 400 milliwatts of AF output power.

The radio can be operated in 'economy' mode by pulling out the volume control. This switches off half of each of the two filaments of the DL96 output tubes, with the obvious effect of saving battery capacity at the cost of some AF output power.

Collette has a design quirk. While you would expect the DM71 magic eye to act as a stylish tuning indicator, in reality it's just a green on/off light. The circuit diagram shows what's going on. The DM71 actually functions as a phase splitter for the DL96 balanced output amplifier. This is probably a workaround for the DAF96 (B7) supplying insufficient drive to the balanced DL96s, which in turn is the result of low signal yield from the FM and AM detectors.

As for period design features you can mention to the Antiques Road Show presenter, the retractable antennas with plastic protective end covers can be aligned at any angle between 0 and 180 degrees to optimise FM reception. They form an open dipole and if the signal is still too weak you can connect an external antenna via a ribbon cable. The round holes in the side panels allow a car radio antenna and a car battery (6 volts!) to be connected. Later models offered more connectivity but USB is not provided as standard. You can tell Colette is off duty or on the way to a picnic or beach rave by the closed lid in front of the tuning scale.

My Colette is in good condition overall with just some scuffs at the front near the underside. The soft rounded corners, light ochre case (once green?) with taupe red hard plastic parts and the gold grille and frame immediately identifies it as 1950s. No repairs were necessary to make this beauty come alive again after 30 years on a dusty attic, except replacing both DAF96s and tidying the battery compartment.

Colette weighs an astonishing 6.5 kgs (14 lbs) with two D batteries installed. I am always on the lookout for her FM-less L4X and L3X sisters to complete the family picture. Sadly for me, they are playing hard to get too. Any help appreciated.

(110705)



Product Development

By Gerard Fonte (USA)



Many hobbyists dream of creating some special widget and making a million dollars with it. It has been done. Look at Bill Gates and Steve Jobs. However, it's not easy. It requires a lot of effort, a good product and often, some luck. (Originally, CP/M was the operating system of choice for home computers until IBM entered the market and decided to use MS DOS from upstart Microsoft.) And

while 80% of new companies fail in the first year because of poor management, 100% fail if they have a rotten product.

What Makes a Good Product

Fundamentally, in our capitalist society, a good product is defined as being worth equal-to or more-than its selling price. It's important to realize that 'worth' is a completely subjective measure. Consumers are satisfied as long as they feel their money was well spent. If not, they won't buy from you again. I'm sure you've had both a good and bad experience in this area. You've had a meter, or a stereo or car that always worked well. And you've one that didn't. Large companies have big marketing firms that can overcome a product's deficiencies. Sometimes and for a limited time. The US car manufacturers found that out the hard way. Small companies don't have that large market to act as a buffer.

Worst of all, start-up companies have to break into a market that is already populated. For them, they have to provide a product that is superior in some way just to get noticed at all. A new product that is exactly the same as an existing product in price and performance simply won't sell. It's just human nature to stay with a manufacturer that a consumer knows and likes. So, the first rule in product development is to provide the consumer with something significantly better in price and/or performance. The only way to draw away satisfied buyers is to make them unsatisfied.

Making a Good Product

Probably the biggest problem with 'hobbyist' products is that they have poor 'look and feel'. People expect a quality product. This means attention to detail. For example, switches have to be appropriate in size and type. A rotary or slide switch for 'On/Off' is a poor choice. Use a push-button or rocker type. Look around and see what mass-produced products look like and make your product similar. Lay out the front panel in a logical and intuitive manner. Make sure that all the printing is clear and concise. There are way too many things to list here. Just design your product around the viewpoint of the user. Don't rationalize away problems by saying that the buyer can put up with a bad choice because your selling price is low. Because now you are asking the customer to choose between a good price and a 'good' product. You have to provide a good product AND a good price.

The second problem is performance. Most hobbyists don't think about the long term use of their device. Will your widget survive a year's use under normal conditions? How about abnormal conditions? Having a good understanding about reliability is important. There are plenty of books of this subject. But good, conservative design goes a long way.

Hobbyists also don't mind tinkering with a design while most users find this unacceptable. Having to adjust a receiver's frequency every 10 minutes is a bad thing. Other radios don't have this problem, so yours shouldn't either. Again, look around. There are thousands of products in your household and probably at least a hundred are electrical in some way. Learn from these successful products. (They have to be successful or you wouldn't have bought them.) Examine the products that compete for your potential customers. And make sure your product performs better and costs less.

Naturally, everyone tends to think that their brainchild is beautiful. Unfortunately, beauty is subjective. Spending a lot of time and effort creating something doesn't automatically make that something a sellable item. In fact, it's a liability if it takes a lot of time and effort to put together.

