www.elektor.com May 2009 AUS$ 12.90 - NZ$ 15.50 - SAR 88.70 £ 4.10

elektor

electronics worldwide

... R32C/111 GOES OLED ... XMEGA REVEALED ... TI83+/84+ TURNS CONTROL I/F ... V & I CALIBRATOR ...

USB 3.0 SuperSpeed

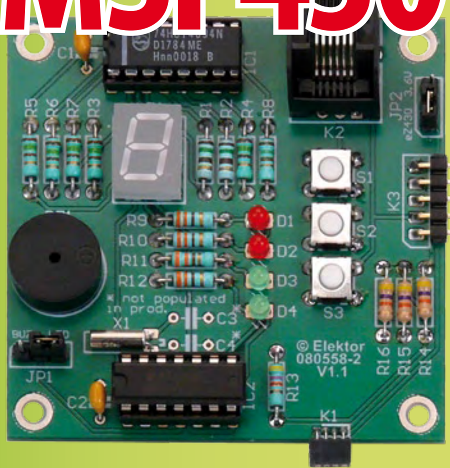
What to expect?

USB Oscilloscopes

Pico and Velleman compared

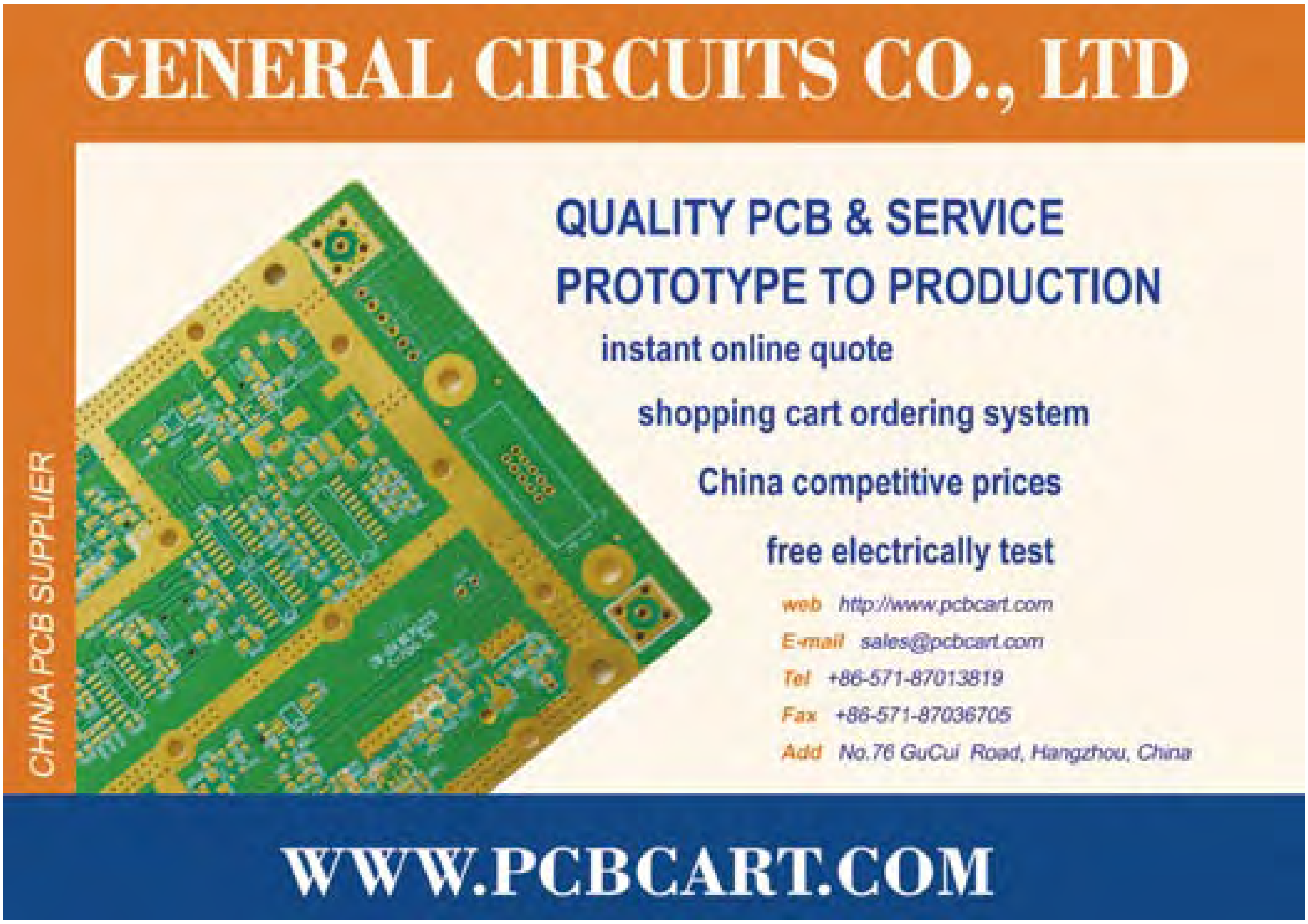
MSP430 all the way

Embedded C course and development board



R21 05
9 770268 45114 2


We add value to PCBs
when others just sell it.
Design
Prototype
Production
- One Stop Manufacturing Service
- Free Electronics Components
- Free Prototyping Assembly
- Professional Consultant
Designing Service
3D Enclosure Designing
Virtual Assembly
PCB Design
Component
Fpc
Quick Prototype
Assembly
Pcb
Keypad
Pcbs
Assembly
Quick Prototype
New: Online Components Store
Components
Fpcs
Keypads
www.EzPCB.com
Email: sales@ezpcb.com


GENERAL CIRCUITS CO., LTD
CHINA PCB SUPPLIER
QUALITY PCB & SERVICE
PROTOTYPE TO PRODUCTION
instant online quote
shopping cart ordering system
China competitive prices
free electrically test
web http://www.pcbcart.com
E-mail sales@pcbcart.com
Tel +86-571-87013819
Fax +86-571-87036705
Add No.76 GuCui Road, Hangzhou, China
WWW.PCBCART.COM

Courseware on course

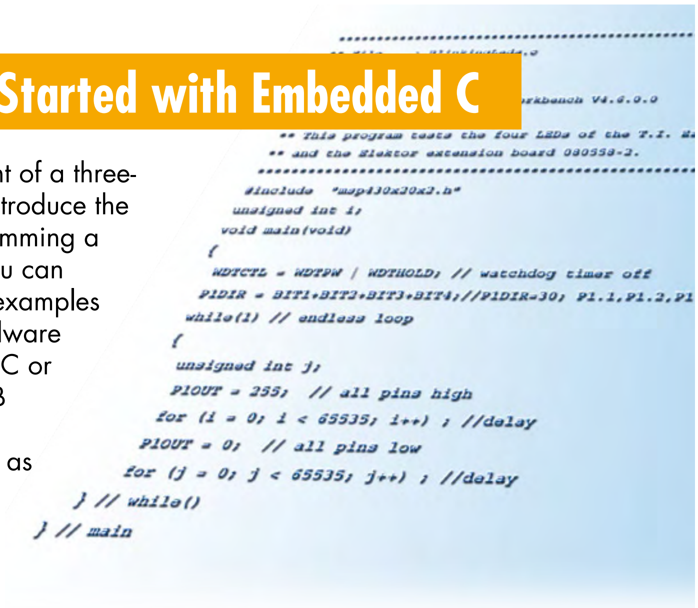
Bascom AVR (6), *FPGA* (9), *Microcontroller Basics* (5) and *Basic Stamp* (7) are just a few courses published in Elektor over the past few years. The figures in brackets are the number of instalments eventually carried in the magazine. True, in some cases far fewer instalments were planned and the author(s) and editors simply got carried away. Those of you with a memory longer than our website (i.e. pre-2000) may recall equally winning courses like *Figuring it Out, 8051 Assembly Language* and *Neural Networks*.
With hindsight the relative success of most of Elektor's courses is due to close interaction between the courseware elements: what's on paper, the hardware supplied, the (free) software, didactics and support from the tutoring party. Good interaction is a condition for reader involvement and the lot either 'taking off' or sinking into oblivion after two months or so. Only the very best of courses make it to the book, CD or 'product bundle' level.
Although there is no shortage of books and online material on the C programming language, much of this is general-purpose at best, with a division between PC programming on the one hand and 'embedded' programming on the other. Still, C being very much a 'broadband' language — also for embedded applications — book authors for obvious reasons may not want to limit themselves to coverage of a specific processor.
However, for a monthly journal like Elektor the strength is exactly there as it is better geared to acute focusing in the field. A good example is the pair of MSP430 articles in this month's issue. In good Elektor tradition, one article is the hardware show (page 18) while the other (page 22) kicks off a short course on C specifically for Texas Instrument's best known 16-bit RISC micro at the electronics enthusiasts' level. As I was able to witness on several Embedded Systems Conference exhibitions, the MSP430 has a huge following particularly in the student area and TI deserves credit for not having lost the connection with the embedded community, which has strong tendency to disappear underground and into rucksacks for software (IP) and hardware respectively. The only disadvantage of TI's student-aimed eZ430 stick is a lack of connectivity so Elektor teamed up with two automotive electronics teachers to churn out an MSP430 development system and a matching *Embedded C* course we hope you will actively participate in.

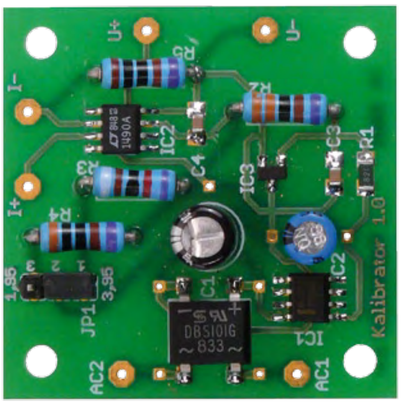
Jan Buiting
Editor

18 Experimenting with

22 Getting Started with Embedded C

This is the first instalment of a three-part series which will introduce the fundamentals of programming a microcontroller in C. You can immediately try all the examples using the MSP430 hardware in combination with a PC or laptop which has a USB interface. The software we've used is available as a free download.





48 V & I Calibrator

It's difficult to be sure that your digital multimeter (DMM) is taking accurate measurements especially if it's a few years old. This handy calibrator gives full scale reference levels of both voltage and current, designed specifically for the scale ranges used by DMMs.

CONTENTS

Volume 35
May 2009
no. 389

the MSP430

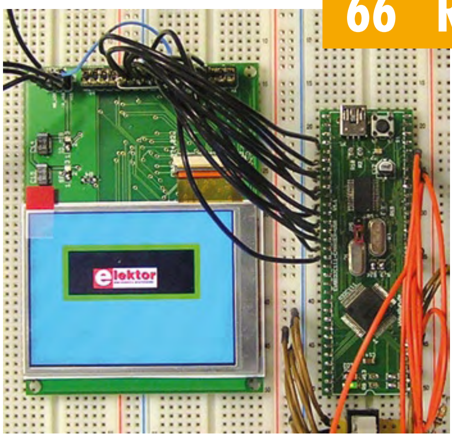
The I/O facilities of TI's USB evaluation sticks for its low-cost MSP430 controllers being limited it's a good idea to design a dedicated experimenter's board. The board and the stick form the hardware basis for an *Embedded C course* also found in this issue.

60 PC-Based Instruments

In this article we examine a pair of two-channel units that also include a built-in function generator:

the PicoScope 2203 and the Velleman PCSGU250.

66 R32C/111 goes OLED



With OLEDs it's not all plain sailing since driving them by microcontroller presents developers with a number of challenges. Continuing our series on the Renesas R32C, we trawl the theory to come up with a highly practical solution using the R32C carrier board.

projects

technology

info & market

infotainment

elektor international media

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.