A good product is easy to build. This is called 'design for production'. For example, there are no screws that are impossible to fasten with ordinary tools. All parts are easily accessible for test, repair or replacement. (One product I saw had surface-mount components under a through-hole IC. That's right, you had to unsolder a 40-pin DIP to get to these parts.) Design for production is really a thorough application of common sense. (Put these surface mount parts on the other side of the circuit board.)

Closely associated is 'design for test'. Before your product goes out the door, you have to be sure it works. This is done by testing it. A good product requires few tests. With today's microcomputers, a self-test routine is both fundamental and mandatory. The faster you can get your product off the production floor, the cheaper you can sell it. Time is indeed money when it comes to testing. Labor costs are probably the biggest factor in pricing your product.

Selling and Marketing

Selling and marketing is a major hurdle for most entrepreneurs. You may have an incredible product, but if no one knows about it, you can't sell any. It takes time and money to advertise and market your product. Note that selling/advertising is talking to your customers while marketing is listening to your customers. Hopefully, you've done some market research before you designed and built your product. If you have, then you know that there are people who want to buy what you have to sell. Now all you have to do is inform them that their dream product is available for sale. Proper marketing makes advertising and sales much easier and cheaper.

The Big Picture

You can see that product development does not happen by accident. It's an integrated process that consists of a number of elements. And while your newly finished project isn't likely to be a product, it doesn't mean that it can't become one.

(100356)

ELEKTOR SHOWCASE

To book your showcase space contact Elektor International Media

Tel. 0031 (0) 46 4389444

Fax 0031 (0) 46 4370161

BRITISH AMATEUR ELECTRONICS CLUB ARCHIVE



<http://baec.tripod.com/>

The British Amateur Electronics Club Archive Website. Archives from 140+ Newsletters from 1966-2002. Currently have interesting and useful selected articles from 21 Newsletters. Also a section about built electronics projects with schematics and photos. Plus useful info., downloads and links. NO ADVERTISEMENTS!

FIRST TECHNOLOGY TRANSFER LTD.

<http://www.ftt.co.uk>

- Training and Consulting for IT, Embedded and Real Time Systems
- Assembler, C, C++ (all levels)
- 8, 16 and 32 bit microcontrollers
- Microchip, ARM, Renesas, TI, Freescale
- CMX, uCOSII, FreeRTOS, Linux operating systems
- Ethernet, CAN, USB, TCP/IP, Zigbee, Bluetooth programming



HEXWAX LTD

www.hexwax.com

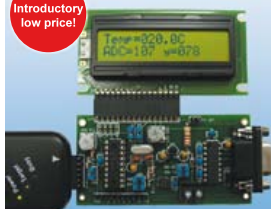
World leaders in Driver-Free USB ICs:

- USB-UART/SPI/I2C bridges
- TEAleaf-USB authentication dongles
- expandIO-USB I/O USB expander
- USB-FileSys flash drive with SPI interface
- USB-DAQ data logging flash drive

PIC demo kit

- Demonstration • Development
- Building block

Introductory low price!



Open source examples in C and Assembler

Includes:
PIC16F1827,
Temperature sensor,
RS232,
LCD display,
Analogue input,
5V regulator

Kit price including LCD £20 or assembled £35 + P&P

www.cstech.co.uk/picdemo

FLEXIPANEL LTD

www.flexipanel.com

TEAclippers - the smallest PIC programmers in the world, from £20 each:

- Per-copy firmware sales
- Firmware programming & archiving
- In-the-field firmware updates
- Protection from design theft by subcontractors



SCOPES and more

HAMEG®

Instruments

A Rohde & Schwarz Company

www.hameg.com

Great Value in
TEST & MEASUREMENT

EASYSYNC LTD.

www.easysync-ltd.com/

Supplier of communications and instrumentation products with specialist expertise in serial connectivity solutions based on USB, CAN and RS232/RS422/RS485 interfaces.

- USB to Serial RS232/RS422/RS485 converter cables.
- CANbus solutions
- Ethernet to Serial Adapters or to USB hubs.
- USB based Logic Analysers, Oscilloscopes & Data Loggers.
- OEM & ODM design services.



FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD.

www.ftdichip.com <<http://www.ftdichip.com>>

FTDI specialise in USB silicon, hardware and software solutions.

- USB WHQL complaint drivers.
 - USB host and slave solutions.
 - Free firmware development tools.
 - USB IC's, modules, cables and turnkey custom solutions.
 - World renowned FOC application support.
- USB MADE EASY



minty
geek
a fresh approach

Cool portable electronic projects and kits



www.mintygeek.com

ELNEC

www.elnec.com

Europe's leading device programmers manufacturer:

- reliable HW:
 - 3 years warranty for most programmers
- support over 65.000 devices
- free SW updates
- SW release: few times a week
- excellent technical support:
 - Algorithms On Request, On Demand SW
- all products at stock / fast delivery



TO BOOK YOUR SHOWCASE SPACE CONTACT ELEKTOR INTERNATIONAL MEDIA

Tel. 0031 (0) 46 4389444

Fax 0031 (0) 46 4370161

www.elektor.com

products and services directory

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
www.coolcomponents.co.uk
www.oceancontrols.com.au
www.maxbotix.com

ROBOT ELECTRONICS

<http://www.robot-electronics.co.uk>

Advanced Sensors and Electronics for Robotics

- Ultrasonic Range Finders
- Compass modules
- Infra-Red Thermal sensors
- Motor Controllers
- Vision Systems
- Wireless Telemetry Links
- Embedded Controllers



ROBOTIQ

<http://www.robotiq.co.uk>

Build your own Robot!

Fun for the whole family!

Now, available in time for X-mas

- Arduino Starter Kits *NEW!!*
- Lego NXT Mindstorms
- Affordable Embedded Linux Boards
- Vex Robotics (kits and components)
- POB Robots (kits and components)

email: sales@robotiq.co.uk Tel: 020 8669 0769



www.elektor.com

TO BOOK YOUR SHOWCASE SPACE CONTACT ELEKTOR INTERNATIONAL MEDIA

Tel. 0031 (0) 46 4389444

Fax 0031 (0) 46 4370161

TYDER

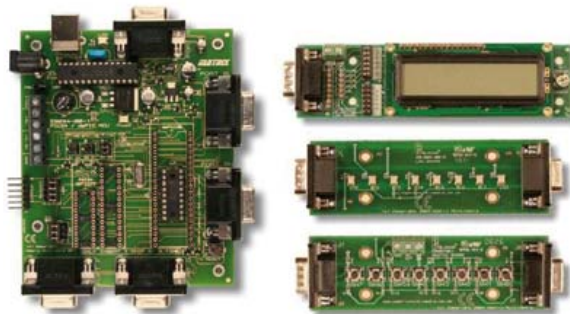
<http://www.tyder.com>

- ONEoverT Digital Filter Design Software (Full version for only £30)
- Design FIRs, IIRs, NCOs, FFTs for DSPs and FPGAs
- VHDL Code Generators
- Makes DSP design simple
- Download demos from website



dsPIC/PIC24-Bundle

Advantageous hardware/software solution for rapid project development



Contents:

- Flowcode 4 for dsPIC/PIC24 (Professional Version)
- USB dsPIC/PIC24 Microcontroller Multiprogrammer
- LCD Board
- LED Board
- Switch Board
- Plug top power supply
- USB cable

This solution is perfect for anyone wanting to develop systems based around Microchip's powerful 16 bit core products. The pack is supplied with a dsPIC30F2011 device, and is fully compatible with the full range of E-block boards and accessories. Datasheets on each individual item are available separately.

Bundle Price:
Only £299.00

15% DISCOUNT to the sum of the individual parts!

Order now at www.elektor.com/dspic-bundle

Going Strong

A world of electronics from a single shop!



Bestseller!

More Kits & Modules available at www.elektor.com/kits

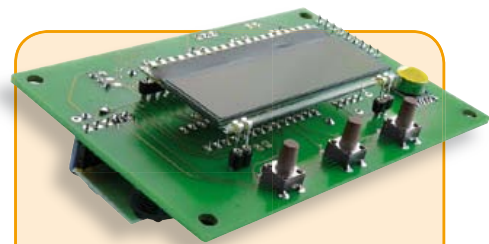
Kits & Modules

For alpha, beta and gamma radiation
Improved Radiation Meter
 (November 2011)

This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly 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.

Kit of parts incl. display and programmed controller

Art.# 110538-71 • £35.50 • \$57.30



USB Long-Term Weather Logger

(September 2011)

This stand-alone data logger displays pressure, temperature and humidity readings generated by I²C 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 • £31.10 • US\$50.20



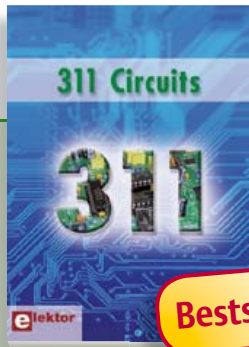
Audio DSP Course

(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 • £115.70 • US\$186.70



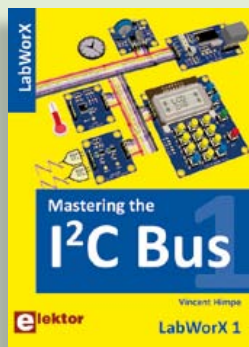
Bestseller!