Volume 35, Number 389, May 2009 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: Antoine Authier (Head), Ton Giesberts, Luc Lemmens, Daniel Rodrigues, Jan Visser, Christian Vossen

Editorial secretariat: Hedwig Hennekens (secretariaat@elektor.nl)

Graphic design / DTP: Giel Dols, Mart Schroijen

Managing Director / Publisher: Paul Snakkers

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

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: Huson International Media, Cambridge House, Gogmore Lane, Chertsey, Surrey KT16 9AP, England.
Telephone: +44 1932 564999, Fax: +44 1932 564998
Email: p.brady@husonmedia.com
Internet: www.husonmedia.com
Advertising rates and terms available on request.

C++

Dear Editor — I have created an engine for displaying 3D graphics with Direct3D9 and C++ Builder. The engine source code is free and all classes described are documented at: http://bcbjournal.org. The url for the engine is: www.gtokas.com/index.php?q=TCDX9en. If you think it worth writing an article about 3D graphics rendering along with technical documentation on how all those works, please respond.

George Tokas (Greece)

We'll happily evaluate your article proposal, George.

Dear Elektor — just writing in to say we have a completely free online Science and Engineering Encyclopedia http://www.diracdelta.co.uk and have a list of electronics topics at www.diracdelta.co.uk/science/source/e/l/electronics/source.html.
A great resource for professional and students alike.

Charlie Hawkins (United Kingdom)

Thanks for the tip Charlie!

What CAD?

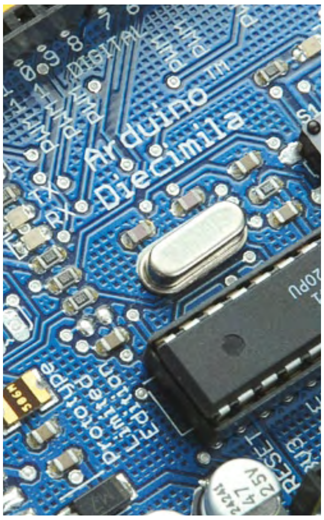
Hi Jan — does Elektor use Eagle for its schematic capture and PCB layout work? If so, in addition to PCB artwork, can Elektor also provide Eagle libraries for the modules it creates? I'm particularly interested in the ATM18 microcontroller module and want to use it in some of my own home-brew designs.

David Bannister (Singapore)

Elektor labs employ Altium Designer for their engineering-level schematics, BOMs and PCB designs. Altium is also used for most circuit diagrams you can see in the magazine. We also use McCAD for printed schematics and the odd block diagram. For both programs, Elektor employs large component shape libraries compiled and refined over many years, hence our house style which has remained basically unchanged since the mid 1970s. These libraries are proprietary.
Our published circuits are copyrighted and intended for educational and private use only, see the Copyright Notice on page 7. Commercial use is subject to written approval from the Publisher.

Arduino and Bascom

Dear Editor — the March issue had an article on Arduino. I agree with you that the programming environment supplied with Arduino is easy to learn. However, sometimes



you want to use programs from another source or use Basic (such as Bascom). In this case, Arduino is still an inexpensive and robust ready-to-use microcontroller module with a USB port.
The following website provides some information on using Arduino in combination with Bascom:
http://med.hro.nl/kemjt/Send_Eng.htm

Jos van Kempe (The Netherlands)

PS: I used Arduino and Bascom for a textbook on control engineering at the technical college level.

Thanks for this information, which certainly deserves publication in our Mailbox.

An Eye for Distance

Dear Editor — I noticed an inconsistency in this article (Optical triangulation with the ATM18, Elektor February 2009, *Ed.*).
The voltage divider for the output signal is described at one point in the text as consisting of 4.7 kΩ and 5.6 kΩ resistors, which corresponds to the colour-code markings of the resistors in Figure 6. However, the voltage divider is twice described as consisting of 4.7 kΩ and 6.8 kΩ resistors, and these values are shown in the schematic diagram in Figure 7. To arrive at the reference voltage of the ATMega (1.1 V) from the output level of 2.4 V (or 2.7 V), the value would have to be 5.6 kΩ.

Michael Kaiser (Germany)

The correct values for the voltage divider are 5.6 kΩ (R2) and 4.7 kΩ (R1).

Burkhard Kainka

Hard-disk substrates

Dear Jan — the current edition of i-Trixx (week 2/2009, *Ed.*) includes a fun construction project using platters from a discarded SCSI disk drive. In an aside, the author mentions that aluminium discs are used for data storage media.
I have occasionally dismantled used disk drives in the past, but they did not always have aluminium platters. Some of them broke like glass when I tried to bend them. Is more detailed information available on the materials used to make disk platters?

Carsten Bohemann (Germany)

Your observations are correct and your bending experiments are very consistent. Your supposition is also correct: the platters of hard-disk drives sometimes break like glass for the simple reason that they are in fact made from glass. Disk platters are made either from very stiff, lightweight aluminium alloys or from glass, which has the advantage of no eddy currents. You can clearly identify the actual platter material by using an inexpensive metal detector of the type used to locate electrical wiring (available in DIY building shops) or by using an ohmmeter to test the conductivity of the rim of the platter.

Dr Thomas Scherer

Lead-free soldering

Dear Elektor — I recently bought two project kits from the Elektor Shop (for the ATM18 project and the Four-channel Logic Analyser). Now I wonder whether these kits conform to the RoHS regulations. Do I also have to use lead-free solder for assembly? I have already purchased new solder (Sn95.5Ag3.8Cu0.7) – can I use this? What is the best soldering temperature?

Martin Baumberger (Germany)

These kits, like all current Elektor kits and modules, conform to the RoHS regulations. As these kits are intended for private users instead of commercial equipment production, RoHS compliance is actually not mandatory. As a non-com-

mercial producer (private user), you can and may use solder containing lead (such as SnPb 60/40) for all kits and modules, and such solder is still widely available commercially.

If you wish to use Sn95.5Ag3.8Cu0.7 solder in your projects, you should bear in mind that it has a melting temperature of 217 °C, which is 34 °C higher than the melting temperature of tin/lead solder (SnPb 60/40). In practice, this somewhat higher melting temperature is hardly noticeable with the usual sorts of soldering irons for electronics assembly, which typically have a soldering-tip temperature of 350 °C. However, the temperature should not exceed 380 °C. A temperature of around 350 °C is usually adequate. One thing that takes getting used to is that the solder joints are not shiny as with lead solder, but instead turn dull as soon as the solder cools and hardens. You can solder as nicely as you please, but the results always look like 'cold' solder joints.

Elektor published an extensive article on lead-free soldering in the May 2000 (!) issue, with additional articles in the June 2005 and May 2006 issues.

Footprints in Eagle

I'm presently working on a PCB design in Eagle, using the standard version (5.2.0). No matter what I try, I can't manage to edit the footprints of my ICs and passive components.

I would like to use somewhat larger pads (the connection points for the components), since I wouldn't be able to do anything with the finished PCB because the pads are much too small.

Can someone tell me how to change my footprints when I am designing a PCB?

Steven33 (Elektor Forum user)

A component is called a 'device' in Eagle, and every device consists of a symbol and a package. The properties of the footprint are specified in the package. This means that if you want to modify the pads, you have to do so in the package. The procedure is not always especially intuitive, and you will have to consult the user guide more than once, but it is in fact described in the user guide.

Here we can describe the procedure with a brief example. Suppose you need a common garden-variety resistor, and you select 0309/12 from the RCL library. The size and lead spacing are OK, but the pads are much too small. To change this, use the menu bar to open the library: Library -> Open -> rcl.lbr.

If you select the symbol icon for R-EU_, you will see a resistor symbol with a long list of possible packages. Here you have to look for 0309/12 and then edit it. Here it's a good idea to first make a copy and then edit the copy. To do this, type the following command line at the top of the window: copy 0309/12.pac@rcl.lbr 0309/12s. The package will be displayed after this, and you can edit it directly.

First enter your desired settings for the size, shape and drilled hole diameter of the new pad. Then remove the old pads and place the new ones. Change the names of the pads from P$1 and P$2 to P1 and P2, and then click 'Save All' to save your changes.

Back at the device, click the 'New' button at the bottom left in the window. You will see a list of all the packages in this library. Find '0309/12s' in the list and click it. The new package will be shown at the top right. Then click the 'Connect' button. This takes you to a window where you can link the pin numbers of the symbol to the pads of the package. In this case, click the large 'Connect' button twice.

The new package is now present in the list, but a quotation mark is shown in the 'Variant' column. Right-click this, select 'Rename', and enter a name of your choice.

Finally, update the library and (as a check) select Library -> Use -> rcl.lbr once again. The resistor with the new footprint should be available now.

Good luck!

petrus bitbyter (Elektor Forum user)

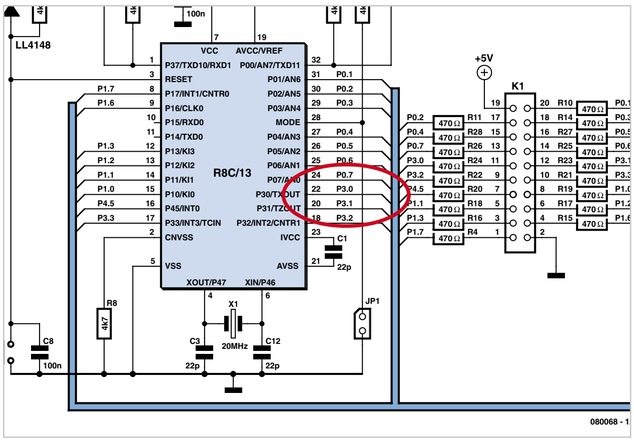
Corrections & Updates

Transistor Curve Tracer

February 2009, p. 24-31, no. 080068-I

In the circuit diagram (Figure 2, section a), the bussed connection between pin 22 (P3.0) of the R8C/13 module and resistor R24 is missing. This connection is however present on the circuit board, for which no modification is required.

In the component list, transistor T2 should be a type BC557A, not BC547A. No modification is required to the PCB or the schematic.

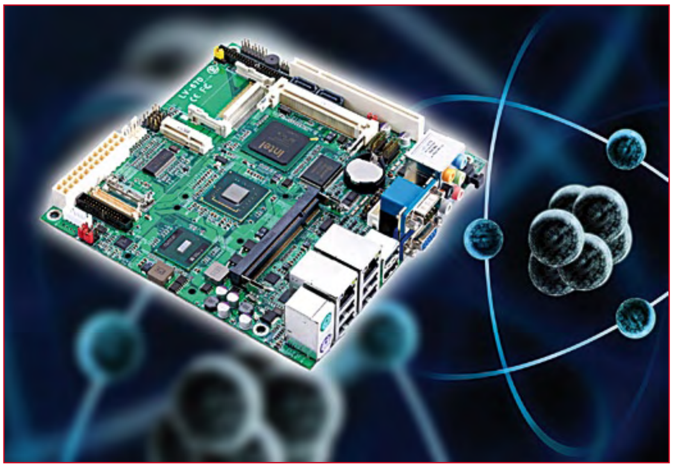


MailBox Terms

- Publication of reader's orrespondence is at the discretion of the Editor.
- Viewpoints expressed by correspondents are not necessarily those of the Editor or Publisher.
- Correspondence may be translated or edited for length, clarity and style.
- When replying to Mailbox correspondence, please quote Issue number.
- Please send your MailBox correspondence to:

editor@elektor.com or Elektor, The Editor, 1000 Great West Road, Brentford TW8 9HH, England.

LV-67D mini-ITX motherboard based on Atom N270



BVM has added the LV-67D to its extensive family of mini-ITX form factor motherboards for embedded applications. The LV-67D uses the Intel 945GSE chipset and incorporates the 45nm Intel Atom N270 processor. The chip's power consumption is particularly low at 2.5 W, and with extensive I/O capability, the board is ideal for embedded applications such as digital signage, kiosks, point of sale terminals, thin clients, digital security, residential gateways and commercial and industrial control equipment where ultimate computing power is less important than power conservation. The onboard Intel GMA 950 32-bit 3D graphics engine offers LVDS, DVI, CRT or HDTV capability and multiple graphics displays.