Creative solutions for all areas of electronics

311 Circuits

311 Circuits is the twelfth volume in Elektor's renowned 30x series. This 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.

420 pages • ISBN 978-1-907920-08-0
£29.50 • US \$47.60

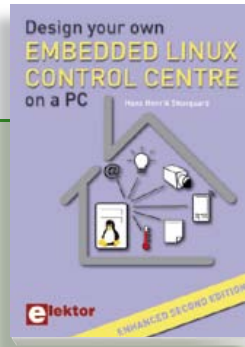


LabWorX: Straight from the Lab to your Brain

Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C 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 I²C 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 I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9
£29.50 • US \$47.60



Enhanced second edition: 180 new pages

Design your own Embedded Linux Control Centre on a PC

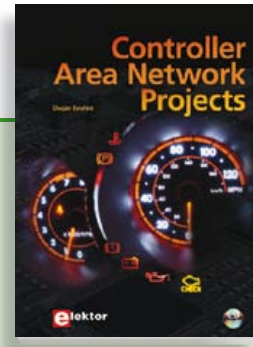
The main system described in this book re-uses an old PC, a wireless mains outlet with three switches and one controller, and a USB webcam. All this is linked together by Linux. This book will serve up the basics of setting up a Linux environment – including a software development environment – so it 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
£34.50 • US \$55.70

More information on the
Elektor Website:
www.elektor.com

Elektor
Regus Brentford
1000 Great West Road
Brentford
TW8 9HH
United Kingdom
Tel.: +44 20 8261 4509
Fax: +44 20 8261 4447
Email: order@elektor.com

elektor

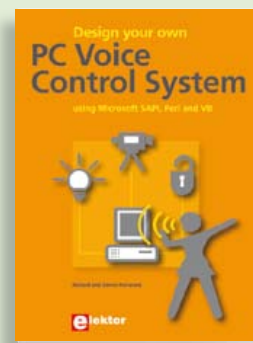


Free mikroC compiler CD-ROM included

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
£29.50 • US \$47.60

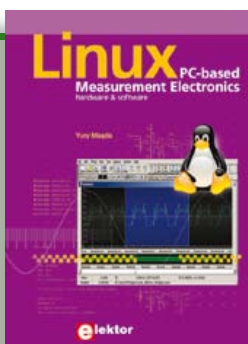


Talk with your computer

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!

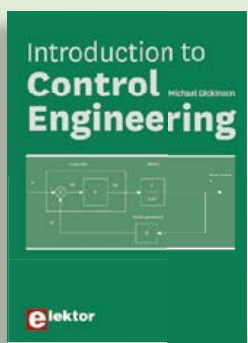
216 pages • ISBN 978-1-907920-07-3
£29.50 • US \$47.60



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!

264 pages • ISBN 978-1-907920-03-5
 £29.50 • US \$47.60



Solutions for control system applications
Introduction to Control Engineering

This book is intended as a source of reference for hardware and software associated with instrumentation and control engineering. Examples are presented from a range of industries and applications. Throughout the book, circuit diagrams and software listings are described, typical of many measurement and control applications. The hardware and software designs may be used as a basis for application by the reader. The book contains examples of PIC, PLC, PAC and PC programming.

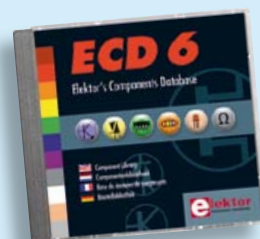
164 pages • ISBN 978-0-905705-99-6
 £27.50 • US \$44.40



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.

ISBN 978-1-907920-06-6
 £34.50 • US \$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 databank applications are fully interactive, allowing the user to add, edit and complete component data.

ISBN 978-90-5381-258-7
 £24.90 • US \$40.20



RFID, NFC, Zigbee, GPS and more
DVD Wireless Toolbox

On this DVD-ROM you'll find a number of technical documents and tools that will enable you to add wireless data exchange to your electronics systems. The choice of equipment depends on the transmission distance: a few centimetres using Near Field Communication (NFC) or Radio Frequency Identification (RFID), tens of metres with the Bluetooth, Wi-Fi or ZigBee systems, or indeed thousands of kilometres using a module for receiving GPS data. The DVD contains technical documentation (spec. sheets, application notes, user guides, etc.) on various devices according to the frequency and/or protocol used. All of the documents are PDF files (in English).

ISBN 978-90-5381-268-6
 £28.50 • US \$46.00



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

This DVD-ROM contains the full range of 1990-1999 volumes (all 110 issues) of Elektor Electronics magazine (PDF). The more than 2,100 separate articles have been classified chronologically by their dates of publication (month/year), but are also listed alphabetically by topic. A comprehensive index enables you to search the entire DVD.

ISBN 978-0-905705-76-7
 £69.00 • US \$111.30

January 2012 (No. 421) £ US \$
 + + + Product Shortlist January: See www.elektor.com + + +

December 2011 (No. 420)

Here comes the Bus! (10)

110258-1	Experimental Node board.....	5.30.....	8.60
110258-1C3	3 pcs Experimental Node board	11.50.....	18.60
110258-91	USB/RS485 Converter, ready made module	22.20.....	35.90

USB Data Logger

110409-1	Printed circuit board.....	9.75.....	15.70
110409-41	Programmed controller PIC24FJ64GB002-I/sp dil-28s.....	13.30.....	21.40

November 2011 (No. 419)

Improved Radiation Meter

110538-41	Programmed controller ATmega88PA-PU.....	9.35.....	15.10
110538-71	Kit of parts incl. display and programmed controller	35.50.....	57.30

Simple Bat Detector

110550-1	PCB, bare	8.85.....	14.30
----------------	-----------------	-----------	-------

OnCE/JTAG Interface

110534-91	Programmer board, assembled and tested	35.60.....	57.30
-----------------	--	------------	-------

Here comes the Bus! (9)

110258-1	Experimental Node board.....	5.30.....	8.60
110258-1C3	Printed circuit board 3x print Experimental Node	11.50.....	18.60
110258-91	USB/RS485 Converter, ready made module	22.20.....	35.90

Dual Linear PSU for Model Aircraft

081064-1	Printed circuit board.....	14.50.....	23.80
----------------	----------------------------	------------	-------

October 2011 (No. 418)

Versatile Board for AVR Microcontroller Circuits

100892-1	Printed circuit board.....	11.55.....	18.70
----------------	----------------------------	------------	-------

Audio DSP Course (4)

110001-91	PCB, populated and tested DSP board.....	115.70.....	186.70
110001-92	Bundle DSP board (110001-92) with Programmer (110534-91)	133.50.....	215.00

Here comes the Bus! (8)

110258-1	Experimental Node board.....	5.30.....	8.60
110258-1C3	Printed circuit board Experimental Nodes (3 PCBs).....	11.50.....	18.60
110258-91	USB/RS485 Converter, ready made module	22.20.....	35.90

September 2011 (No. 417)

eC-Reflow-Mate

100447-91	Professional SMT reflow oven	2170.00.....	3495.00
-----------------	------------------------------------	--------------	---------

USB Long-Term Weather Logger

100888-1	Printed circuit board.....	16.00.....	25.90
100888-41	Programmed controller ATMEGA88-20PU	8.85.....	14.30
100888-71	HH10D humidity sensor module	7.10.....	11.50
100888-72	HP03SA air pressure sensor module	5.75.....	9.30
100888-73	Kit of parts incl. PCB, controller, humidity sensor and air pressure sensor modules	31.10.....	50.20

I²C Sensors

100888-71	HH10D humidity sensor module	7.10.....	11.50
100888-72	HP03SA air pressure sensor module	5.75.....	9.30

E-Blocks go Twitter

EB003	E-blocks Sensor board	21.60.....	34.90
EB005	E-blocks LCD board.....	24.00.....	38.80
EB006	E-blocks PIC Multiprogrammer.....	72.00.....	116.20
EB007	E-blocks Switch board	14.40.....	23.30
EB059	E-blocks Servo board.....	14.40.....	23.30
EB069	E-blocks Wireless LAN board.....	132.00.....	212.90
TEDSS14	Flowcode 4 for dsPIC/PIC24	178.80.....	288.40

FT232R USB/Serial Bridge/BOB

110553-91	PCB, assembled and tested.....	12.90.....	20.90
-----------------	--------------------------------	------------	-------

Here Comes the Bus! (7)

110258-1	Experimental Node board, bare	5.30.....	8.60
110258-1C3	3 x Experimental Node board, bare	11.50.....	18.60
110258-91	USB/RS485 Converter, ready made module	22.20.....	35.90