The N270 runs at 1.6 GHz with a 533 MHz FSB addressing up to 2 GB of RAM. The LV-67D provides a comprehensive selection of I/O: two Giga LAN ports, 8× USB 2.0 ports, 5× RS232 and one RS485 serial port. 2× 150MB/s SATA interfaces give access to mass storage and the board has IDE support for a solid-state disk. Extended interface facilities include a PCI Express mini card socket, a Mini-PCI socket and a PCI slot. System management functions include an 8-bit GPIO programmable interface and a 256-level watchdog timer.

www.bvmltd.co.uk

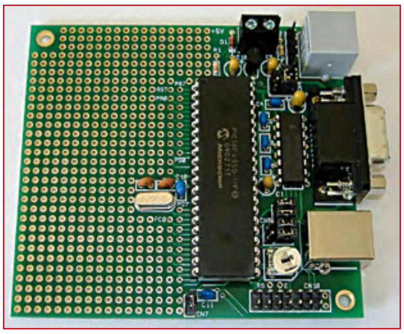
(090169-XI)

CO Gas Sensor Module

Parallax' CO Gas Sensor Module is designed to allow a microcontroller to determine when a preset CO gas level has been reached or exceeded. Interfacing with the sensor module is done through a 4-pin SIP header and requires two I/O pins from the host microcontroller. The sensor module is mainly intended to provide a means of comparing carbon monoxide sources and being able to set an alarm limit when the source becomes excessive.
The new module employs the MQ-7 CO gas sensor, has an easy SIP interface and is compatible with most microcontrollers. The module costs $29.99 plus shipping.

www.parallax.com (search '27931')

(090231-I)



Low-cost PIC18F4550-USB prototyping kit

C S Technology Ltd. have released a PIC prototyping board in kit form for 40 pin PIC microcontrollers, including the 18F4550 USB version.
The board includes a large prototyping area, RS232 and USB connectivity, a 5-pin programming header and Microchip ICD2 compatible connector, together with selectable on-board 5 V regulator and an LCD display connector.

This new kit adds to CST's range of 18- and 28-pin PIC proto kits, CTCSS and DTMF kits.
The complete kit of parts including PCB costs just £ 14.99 plus P&P. C S Technology also offer a PIC program development and prototyping service.

www.cstech.co.uk

(090231-IV)

Industry's highest density transceiver FPGAs

Altera's second member of the Stratix® IV GX FPGA family, the EP4SGX530 is 60 percent larger than the largest transceiver FPGA on the market. The device offers 530 K logic elements (LEs), up to 48 transceivers operating at up to 8.5 Gbps, 20.3 Mbits of RAM and 1,040 embedded multipliers. Stratix IV GX devices target numerous applications in the communications, broadcast, test, medical and military markets.
Stratix IV GX FPGAs incorporate up to four hard intellectual property (IP) cores for PCI Express Gen1 and Gen2 (x1, x4 and x8), and also support a wide range of protocols including Serial RapidIO®, 40G/100G Ethernet, XAUI, CPRI (including 6G CPRI), CEI-6G, GPON, SFI-5.1 and Interlaken.
The Stratix IV GX EP4SGX530 and EP4SGX230 devices are currently shipping, with other family members scheduled to ship in 2009.

www.altera.com/pr/stratix4

(090231-VII)

AVR32 digital audio gateway reference design

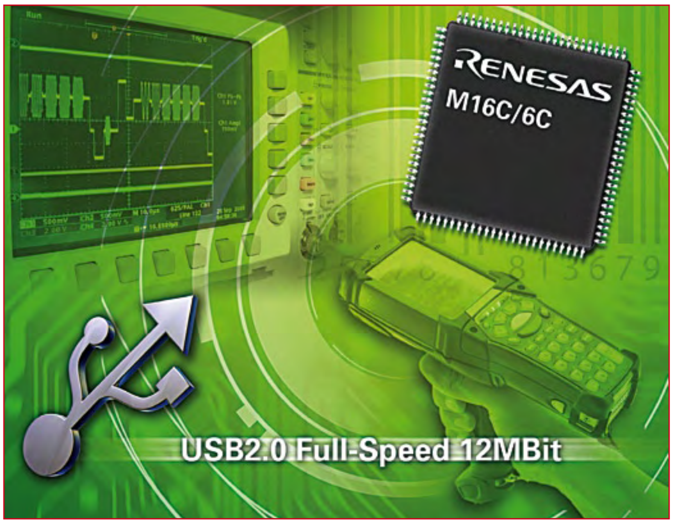
Atmel® Corporation recently announced the AVR®32 ATEVK1105 Digital Audio Gateway kit, demonstrating digital audio streaming, decoding and playback. These audio capabilities serve the exploding market of audio accessories and peripherals that connects home and car Hi-Fi audio systems to the digital age. This includes the popular iPod® docking stations.
The kit is based on the AVR high performance AT32UC3A 32-bit Flash microcontrollers and provides developers with a ready-to-use hardware/software platform and a variety of interfaces and evaluation capabilities to meet their audio systems requirements and get faster to market.
The fast AVR32 CPU featuring DSP instructions is perfect for audio decoding tasks, and the UC3A handles two audio interfaces. For high quality stereo output, the chip has a stereo 16-bit bitstream audio DAC with internal FIR and Comb filters. For 4-channel or full surround sound, an IIS interface is available for connection to an external audio codec. Both interfaces are supported with drivers that make full use of the AVR32 peripheral DMA controller which significantly reduces the CPU overhead.
The ATEVK1105 board comes preloaded with software that demonstrates audio playback. A booted kit will scan any USB mass storage device for MP3 or other audio files, and play them back. The kit's software will even scan the ID3 tag and present album artist and song information on the 2" QVGA onboard display. MP3 decoding from a USB mass storage device requires only a third of the AVR32's processing capacity, leaving plenty of headroom for running the operation system, streaming the data and refreshing the display.
Customers with a license from Apple® are able to interface the kit to a iPod or iPhone® using an authorized Apple authentication chip adapter.
For applications where the on-chip 512 KB Flash and 64 KB SRAM is not sufficient, the EVK1105 demonstrates how to connect an external SDRAM, serial DataFlash, SD card reader and USB hard disk or memory stick.
The ATEVK1105 also features connectors for future wireless expansion modules supporting IEE802.15.4™/Zigbee®, and Bluetooth®.
Atmel provides all the source code free of charge, including software drivers for all peripherals, TCP/IP stack and various USB class drivers. This is available in the AVR32 Software Framework, a software library integrated with the AVR32 Studio development suite. Atmel also releases the schematics and Gerber files to allow customers to easily incorporate elements of the kit into their own designs.
The ATEVK1105 Digital Audio Gateway kit will be available from Atmel's distributors in March 2009 with a resale price of US$179 plus shipping.

www.atmel.com/avr32

(090231-II)

USB Support for Renesas M16C/6C Microcontroller

Renesas Technology Europe, its Gold Alliance Partner Thesycon and MSC Vertriebs GmbH supply a complete USB stack to support the Renesas Starter Kit (RSK) RSKM16C/6C, based on the M16C/6C.
The M16C/6C is a part of the popular M16C microcontroller platform with added support for USB 2.0. The M16C/6C product group includes a total of 16 different models, and the new components are completely compatible with earlier versions. The new USB 2.0 compliant, full-speed 12 MB/s interface supports standard Control, Bulk and Interrupt transfer types. The components include the M16C/60 16-bit CISC CPU core that works with clock rates of up to 32 MHz and a power supply of between 2.7 and 5.5 VDC.



The software stack complies with the USB 2.0 specification and supports Control, Bulk and Interrupt transfer modes at maximum speed. It also includes complete USB request processing and expanded error recovery mechanisms for faultless operation. The USB stack firmware is written in ANSI-C and supports Renesas' High-Performance Embedded Workshop (HEW) development environment. To facilitate integration, the software is designed as a library and provided in source code form. The library does not require specific operating system support, enabling it to be integrated into any embedded operating system and used in standalone applications.
All products are available at the MSC webshop.

www.msc-toolguide.com/renesas

(090231-III)

Energy meter survives raging fire

George Municipality, in the Western Cape, South Africa, recently witnessed the power of Conlog's meter technology when one of the company's meters, the single phase BEC23PL/ T, survived a fire.

The meter was discovered when a customer contacted George Municipality to report that he was unable to read the remaining credit on the meter. When technicians visited the customer's home, they discovered that the meter had actually endured severe fire and survived! The meter was still completely functional despite its appearance and exposure to the exceedingly high temperatures. Remarkably, the meter technology and memory had not been affected, which means the consumer could still buy and enter electricity credit into his meter. In addition, the Municipality could also access meter information using the meter number as usual.

Conlog meters are usually tested to withstand a maximum of 960°C. In this case, an average house fire can reach over 1100°C.

www.conlog.co.za

(090231-V)

GPS industry's smallest standalone receiver

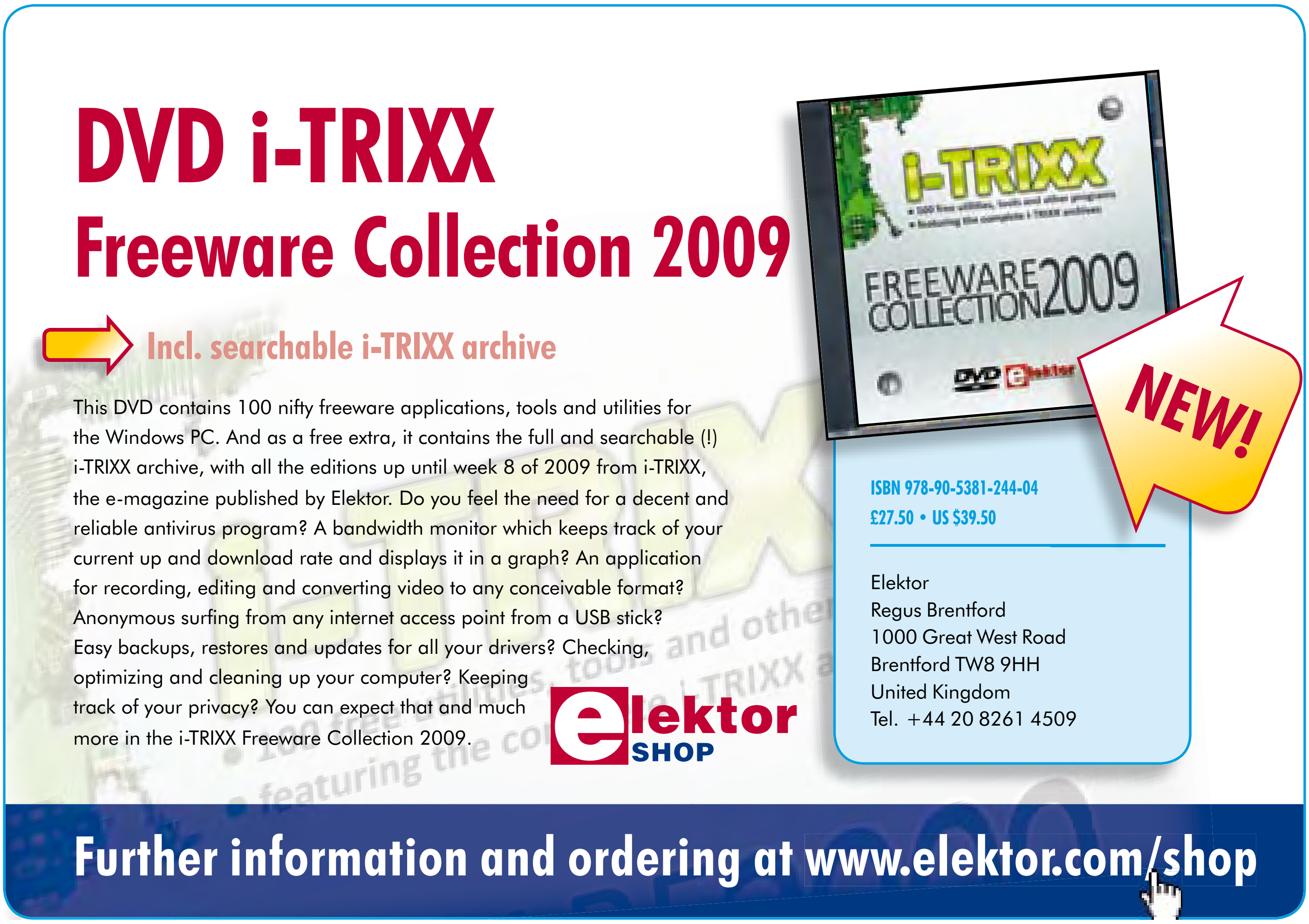
Alpha Micro Components has added u-blox' AMY-5M to its portfolio of GPS components. The new AMY-5M is the GPS industry's smallest standalone receiver currently available on the market. AMY-5M is miniature in size (6.5×8×1.2mm) and allows integration into the smallest portable devices. The fully tested ROM-based solution features the high performance 50-channel u-blox 5 positioning engine and has been developed for easy design implementation. The module also works as a standalone device which can operate at 1.8 V or 3 V, and does not require host integration or additional components to function.
Designed to withstand a temperature range of –40˚C up to +85˚C, the receiver integrates a standard crystal, which brings very fast acquisition and tracking performance at an economical price. Furthermore, AMY-5M handles u-blox' Assisted GPS (AssistNow-Online and AssistNow-Offline) providing even better startup and tracking performance under weak signal conditions. AMY-5M can be assembled on a 2-layer PCB, which leads to additional cost savings.

www.alphamicro.net/amy

(090231-X)


THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT
Specialising in Prototype PCBs
NEW! Free Laser Stencil on all Prototype PCB orders
NEW! 1 WD prototype service
NEW! Chemical Tin Finish (no extra cost)
Free Phone
UK 0800 389 8560
Simply send your layout files and order ONLINE
PCB-POOL.COM • sales@pcb-pool.com
Beta LAYOUT


PEAK electronic design ltd
tel. 01298 70012
fax. 01298 70046
www.peakelec.co.uk
sales@peakelec.co.uk
The famous Peak Atlas Range:
Atlas DCA
includes new premium probes!
£46.81 +VAT = £53.83
Atlas DCA Model DCA55
Semiconductor Analyser
Identifies type and pinout!
Atlas ESR
£75.74 +VAT = £87.10
Atlas ESR Model ESR60
ESR and Capacitance Meter
Resolution of 0.01 ohms!
Atlas LCR
£67.23 +VAT = £77.31
Atlas LCR Model LCR40
Inductor, Capacitor, Resistor Analyser
Automatic part identification.
Atlas SCR
includes new premium probes!
£84.26 +VAT = £96.90
Atlas SCR Model SCR100
Triac and Thyristor Analyser
Auto gate test current up to 100mA
UK: Please add £2 p&p to your order. Prices include UK VAT.
Please see website for overseas pricing.


DVD i-TRIXX
Freeware Collection 2009
Incl. searchable i-TRIXX archive
This DVD contains 100 nifty freeware applications, tools and utilities for the Windows PC. And as a free extra, it contains the full and searchable (!) i-TRIXX archive, with all the editions up until week 8 of 2009 from i-TRIXX, the e-magazine published by Elektor. Do you feel the need for a decent and reliable antivirus program? A bandwidth monitor which keeps track of your current up and download rate and displays it in a graph? An application for recording, editing and converting video to any conceivable format? Anonymous surfing from any internet access point from a USB stick? Easy backups, restores and updates for all your drivers? Checking, optimizing and cleaning up your computer? Keeping track of your privacy? You can expect that and much more in the i-TRIXX Freeware Collection 2009.
i-TRIXX
FREEWARE COLLECTION 2009
NEW!
ISBN 978-90-5381-244-04
£27.50 • US $39.50
Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom
Tel. +44 20 8261 4509
elektor SHOP
Further information and ordering at www.elektor.com/shop

PICDEM™ Lab Development Kit

Microchip announces the PICDEM™ Lab Development Kit, a comprehensive entry-level development platform for all of Microchip's 8-bit Flash PIC® microcontrollers (MCUs) with 20 or fewer pins. Aimed at educators, students and newcomers to microcontrollers, the PICDEM Lab Development Kit comes complete with five popular 8-bit PIC MCUs, along with a selection of discrete components, a PICkit™ 2 Debugger/Programmer and a CD containing a User's Guide, labs and application examples. The kit provides everything Microchip announces the PICDEM™ Lab Development Kit, a comprehensive entry-level development platform for all of Microchip's 8-bit Flash PIC® microcontrollers (MCUs) with 20 or fewer pins. Aimed at educators, students and newcomers to microcontrollers, the PICDEM Lab Development Kit comes complete with five popular 8-bit PIC MCUs, along with a selection of discrete components, a PICkit™ 2 Debugger/Programmer and a CD containing a User's Guide, labs and application examples. The kit provides everything needed to quickly and easily develop applications using Microchip's 8-bit PIC Microcontrollers.

A solderless prototyping area on the PICDEM Lab development board allows users to explore a number of application examples described in the 'hands-on' labs from the PICDEM Lab User's Guide that comes with the kit. The easy-to-follow labs provide an intuitive introduction to common peripherals and then move into a variety of application examples to reinforce core concepts. All of the code examples are written in the high level programming language, C, and can be compiled using the HI TECH C compiler, available as a free download from www.microchip.com/HI-TECH.

The PICDEM Lab comes complete with the following:

- PICDEM Lab Development Board, with samples of five 8-bit PIC Microcontrollers
- Component Kit
- PICkit™ 2 Debugger/Programmer
- CD containing comprehensive user's guide, labs and application examples.
- Free HI-TECH C compiler

The PICDEM Lab Development Kit (# DM163035) is available today for $124.99 (plus shipping) from microchipdirect.

www.microchipdirect.com
www.microchip.com/picdemlab

(090231-IX)

RS lowers prices on over 45,000 electronics components

With reduction levels as high as 50% on some products, RS have placed particular focus on prices for higher volumes typically used in prototyping and small batch production.

The new price reduction programme is the latest in a series of initiatives from RS to provide customers with improved prices for design and production. From introducing Production Packaging for small batch production customers, through to flexible pricing services offered for larger orders, RS has consistently increased the level of pricing support offered to customers in the last 12 months.

Examples of the savings to be made on electronics components from RS include the FM series of electrolytic capacitors, with average reductions of over 22%, FFC connectors used for high density interconnection and PCBs, where prices have fallen by an average of 15%, and low drop out (LDO) positive and negative voltage regulators, with average reductions of over 10%.

rswww.com

(090231-XI)

The world's largest ever desktop Linux deployment

Userful and ThinNetworks have been selected to supply 356,800 virtualized desktops to schools in all of Brazil's 5,560 municipalities.

This initiative will provide computer access to millions of children and adolescents throughout the country. It is a historical achievement, being: the world's largest ever virtual desktop deployment; the world's largest ever desktop Linux deployment, and a new record low-cost for PCs with the PC sharing hardware and software costing less than $50 per seat.

Userful offers the features of a full PC including high performance video for less than $50 per additional seat in large deployments (not including monitors and keyboards) and uses standard PC hardware including additional low-cost video cards and USB/2-way-audio hubs from ThinNetworks. Userful and ThinNetworks are providing the desktop virtualizaton and PC sharing software and hardware while Positivo, Daruma, and Itaultec are providing the PCs and services. Userful and ThinNetworks have been selected in a competitive bidding process for all three phases of the project. The first phase, 18,750 workstations in rural schools, has already been installed and they are functioning well.

Userful's cost saving ability to turn 1 computer into up to 10 independent workstations enabled the Brazilian government to supply its schools with an unprecedented number of computer workstations. Savings of 60% in up-front costs, 80% in annual power savings and additional savings in ongoing administration and support costs as compared to a traditional PC-per-workstation solution all contributed to making 356,800 new workstations for Brazil's school children possible.

Desktop computers sit idle while we check our e-mail, surf the web, or type a document. Userful's PC sharing and virtualization technology leverages this unused computing power to create an environmentally efficient alternative to traditional desktop computing. Up to 10 users can work on a single computer by simply attaching extra monitors, mice and keyboards. "This deployment alone saves more than 170,000 tons of CO2 emissions annually, the same as taking 28,000 cars off the road, or planting 41,000 acres of trees", said Sean Rousseau, Marketing Manager at Userful. Turning 1 computer into 10 reduces computer hardware waste "e-waste" by up to 80%, further decreasing its environmental footprint. A free 2-user version of Userful Multiplier software for home use is available from the Userful website.

Project steward ThinNetworks is the exclusive distributor of Userful solutions in Brazil. Their unique knowledge and understanding of the challenges faced by the Brazilian Ministry of Education allowed ThinNetworks, with the help of Userful Multiplier, to successfully install low-cost computer labs in both urban and rural schools, where in most cases there are inadequate and insufficient facilities.

http://userful.com/free-2-user

(090231-XIII)

Free LED product characterisation tool

Cree, Inc. launched their Product Characterization Tool, an interactive LED design tool that simplifies the task of translating nominal LED performance to real-world conditions.

The online tool, accessible via the Cree website, allows users to easily characterise any XLamp® LED over a wide range of operating conditions, including drive current, flux bin, price and junction temperature. It also calculates metrics such as lumen output, lumens per watt, lumens per dollar and more.

The Product Characterization Tool (PCT) introduces advanced functionality not commercially offered by any LED supplier. PCT can perform simple LED system design based on a target total lumen output, calculating parameters such as number of LEDs required and total system efficacy. The calculated system parameters take into account electrical, optical and thermal losses associated with LED system performance.

In addition, PCT allows users to compare up to three different XLamp LED configurations at once, empowering customers to best choose between the industry-leading XLamp XR, MC and XP package families.

www.cree.com/pct

(090231-XII)

LED tube lamp replaces 40-watt fluorescent tube

Toshin Electric Co. has launched the 'Bikei', an LED tube lamp which can replace 40-watt straight tube fluorescent lamps. The company enhanced the heat radiation efficiency of the Bikei by forming aluminium radiator fins on part of its tube. Also, it downsized the substrate on which LEDs are mounted and minimized the shadow made by the substrate so that light reaches farther.

The luminance is 370 lx at 1 m below the lamp. The company expects the lamp to be used as lighting for tunnels, parking lots, subways, factories, stores and streets, for example. The lamp uses a total of 120 blue LED chips manufactured in Japan. White light is generated by combining units consisting of three blue LED chips with red and green fluorescent materials. As a result, its average colour rendering index (R_a) is as high as 92. General fluorescent lamp-type LED tube lamps have an R_a of about 70, while the R_a of normal fluorescent lamps is about 84 to 88, according to Toshin Electric.

Because the LED tube lamp is designed to use a socket for fluorescent tubes, it can replace a fluorescent lamp without changing or removing the socket. It supports both glow-starter and rapid-start methods. Its power consumption varies from 20 to 24 W depending on what lighting device it is used with. Its rated life is 40,000 hours, which reportedly is about five times the life of general fluorescent lamps.

The lamp uses polycarbonate for its tube cover, which is available in translucent white and transparent colours. The weight is 500 g. The suggested retail price is ¥28,000 (approx. $306) per unit for both models.

Toshin Electric is planning to sell the lamp in cases containing five units each. But it will also be sold in single units through the company's direct online shop in due course. The company aims to sell 50K units in the first year.

http://www.toshin-et.co.jp/

(090231-VIII)

UV LEDs provide uniform optical light Patterns

Providing design engineers with devices capable of uniform optical light patterns over an extended temperature range, TT electronics OPTEK Technology has developed a family of ultraviolet LEDs in the UV-A spectrum. Designated the OUE8A Series, the UV LEDs are offered in a variety of wavelengths and housed in hermetic TO-46 metal can packages.

The UV LEDs are ideal for medical and industrial applications including phototherapy treatment, dental applications, fluorescence and ultraviolet-visible spectroscopy; photo-catalyst curing of inks, coatings and adhesives; as well as paper currency and document validation.

Additional applications for the OUE8A Series UV LEDs include aircraft coatings and biohazard detection equipment/systems, skin dermatology, forensics, and automotive oil, A/C, and fluid detection.