Bestsellers

Books	1	311 Circuits ISBN 978-1-907920-08-0 £29.50US \$47.60
	2	Controller Area Network Projects ISBN 978-1-907920-04-2 £29.50US \$47.60
	3	Design your own PC Voice Control System ISBN 978-1-907920-07-3 £29.50US \$47.60
	4	Linux – PC-based measurement electronics ISBN 978-1-907920-03-5 £29.50US \$47.60
	5	Mastering the I²C Bus ISBN 978-0-905705-98-9 £29.50US \$47.60
CD/DVD-ROMS	1	CD 1001 Circuits ISBN 978-1-907920-06-6 £34.50US \$55.70
	2	DVD Elektor 1990 through 1999 ISBN 978-0-905705-76-7 £69.00US \$111.30
	3	CD Elektor's Components Database 6 ISBN 978-90-5381-258-7 £24.90US \$40.20
	4	DVD Elektor 2010 ISBN 978-90-5381-267-9 £23.50US \$37.90
	5	CD ATM18 Collection ISBN 978-0-905705-92-7 £24.50US \$39.60
Kits & Modules	1	Improved Radiation Meter Art. # 110538-71
	2	FT232R USB/Serial Bridge/BOB Art. # 110553-91
	3	USB Long-Term Weather Logger Art. # 100888-73
	4	Audio DSP Board + Programmer Art. # 110001-92
	5	Pico C Meter Art. # 100823-71

Order quickly and securely through
www.elektor.com/shop
 or use the Order Form near the end
 of the magazine!

Elektor
 Regus Brentford
 1000 Great West Road
 Brentford TW8 9HH • United Kingdom
 Tel. +44 20 8261 4509
 Fax +44 20 8261 4447
 Email: order@elektor.com



Sky Light Meter

This project involves a sensor array with a dedicated data recording system, designed to measure sky light intensity in five directions. Although the device was originally designed to record celestial light at the time of eclipses, it should have wider application areas since temperature and humidity values are also measured. We used programmable light sensors type TSL230 cleverly controlled by a PIC controller type 18F4455.



Enhanced Pico-C Meter

Our Pico-C-Meter originally published in the April 2011 edition is a very handy and compact instrument capable of quickly and accurate capacitance measurements up to 2 nF with a resolution of 0.1 pF. Thanks to rewritten software it's now possible to properly extend the range at both sides, now boasting 0.01 pF to 500 nF. Moreover, the circuit can double as a signal generator. With some modifications, it's even possible to add an input for period and frequency measurements. For the latter version a new PCB layout was designed.



Lambda Probe RS232 Interfacing

The interface for broadband lambda probes described in this article enables accurate measurement of the oxygen content of combustion gases, all without any kind of adjustment. By way of an RS232 connection to the interface board, all the diagnostic capabilities of the CJ125 sensor are used, while all data and operating status information of the circuit can be read in detail. Next month we describe the RS232 commands that allow the various settings to be made.

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

Elektor UK/European February 2012 edition: on sale January 19, 2011.

Elektor USA February 2012 edition: published January 16, 2011.

www.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com

Elektor on the web

All magazine articles back to volume 2000 are available individually in pdf format against e-credits. Article summaries and component lists (if applicable) can be instantly viewed to help you positively identify an article. Article related items and resources are also shown, including software downloads, hyperlinks, circuit boards, programmed ICs and corrections and updates if applicable.

In the Elektor Shop you'll find all other products sold by the publishers, like CD-ROMs, DVDs, kits, modules, equipment, tools and books. A powerful search function allows you to search for items and references across the entire website.

Also on the Elektor website:

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



ORDERING INSTRUCTIONS, P&P CHARGES

All orders, except for subscriptions (for which see below), must be sent BY POST or FAX to our Brentford address using the Order Form overleaf. Online ordering: www.elektor.com/shop

Readers in the USA and Canada should send orders, except for subscriptions (for which see below), to the USA address given on the order form. Please apply to Elektor US for applicable P&P charges. Please allow 4-6 weeks for delivery.

Orders placed on our Brentford office must include P&P charges (Priority or Standard) as follows: Europe: £6.00 (Standard) or £7.00 (Priority) Outside Europe: £9.00 (Standard) or £11.00 (Priority)

HOW TO PAY

All orders must be accompanied by the full payment, including postage and packing charges as stated above or advised by Customer Services staff.

Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London. IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP).

Please ensure your full name and address gets communicated to us.

Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.

GCredit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in this magazine. If difficulties in the supply of components are envisaged, a source will normally be advised in the article. Note, however, that the source(s) given is (are) not exclusive.