The OUE8A Series UV LED family includes devices to cover many sub-bands in the 375 nm to 425 nm, with output powers from 1.8 mW to 15.4 mW.

All of OPTEK's UV LED devices have an 18º typical half power emission angle and a pulsed forward current of 1.6 A. Operating temperature ranges from –40ºC to +85ºC. The OUE8A Series UV LEDs are RoHS compliant and ESD protected for reliable operation.

www.optekinc.com/viewparts.aspx?categoryID=81

(090231-VI)

OK. Now you need a ... GPS system

SmartGPS module connected to *EasyPIC5* Development System

By Dusan Mihajlovic
MikroElektronika - Hardware Department

The Global Positioning System (GPS) is one of the leading technologies used for navigation purposes today. It is widely used in automotive navigation systems. Connection between a GPS receiver and the microcontroller as well as determination of latitude and longitude will be described here.

The Global Positioning System (GPS) is based on a large number of satellites radiating microwave signals for picking up by GPS receivers to determine their current location, time or velocity. GPS receivers can communicate with a microcontroller or a PC in different ways. A common path is via the serial port, while the most commonly used protocol for transmitting data is called NMEA.

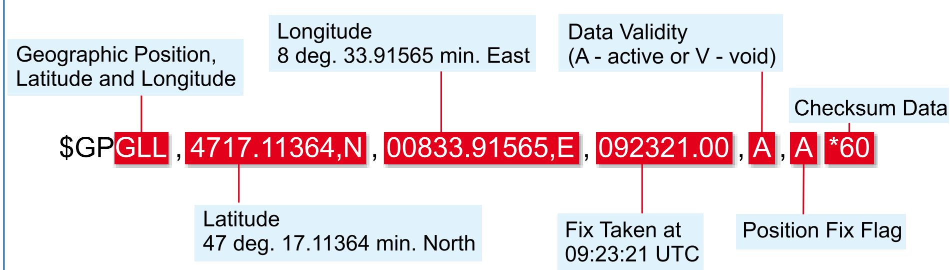
Principle of operation

The NMEA protocol is based on strings. Every string starts with the $ sign (ASCII 36) and terminates with a sequence of signs starting a new line such as CR (ASCII 13) and LF (ASCII 10). The meaning of the whole string depends on the first word. For example, a string starting with $GPGLL gives information about latitude and longitude, exact time (Universal Coordinated Time), data validity (A – Active or V - Void) and checksum enabling you to check whether data is regularly received. Individual data items are separated by a comma ‘,’.
Each second a set of NMEA strings is sent to the microcontroller. In the event that data on latitude and longitude are not fixed (i.e. if a GPS receiver fails to determine its location) or when other data is not determined, the GPS receiver will keep outputting the same set of strings, leaving out any missing data.

Here is a string generated by the GPS receiver which failed to determine its location:

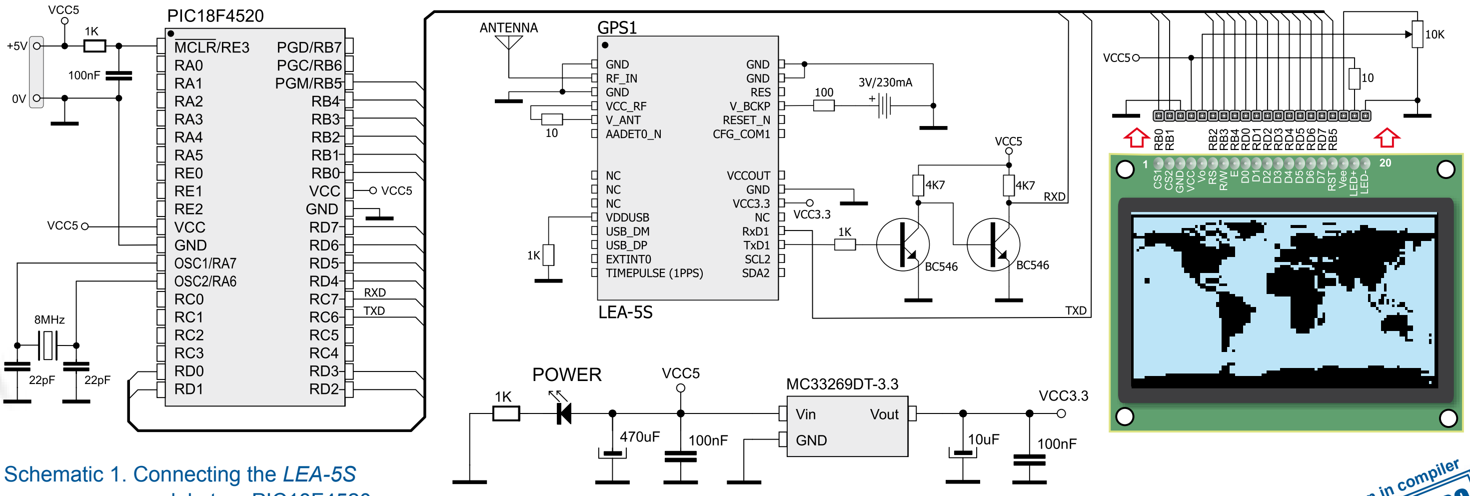
```
$GPGLL,,,,,,V,N*64
```

An example of a complete NMEA string is shown below:



Hardware

Connection between the microcontroller and GPS receiver is very simple. It is necessary to provide only two lines RX and TX for this purpose. Refer to the Schematic 1. The RX line is used for sending data from a GPS receiver to the microcontroller, while the TX line can be used for sending specific commands from the microcontroller to the GPS receiver. The U-Blox LEA-5S GPS receiver is used in this project.
Similar to most GPS receivers, the power supply voltage of this receiver is 3V.



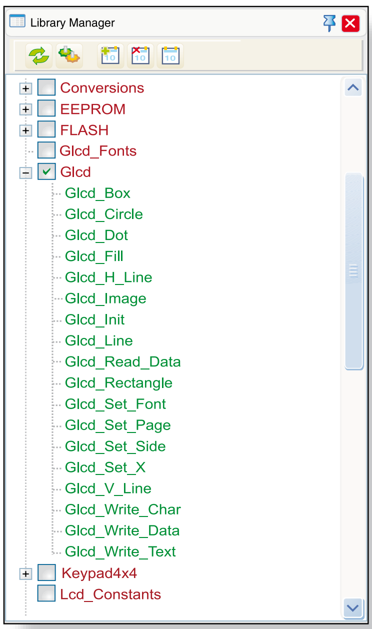
Schematic 1. Connecting the *LEA-5S* module to a PIC18F4520

Since the PIC18F4520 microcontroller uses a 5V supply voltage to operate, it is necessary to use a voltage level translator to convert the Logic One voltage level from 3.3V to 5V.
In this example, a graphic display with a resolution of 128x64 pixels displays a world map with the cursor pointing to your location on the globe.

Software

As you can see, the program code being fed into the microcontroller is very short. Nearly half the code constitutes a bitmap converted into an appropriate set of data. Such conversion enables the microcontroller to display the map. The rest of code consists of receiving NMEA strings from the GPS receiver, calculating latitude and longitude, scaling data to match the display resolution of 128x64 pixels and positioning the cursor at the specified location.

mikroC PRO for PIC® library editor with ready to use libraries such as: GLCD, Ethernet, CAN, SD/MMC etc.



Example 1: Program to demonstrate operation of *LEA -5S* module

Written in compiler mikroC PRO for PIC

```
char txt[768];
signed int latitude, longitude;
char *string;
int i;
unsigned short ready;
extern const unsigned short world_bmp[1024];
char GLCD_DataPort at PORTD;

sbit GLCD_CS1 at RB0_bit;       sbit GLCD_CS1_Direction at TRISB0_bit;
sbit GLCD_CS2 at RB1_bit;       sbit GLCD_CS2_Direction at TRISB1_bit;
sbit GLCD_RS  at RB2_bit;       sbit GLCD_RS_Direction  at TRISB2_bit;
sbit GLCD_RW  at RB3_bit;       sbit GLCD_RW_Direction  at TRISB3_bit;
sbit GLCD_EN  at RB4_bit;       sbit GLCD_EN_Direction  at TRISB4_bit;
sbit GLCD_RST at RB5_bit;       sbit GLCD_RST_Direction at TRISB5_bit;

void interrupt() {
 if (PIR1.F0 == 1) {            //if interrupt is generated by TMR1IF
  //Stop Timer 1:
  T1CON.F0 = 0;                 //Set TMR1ON to 0
  ready = 1;                    //set data ready
  i = 0;                        //reset array counter
  PIR1.F0 = 0;                  //Set TMR1IF to 0
 }
 if (PIR1.F5 == 1) {            //if interrupt is generated by RCIF
  txt[i++] = UART1_Read();
  if (i == 768) i = 0;
  //Stop Timer 1:
  T1CON.F0 = 0;                 //Set TMR1ON to 0
  //Timer1 starts counting from 15536:
  TMR1L = 0xB0;
  TMR1H = 0x3C;
  //Start Timer 1:
  T1CON.F0 = 1;                 //Set TMR1ON to 1
  PIR1.F5 = 0;                  //Set RCIF to 0
 }
}

void Display_Cursor(signed int lat, signed int lon) {
unsigned char latitude_y, longitude_x;

 //Latitude and Longitude scaling for 128x64 display:
 //Latitude: Input range is -90 to 90 degrees
 //Longitude: Input range is -180 to 180 degrees
 latitude_y = ((61*(90 - lat))/180) + 1;
 longitude_x = ((125*(lon + 180))/360) + 1;
 //Cursor drawing:
 Glcd_Dot(longitude_x,latitude_y,2);            //Centar dot
 Glcd_Dot(longitude_x-1,latitude_y,2);          //Left dot
 Glcd_Dot(longitude_x+1,latitude_y,2);          //Right dot
 Glcd_Dot(longitude_x,latitude_y-1,2);          //Uper dot
 Glcd_Dot(longitude_x,latitude_y+1,2);          //Lower dot
 Delay_ms(500);
    Glcd_Image( world_bmp );    //Display World
                                //map on the GLCD
 }
void main() {

  ADCON1 = 0x0F;                // Set AN pins to Digital I/O
  GLCD_Init();
  Glcd_Set_Font(font5x7, 5, 7, 32);
  Glcd_Fill(0x00);
  Delay_ms(100);
  ready = 0;

  //Set Timer1 Prescaler to 1:8
  T1CON.F5 = 1;                 //Set TCKPS1 to 1
  T1CON.F4 = 1;                 //Set TCKPS0 to 1
  //Enable Timer1 interrupt:
  PIE1.F0 = 1;                  //Set TMR1IE to 1
  //Timer1 starts counting from 15536:
  TMR1L = 0xB0;
  TMR1H = 0x3C;
  //Clear Timer1 interrupt flag:
  PIR1.F0 = 0;                  //Set TMR1IF to 0
  //Note: Timer1 is set to generate interrupt on 50ms interval

  UART1_Init(9600);
  //Enable Usart Receiver interrupt:
  PIE1.F5 = 1;                  //Set RCIE to 1
  //Enable Global interrupt and Peripheral interrupt:
  INTCON.F7 = 1;                //Set GIE to 1
  INTCON.F6 = 1;                //Set PEIE to 1

  //Start Timer 1:
  T1CON.F0 = 1;                 //Set TMR1ON to 1

  Glcd_Image( world_bmp );      //Display World map on the GLCD

  while(1)
  {
   RCSTA.F1 = 0;                //Set OERR to 0
   RCSTA.F2 = 0;                //Set FERR to 0

   if(ready == 1) {             //if the data in txt array is ready do:
    ready = 0;
    string = strstr(txt,"$GPGLL");
    if(string != 0) {       //If txt array contains "$GPGLL" string we proceed...
     if(string[7] != ',') {  //if "$GPGLL" NMEA message have ',' sign in the 8-th
                  //position it means that tha GPS receiver does not have FIXed position!
      latitude = (string[7]-48)*10 + (string[8]-48);
      longitude = (string[20]-48)*100 + (string[21]-48)*10 + (string[22]-48);
      if(string[18] == 'S') {     //if the latitude is in the South direction it has minus sign
       latitude = 0 - latitude;
      }
      if(string[32] == 'W') {   //if the longitude is in the West direction it has minus sign
       longitude = 0 - longitude;
      }
      Display_Cursor(latitude, longitude);   //Display the cursor on the world map
     }
    }
   }
  }
}

unsigned char const World_bmp[1024] = {
255,129,1,1,1,129,129,129,129,193,129,129,129,129,129,129,129,129,129,129,129,
225,161,161,97,97,209,209,129, 49, 49,201, 201,201,201, 97,205,205,129,137, 25,5
7,57,57,121,249,249,249,249,249,253,253,121,121,113,9,9,1,1,1,1,1,1,1,1,1,1,17,17,
145,145,145,145,129,129,129,1,1,1,1,9,73,73,73,73,193,65,65,129,129,193,193,129,
193,193,241,241,241,241,225,225,225,193,193,193,193,193,193,193,193,193,129,
193,193,225,225,129,129,129,129,129,129,129,129,129,129,129,255,255,1,33,17,17,
15,15,15,15,15,7,7,7,7,15,15,31,63,63,63,63,255,255,255,255,255,255,255,255,251,2
51,240,240,240,240,226,252,252,249,249,250,240,240,1,1,1,1,3,1,1,0,0,0,0,0,2,2,0,0,
0,24,24,224,224,224,224,244,239,239,255,255,255,255,255,255,255,255,255,254,25
4,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,95,95,3,3,3,3,63,15,15,3,3,3,
3,3,1,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,15,63,63,255,255,255,255,255,63,
63,63,63,63,63, 63,135,135,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,192,192,192,243,243,2
51,251,251,251,251,247,231,231,243,247,247,247,230,236,124,124,255,255,220,60,
61,61,63,126,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,59,59,3,7,3,27,12,7,7,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,1,1,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,3,6,6,13,13,13,13,17,242,242,242,242,240,224,224,192,
192,192,192,0,0,0,0,0,0,0,0,0,0,31,31,31,63,63,63,63,63,63,255,255,255,255,255,255
,255,255,255,255,248,248,247,247,55,3,3,3,3,0,1,1,3,3,15,15,7,0,0,1,1,3,3,239,15,15,
1,129,224,174,46,128,0,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,63,63,255,255,255,255,255,255,255,
255,255,255,254,254,12,12,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,255,255,255,255,255,255,25
5,255,255,63,63,193,193,240,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,4,9,129,193,192,
225,224,226,224,242,227,227,228,228,8,8,0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,31,31,15,15,15,15,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,15,15,15,15,15,3,3,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,15,15,7,7,7,7,7,31,31,127,127,70,70,0,0,0,0,0,0,208,208,0,2
55,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,135,135,1
93,64,68,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128
,128,128,128,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128,128,128,128,128,0,0,0,0,0,0,128,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,240,240,240,240,248,248,248,248,248,248,248,2
48,248,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,25
2,252,252,252,252,254,254,255,255,255,252,252,248,248,248,248,248,248,248,248,
248,248,248,252,252,252,254,254,254,254,254,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,254,254,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,250,250,250,216,216,248,255};
```