TERMS OF BUSINESS

Delivery Although every effort will be made to dispatch your order within 2-3 weeks from receipt of your instructions, we can not guarantee this time scale for all orders. **Returns** Faulty goods or goods sent in error may be returned for replacement or refund, but not before obtaining our consent. All goods returned should be packed securely in a padded bag or box, enclosing a covering letter stating the dispatch note number. If the goods are returned because of a mistake on our part, we will refund the return postage. **Damaged goods** Claims for damaged goods must be received at our Brentford office within 10-days (UK); 14-days (Europe) or 21-days (all other countries). **Cancelled orders** All cancelled orders will be subject to a 10% handling charge with a minimum charge of £5.00. **Patents** Patent protection may exist in respect of circuits, devices, components, and so on, described in our books and magazines. Elektor does not accept responsibility or liability for failing to identify such patent or other protection. **Copyright** All drawings, 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 copyright and may not be reproduced or transmitted in any form or by any means, including photocopying and recording, in whole or in part, without the prior permission of Elektor in writing. Such written permission must also be obtained before any part of these publications is stored in a retrieval system of any nature. Notwithstanding the above, 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 connexion 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 in respect of the goods. **Law** Any question relating to the supply of goods and services by Elektor shall be determined in all respects by the laws of England.

January 2012

SUBSCRIPTION RATES FOR ANNUAL SUBSCRIPTION

	Standard	Plus
United Kingdom & Ireland	£54.00	£66.50
Surface Mail		
Rest of the World	£68.50	£81.00
Airmail		
Rest of the World	£86.00	£98.50
USA & Canada	See www.elektor.com/usa for special offers	

HOW TO PAY

Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London. IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP). Please ensure your full name and address gets communicated to us.

Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.

Credit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

SUBSCRIPTION CONDITIONS

The standard subscription order period is twelve months. If a permanent change of address during the subscription period means that copies have to be despatched by a more expensive service, no extra charge will be made. Conversely, no refund will be made, nor expiry date extended, if a change of address allows the use of a cheaper service.

Student applications, which qualify for a 20% (twenty per cent) reduction in current rates, must be supported by evidence of studentship signed by the head of the college, school or university faculty. A standard Student Subscription costs £43.20, a Student Subscription-Plus costs £55.70 (UK only).

Please note that new subscriptions take about four weeks from receipt of order to become effective. Cancelled subscriptions will be subject to a charge of 25% (twenty-five per cent) of the full subscription price or £7.50, whichever is the higher, plus the cost of any issues already dispatched. Subscriptions cannot be cancelled after they have run for six months or more.

January 2012

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

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 | € 4.99!

Further information at

www.elektor.com/app

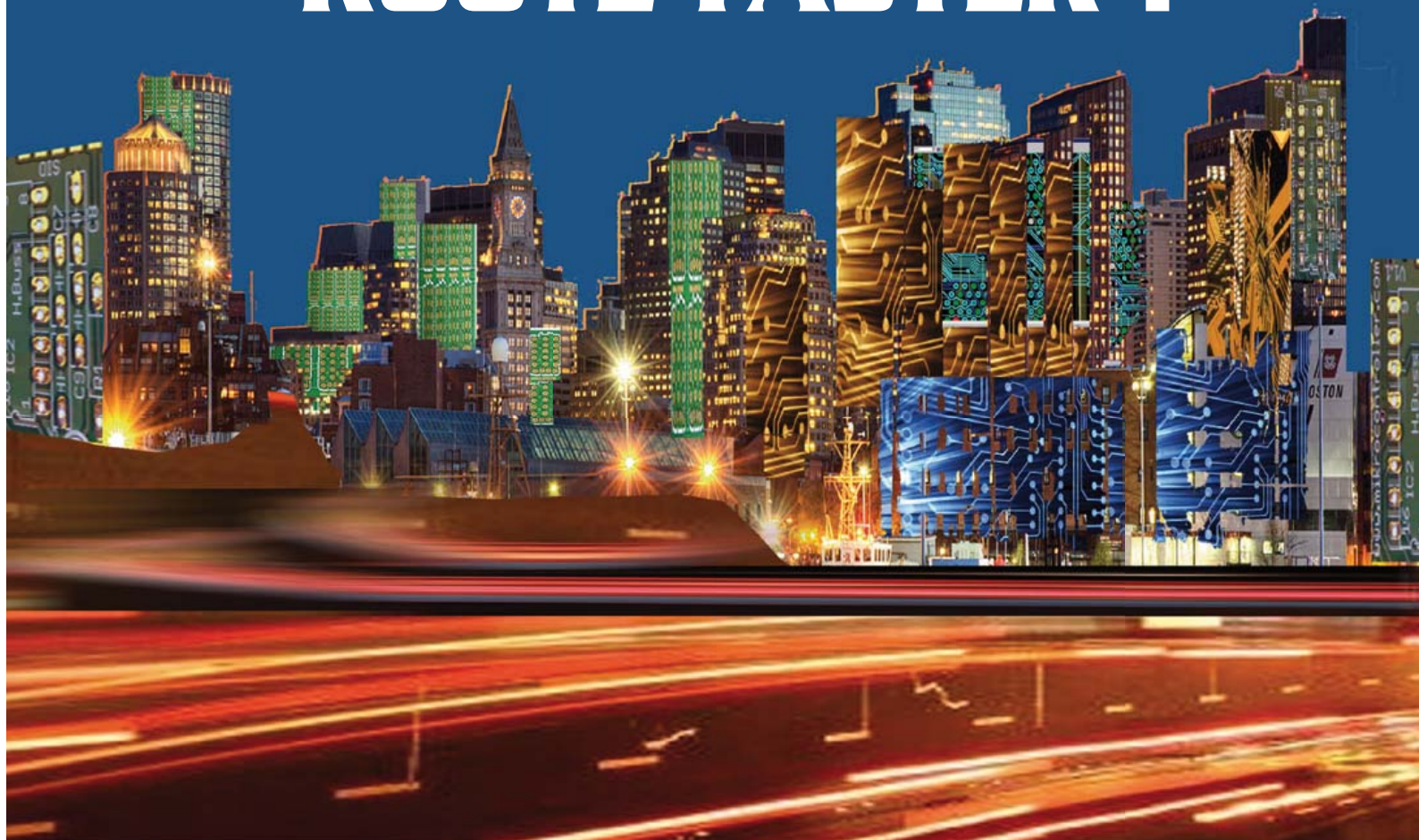
INDEX OF ADVERTISERS

AudioXpress	www.cc-webshop.com	42	Labcenter	www.labcenter.com	88
BAEC, Showcase	http://baec.tripod.com/	78	Maxbotix, Showcase	www.maxbotix.com	79
Beta Layout	www.pcb-pool.com	23, 78	MikroElektronika	www.mikro.com	3
CS Technology, Showcase	www.cstech.co.uk/picdemo	78	Minty Geek, Showcase	www.mintygeek.com	78
DesignSpark chipKIT™ Challenge	www.chipkitchallenge.com	2	Pico Technology	www.picotech.com	19
Easysync, Showcase	www.easysync-ltd.com	78	Quasar Electronics	www.quasarelectronics.com	15
Eltec, Showcase	www.eltec.com	78	Robot Electronics, Showcase	www.robot-electronics.co.uk	79
Eurocircuits	www.eurocircuits.com	69	Robotiq, Showcase	www.robotiq.co.uk	79
EzPCB/Beijing Draco Electronics Ltd.	www.v-module.com	23	Showcase	78, 79	
First Technology Transfer Ltd, Showcase ...	www.ftt.co.uk	78	Toroidy	www.toroidy.pl	23
FlexiPanel Ltd, Showcase	www.flexipanel.com	78	Tyder, Showcase	www.tyder.com	79
Future Technology Devices, Showcase	www.ftdichip.com	78			
Hameg, Showcase	www.hameg.com	78			
HexWax Ltd, Showcase	www.hexwax.com	78			
Jackaltac	www.jackaltac.com	11			

Advertising space for the issue 21 February 2012 may be reserved not later than 24 January 2012

with Elektor International Media - Allee 1, 6141 AV Limbricht, the Netherlands
 Telephone 0031 (0) 46 4389444 - Fax 0031 (0) 46 4370161 -
 e-mail: advertenties@elektor.com to whom all correspondence,
 copy instructions and artwork should be addressed.

ROUTE FASTER !



WITH PROTEUS PCB DESIGN

Our completely new manual router makes placing tracks quick and intuitive. During track placement the route will follow the mouse wherever possible and will intelligently move around obstacles while obeying the design rules.

All versions of Proteus also include an integrated world class shape based auto-router as standard.

PROTEUS DESIGN SUITE **Features:**

- Hardware Accelerated Performance.
- Unique Thru-View™ Board Transparency.
- Over 35k Schematic & PCB library parts.
- Integrated Shape Based Auto-router.
- Flexible Design Rule Management.
- Polygonal and Split Power Plane Support.
- Board Autoplacement & Gateswap Optimiser.
- Direct CAD/CAM, ODB++, IDF & PDF Output.
- Integrated 3D Viewer with 3DS and DXF export.
- Mixed Mode SPICE Simulation Engine.
- Co-Simulation of PIC, AVR, 8051 and ARM7.
- Direct Technical Support at no additional cost.

Prices start from just £150 exc. VAT & delivery

labcenter  www.labcenter.com
Electronics

Visit our website or
phone 01756 753440
for more details

Labcenter Electronics Ltd. 53-55 Main Street, Grassington, North Yorks. BD23 5AA.
Registered in England 4692454 Tel: +44 (0)1756 753440, Email: info@labcenter.com