Functions used in the program

Glcd_box()	Draw filled box
Glcd_circle()	Draw circle
Glcd_Dot()	Draw dot*
Glcd_Fill()	Delete/fill display*
Glcd_H_Line()	Draw horizontal line
Glcd_Image()	Import image*
Glcd_Init()	LCD display initialization*
Glcd_Line()	Draw line
Glcd_Read_Data()	Read data from LCD
Glcd_Rectangle()	Draw rectangle
Glcd_Set_Font()	Select font*
Glcd_Set_Page()	Select page
Glcd_Set_Side()	Select side of display
Glcd_Set_X()	Determine X coordinate
Glcd_V_line()	Draw vertical line
Glcd_Write_Char()	Write character
Glcd_Write_Data()	Write data
Glcd_Write_Text()	Write text

* Glcd library functions used in the program

Other *mikroC PRO for PIC* functions used in program:

Usart_Init()	strstr()
Usart_Read()	Delay_ms()

GO TO Code for this example written for PIC® microcontrollers in C, Basic and Pascal as well as the programs written for dsPIC® and AVR® microcontrollers can be found on our website: **www.mikroe.com/en/article/**

Experimenting with

Low-cost development system with a USB interface

Emile van de Logt, MA MSc (The Netherlands)

Texas Instruments supplies handy USB evaluation sticks with related software for its low-cost MSP430 controllers. Unfortunately the I/O facilities are somewhat limited. These can be substantially enhanced with the help of the experimenter's board described here. This combination forms the hardware basis for a mini-course 'Starting with embedded C', which can be found elsewhere in this issue.

Sometimes several initiatives converge at just the right time to create a new concept. For some time Rotterdam University had been looking for a low-cost development system for its students in Automotive and Electronic Engineering, which could be put to use in microcontroller tuition. In addition, for logistical reasons Elektor was looking for a more practical replacement for the very popular E-blocks for its Embedded C Programming workshops. So both Rotterdam University and Elektor were effectively looking for the same thing, although for different reasons.
Once contact was established between Rotterdam University and the Elektor labs, it didn't take long before it was decided to combine forces. The advantages of both the E-blocks as well as the configuration used by Rotterdam University were examined and a joint specification was produced.

For students it is obviously important to obtain the hardware and software as cheaply as possible. The accompanying development system should ideally be included free of charge. It should also be easy to use so that new students can quickly create their first program. Preferably not in assembly language, but in C using a full-featured C compiler! It should be enjoyable to work with, so preferably it should be able to create sound (buzzer), display numbers (7-segment display), flash (LEDs), work via the USB port rather than the parallel port (modern laptops no longer have these) and also include further expansion possibilities (I²C, SPI). It would also be nice not to use an 8-bit system any more, but rather something with a bit more muscle (16 or 32 bits).

Figure 1. The eZ430 evaluation stick, around which this experimenter's system is based.

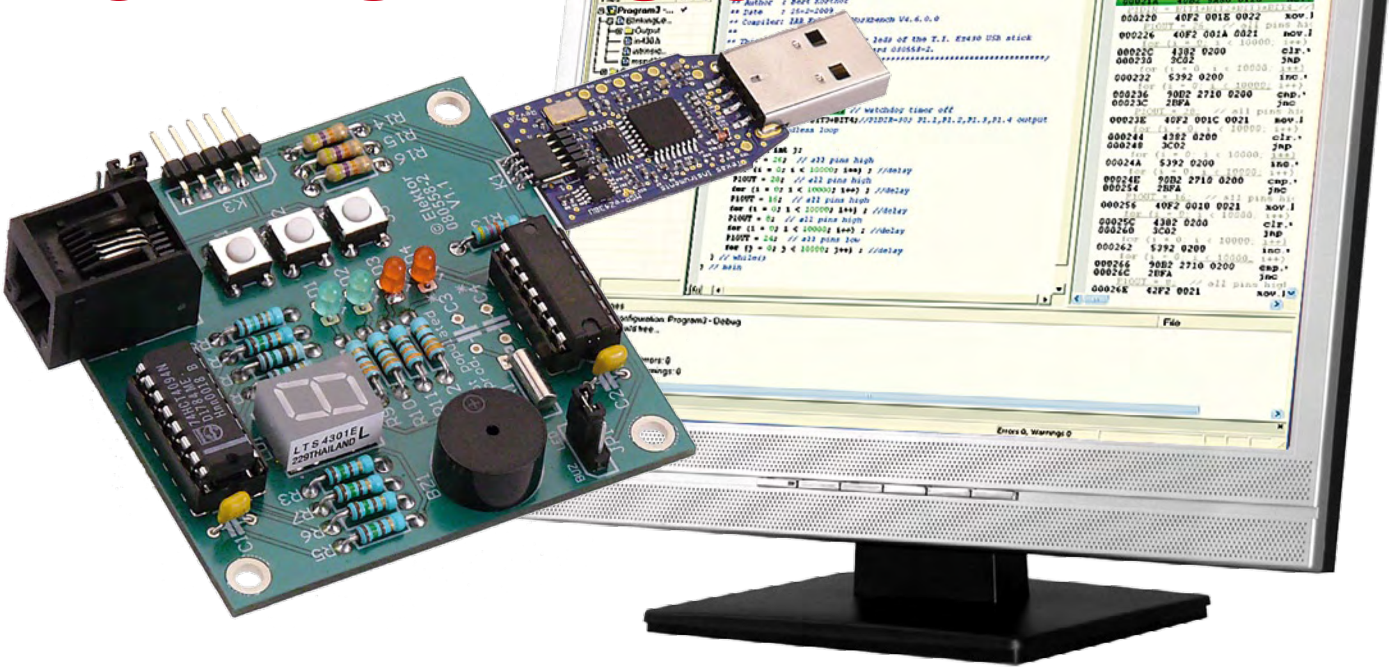
History

Not long ago an average electronics department would have to raise an expense request to obtain such a microcontroller system. These systems were often large dedicated computers, which required expensive software suites to develop embedded applications. Via an ICE (In-Circuit Emulator) the program could then be debugged on the target system.
If we said at the time that we could obtain such a system for less than 50 euros (or the equivalent in pounds or dollars) we would have been met with looks of disbelief. Despite this, there are currently students walking around with such a system in their backpack and they have lessons in programming such a microcontroller!

The MSP430 series

All the big electronics manufacturers supply microcontrollers offering a wide range of functions. However,

the MSP430



when the above criteria are taken into account there is one manufacturer that stands out a bit from the rest, and that is Texas Instruments (TI). The MSP430 series in particular consists of a range of full-featured microcontrollers with a large number of I/O facilities. The most important properties are that they require very little power to run and that they contain a 16-bit processor core.
To keep things as simple as possible for the average electronic engineer, TI has designed evaluation 'boards' for this type of controller, where the complete hardware environment is housed inside a USB stick (**Figure 1**). This hardware environment is known as the eZ430. To this you can connect (via a so-called Spy-Bi-Wire connection) various target boards.
A complete system like this, including a C compiler, can be obtained for the unbelievably low price of under £20! After installing the software and plugging in the USB stick you can immediately start with programming this fascinating microcontroller.

But are there any disadvantages with this system? Unfortunately there are, since the eZ430 system only has limited I/O facilities. The target board inside the USB stick is very small and therefore has only a single LED and one connector. This is not enough if it is to be used as a training tool or in microcontroller workshops.

Specifications

- **Experimenter's board with several I/O possibilities**
- **Powerful 16-bit MSP430F2012 controller running at 16 MHz, 2 KB Flash and 128 bytes RAM**
- **4 indicator LEDs**
- **7-segment display**
- **Piezo buzzer**
- **Three pushbuttons**
- **I²C/SPI connector**
- **Powered via the USB stick or an external adapter**

The Elektor MSP430 board

There are basically two ways in which the number of I/O facilities can be increased in this system:

1. Add a new I/O board to the target board using the existing connector.

2. Design a new target board with the number of required I/O facilities and connect this to the USB stick using the Spy-Bi-Wire interface.

For various reasons it was decided to go with the second option. A separate I/O board using a connector isn't very stable and could sometimes give rise to connection problems. Furthermore, the target board is cheap enough that it makes little difference to the total cost whether the microcontroller is or isn't part of the I/O board.
The circuit diagram of the MSP430 board designed by Elektor and the lecturers from Rotterdam University can be seen in **Figure 2**. The most important part on this I/O board is of course the microcontroller itself, an MSP430-F2012 (IC2). The reason for using this particular microcontroller and not the F2013 as supplied by TI in the eZ430-F2013 kit is that the F2012 has a different type of A/D converter that has a greater range. Apart from that, both microcontrollers are identical.
The experimenter's board has a set of three pushbuttons that are connected

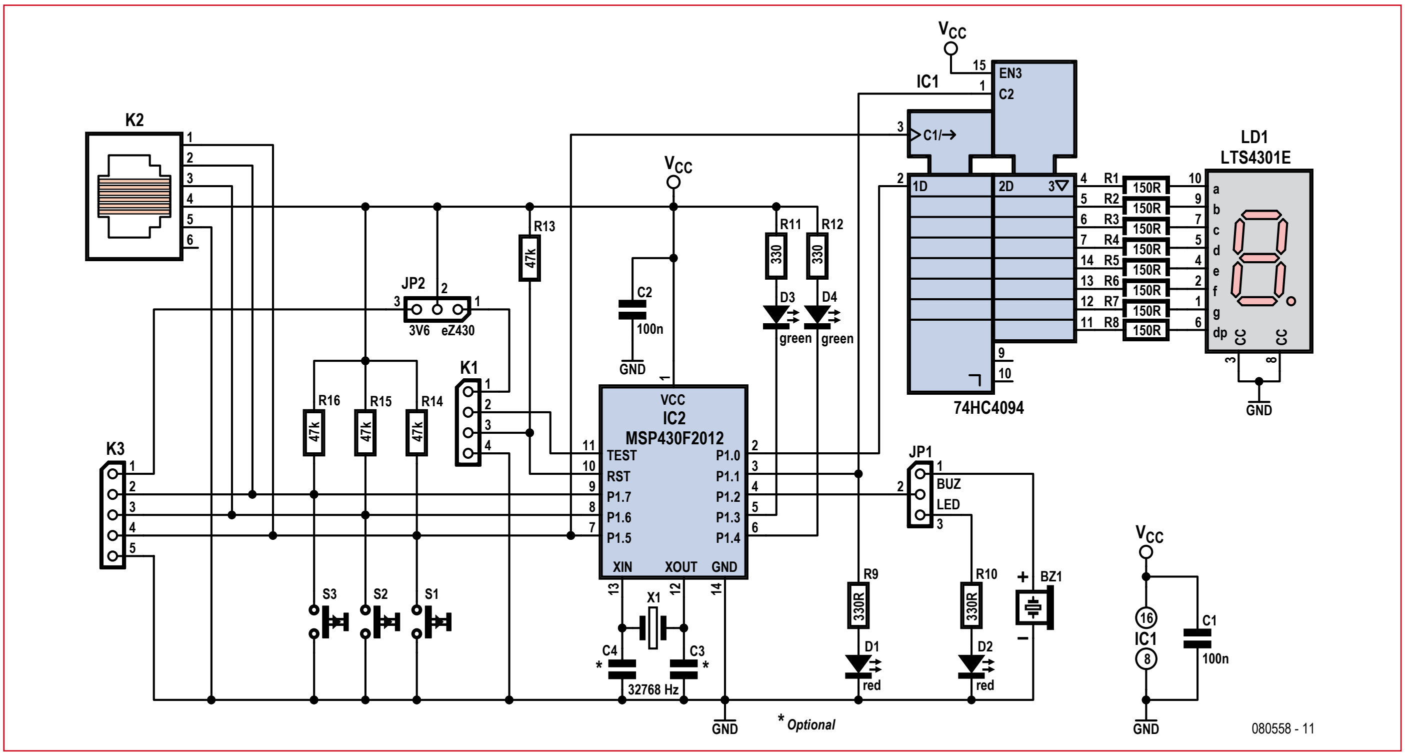


Figure 2. The circuit diagram for the MSP430 experimenter's board.

to P1.5, P1.6 and P1.7. Most of the pins actually have several functions, depending on how they've been configured via the software. Four LEDs have also been included, two green and two red. The green LEDs are connected to the positive supply via a resistor and the red LEDs are connected to GND via a resistor. To light up a green a logical zero has to be programmed on the relevant I/O pin. For red LEDs this is the exact opposite: with these a logical one has to be programmed. It was done this way to quickly give students an insight into the differences of the common configurations that can be found.

JP1 can be used to manually select either the red LED (D2) or the buzzer (BZ1). When the buzzer has been selected via JP1, it is possible to create a tone by quickly switching between logic zero and logic one on pin P1.2. Increasing or decreasing the rate at which the level switches causes the frequency of the tone to change.
7-segment display LD1 is connected to the microcontroller via shift register IC1 (a 74HC4094). To display a digit on LD1 you first need to send an 8-bit code serially to IC1, after which input C2 of IC1 is made high via P1.1. The serial transmission of data to IC1 is done via pins P1.5 (clock) and P1.0 (data), where the most significant bit is sent first.
The board gets its 3.3 V supply voltage via the Spy-Bi-Wire interface (K3). There is therefore no need for a separate power supply, except in cases where the board is used in stand-alone applications. In those cases the setting of JP2 has to be changed so that the 3.3 V supply voltage can come via K3 (pins 1 and 5).
We have also thought about expansion possibilities. The microcontroller has a Universal Serial Interface (USI), which can function as either an SPI or an I^2C serial communications interface. These signals have been made available on both K2 (standard I^2C connector) as well as K3 (expansion connector). The required I/O pins are listed in **Table 1**.
We haven't described the A/D converter yet. The F2012 has a built-in 10-bit analogue-to-digital converter with eight input channels. It is also able to read the state of an internal temperature sensor and measure the value of the supply voltage.

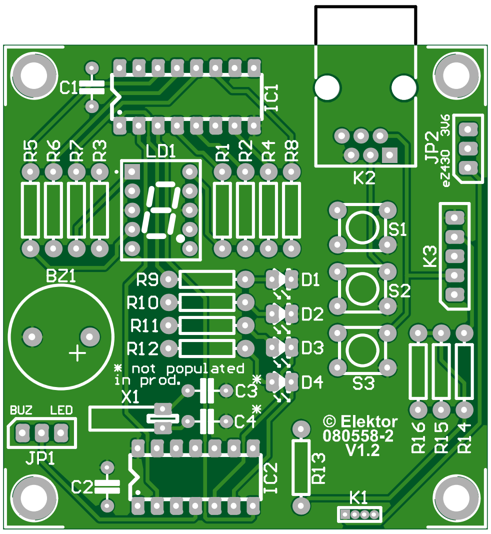
Table 1. I/O pins for external communications

I/O pin	SPI	I^2C	ADC
P1.5	clock		ADC 5
P1.6	serial data out	clock	ADC 6
P1.7	serial data in	serial data	ADC 7

Construction

Now that we've given an overview of the experimenter's board and its various functions, it's time to put this I/O board to work. We decided early on to populate the I/O with standard components and not SMDs, so that the construction wouldn't present any problems for less experienced constructors.

In **Figure 3** you can see the PCB that was designed for the I/O board. Soldering all the parts should be quite straightforward. As usual you should take care with the polarity of the LEDs and ICs. Elektor also supplies a completely populated board, including the sometimes difficult to obtain connector for the connection with the USB stick (Elektor Shop # 080558-91). The EZ430 USB stick can also be obtained from Elektor (Elektor Shop # 080558-92).
The board is connected to the USB



COMPONENT LIST

Resistors
R1-R8 = 150Ω 0.25W
R9-R12 = 330Ω 0.25W
R13-R16 = 47kΩ 0.25W

Capacitors
C1,C2 = 100nF
C3,C4 = optional, not fitted here (see TI Appl. Report SLAA322)

Semiconductors
D1,D2 = low-current LED, red, diam. 3mm
D3,D4 = low-current LED, green, diam. 3mm
LD1 = 7-segment LED display, common cathode (e.g. Lite-On LTS4301E)
IC1 = 74HC4094
IC2 = MSP430F2012IN (TI)

Miscellaneous
S1,S2,S3 = PCB mount pushbutton, 5x5mm (e.g. Tyco FSM4JH)
K1 = right angled 4-pin connector, lead pitch 1.27mm (Mill-Max # 851-93-004-20-001000)
K2 = 6-way RJ11 connector, PCB mount (Molex # 95009-2661)
K3 = 5-way SIL pinheader
JP1,JP2 = 3-way pinheader and jumper
BZ1 = passive piezo buzzer (e.g. Kingstate# KPEG242)
X1 = 32.768kHz quartz crystal
16-pin IC socket for IC1
14-pin IC socket for IC2
PCB, # 080558-2
Ready assembled and tested board: Elektor Shop # 080558-91
TI eZ430-F2013 evaluation kit: Elektor Shop # 080558-92

Figure 3. PCB layout for the board. Mini-connector K1 is used to connect the USB stick (via the middle four pins!).

stick via the Spy-Bi-Wire interface. It's easiest if you open the plastic housing of the stick and remove the circuit boards; the target board can then be removed and in its place you can connect the experimenter's board.

In practice

Does it work? Certainly! Although in this context by 'work' we don't mean if the hardware functions correctly, since that won't be a problem in most cases. Instead we mean if it works well in an educational environment, where students have to familiarise themselves with the complexities of programming in the C language on embedded systems.

The results, both at the Rotterdam University as well as in the Embedded C Programming workshops at Elektor, are very positive. It appears that participants can quickly learn to write simple programs in the C language and can get them to work on the experimenter's board. Most of the participants of these courses won't have had much experience in programming in C, but it isn't long before they start to delve into complex tasks such as writing timer interrupts, creating functions and driving the hardware on board. In subsequent projects, for courses both in Automotive as well as Electronic Engineering, we often come across the same hardware again!

In the Embedded C programming workshop of Elektor a completed and tested version of the board is used, obviously in conjunction with the eZ430 kit from TI. This complete kit can also be ordered by students, so there is no need to order them separately.

Do-it-yourself

After reading this article we wouldn't be surprised if many readers would also like to try their hand at embedded C with the help of this experimenter's board.

A description of the software needed to drive all hardware in this I/O board will be covered in a short three-part course, again in collaboration with Rotterdam University. The first article can be found in this issue of Elektor. It covers the first few steps: the installation of the development environment and the testing of the completed board using the first example program.

(080558-I)

The author

Emile van de Logt is an Electronic Engineering training manager at Rotterdam University.

He studied Electronic Engineering at the Technical University in Eindhoven and Management Studies at the Open University.

Emile spends his spare time designing electronic circuits and he is an amateur beer brewer. He also takes care of the Embedded C Programming workshop and the FPGA-VHDL workshop for Elektor.

Getting started with

Part 1: IAR Embedded Workbench and flashing LED

A.J. (Bert) Korthof (The Netherlands)

This is the first instalment of a three-part series which will introduce the fundamentals of programming a microcontroller in C. You can immediately try all the examples using the MSP430 hardware, which is also described in this issue, in combination with a PC or laptop which has a USB interface. The software we've used is available as a free download. In this way you will learn step by step how you can use the higher programming language C in all kinds of electronics projects.

C is a genuine general-purpose programming language (there are over 400 different languages for computer systems). C is a small, compact language, which is not all that difficult to learn. These days C is used mostly in embedded microcontroller applications. This means devices that contain a microcontroller doing one specific task, such as a coffee maker (compare that to the processor in a PC which runs a variety of programs). The Java language is also quite frequently used for this, but it places much higher demands on the hardware, specifically in terms of speed and memory.
One or more C compilers are available for virtually every commercially available processor. An international standard

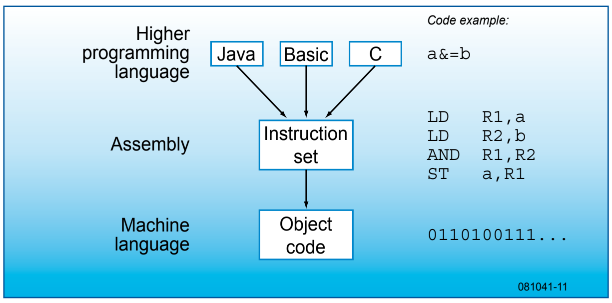


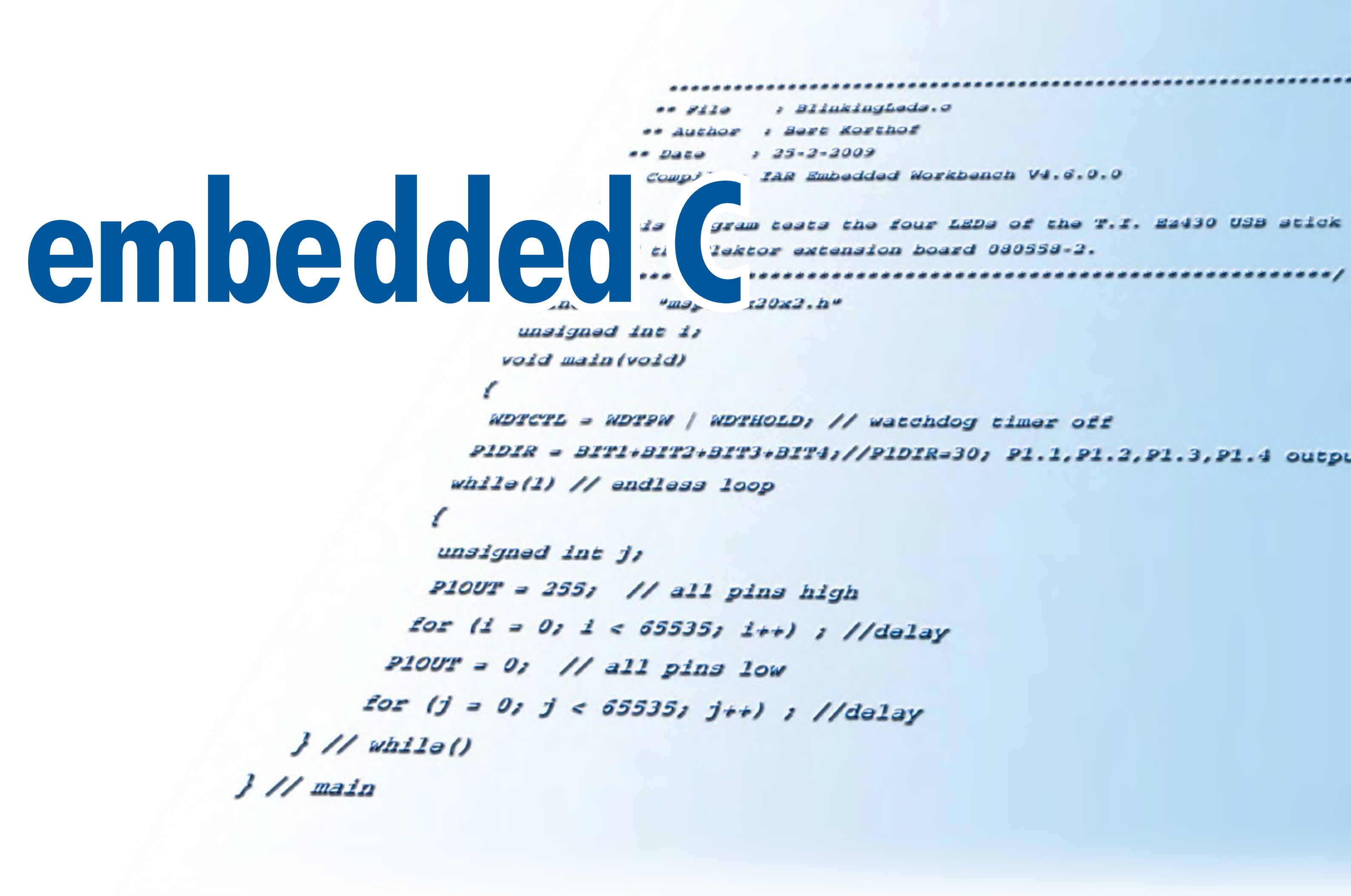
Figure 1. The instructions of a higher programming language are converted into machine language that the processor can understand.

for C has been established: ANSI-C (end of 1988). There are standard library functions, function declarations and definitions. You can really only learn C++ once you know C. As a little known fact, C evolved from the language B. Windows and Unix operating systems are typically written in C or C++. The C language is close to the hardware on which the program will run. The C program lines are converted by the C compiler into assembly language: this language is the closest to the hardware: the (micro)controller (see **Figure 1**).
Nowadays, programming in assembly language is usually only done if the code needs to be extremely compact or run very quickly. Every family of processors, such as those made by Atmel, Microchip and Texas Instruments (TI) has their own unique instruction set and you have to know all the registers and memory locations really well and write much more code yourself, such as for tasks like multiplication or division.

Processor

For this course we chose the MSP430 family made by TI. These are powerful 16-bit processors which are eminently suitable for battery-powered applications such as measuring instruments and intelligent sensors. The specific processor that we use here is the MSP430F2012. Here are a few of its salient features:

- Power supply voltage from 1.8 to 3.6 V
- Internal clock up to 16 MHz
- A 32 kHz watch crystal can be connected directly
- 2 timers which can be used for accurate timing measurement or pulse generation.
- 2 Kbyte flash memory for code and the storage of parameters (non-volatile)
- 128 bytes of RAM for variables
- 10-bit A/D-converter at up to 200 ksamples per second
- USI (universal serial interface), can be used for SPI and I²C



embedded C

The amount of memory available for your own programs is quite small, but you will be surprised how many useful programs (such as interfacing with sensors, controlling simple machines (state machine) data conversion, counters, security applications, etc.) can be made to fit in this small space. The C compiler used here is supplied by IAR and converts code efficiently into machine language. Just about anything that is possible in C you can learn using this compiler. In addition, you can use the same software for the bigger and more powerful processors from the MSP430 family as well!

Hardware and software

This first article describes the organisation of the development environment for programming in embedded C, so that you can easily begin writing your own simple programs and debug them in real time or single step by executing the code in the microcontroller on the Elektor PCB, number 080558-2. The board contains, of course, a microcontroller to run the code and also several examples of sensors (push buttons) and actuators (LEDs, 7-segment display, buzzer), see **Figure 2**.

For the development environment we use the IAR Workbench KickStart software, which is supplied by TI accompanying the eZ430 USB stick.

Making a start with C

We cannot cover an entire book worth of C in these three articles, but there are already plenty of C books and very good courses are available on the Internet (see [1] and [2]).

The C language is not all that difficult to learn, there are only 32 keywords (**Table 1**), C simply does not know any more words — compare that to English or any other normal language.

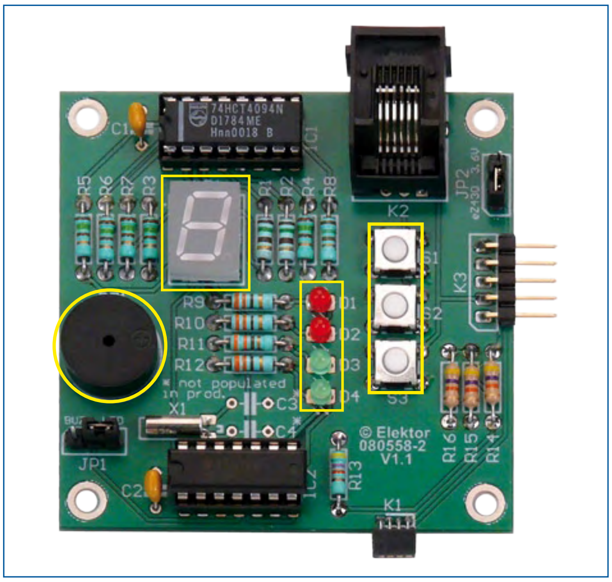


Figure 2. The experimenting board contains several sensors and actuators for interaction with the user.

Table 1. The standard version of ANSI-C has only 32 keywords.

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Here we cover the details that are specific to our hardware and software, details which are not in a C book because the C language is universal! We learn the basic rules for C programs which can run on an ordinary PC. To do this we use the standard header file: *#include "stdio.h"*, which contains the definition for the commands *printf* and *scanf*. This is used to define the standard input and output channels for the hardware that is used.
A standard C program consists of declarations of variables and functions. The function *main()* must always exist. This contains the statements that are carried out sequentially, one after the other. *Main* begins with a left brace and ends with a right brace. Every statement is terminated with a semicolon (;).

The names of variables can be chosen freely, but in the C language we have to indicate clearly what type it is, for example the variable *i*: unsigned int i.
To the MSP430 processor this means an integer in the range from 0 to 65535, the processor by default works with 16-bit numbers.

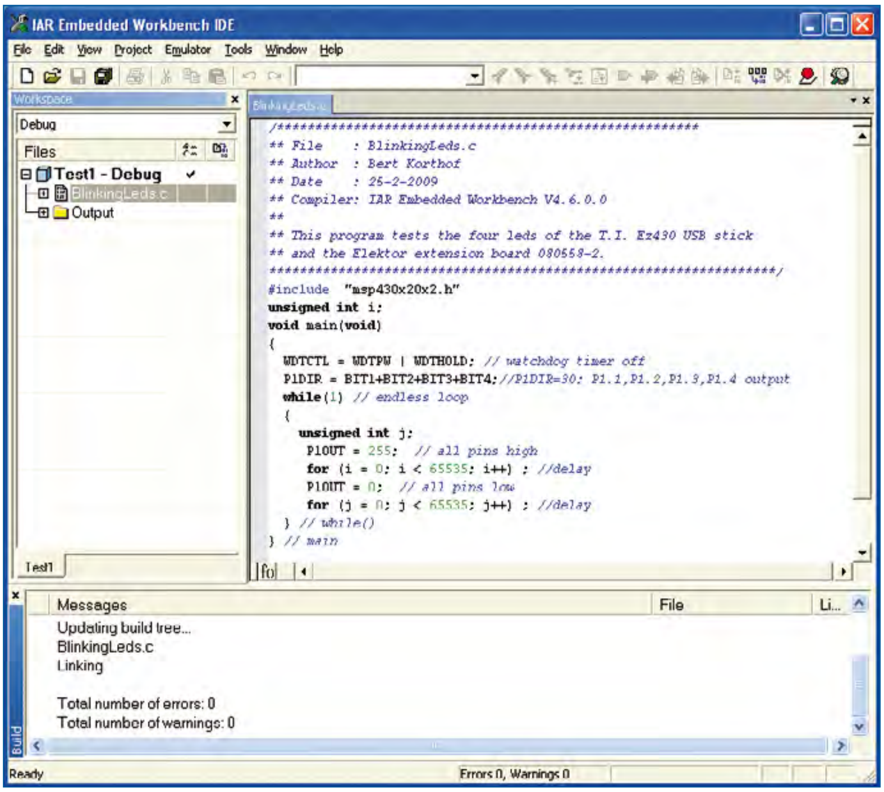


Figure 3.
The file BlinkingLeds.c is linked to the project in IAR Embedded Workbench.

All text between /* and */ is treated as a comment by the compiler. We can also add comments after //.
We obviously do not have a microprocessor board to which we can connect a printer or keyboard (this requires a much more powerful processor). However, we can 'print' by showing numbers on the display and scan the state of the push buttons (*read*).

Each of the port pins of this processor can be individually configured as either an input or an output. We can connect logic-level signals (0 or 3.3 V) to an input, for example using a switch. You cannot do this to an output of course! (Take note: you can get a high current when you connect an output pin set to a High level, to 0 V through a switch!)
For safety, the default values of the bits in the port pin direction registers are cleared, so that the ports are initially configured as inputs!

First program

Our first little C program will drive four LEDs. When we look at the schematic in the construction article we can see that the LEDs are connected in different ways via resistors to microcontroller port P1! To turn the red LEDs on, the microcontroller has to put a logic High level (power supply voltage, 3.3 V) on port pins P1.1 and P1.2. This is called active High. To turn the green LEDs on, port pins P1.3 and P1.4 have to be made Low (0 V), because a pull-up resistor is used here. These are therefore active Low. As a programmer we have to keep these things in mind. To prevent time-consuming mistakes in the code the 'software guy' will therefore also have to be familiar with the hardware.
Launch the IAR Workbench, incorporating the C compiler, simulator and debugger. Then create a new workspace for a new project which contains the C statements in a text file which you call *BlinkingLeds.c*, so that the C compiler can recognise this as a C program. In addition you have to tell the compiler which hardware this program will run on. Because this requires going through a number of steps and the selection of various options, we have described this process in some detail in a supplementary article *Getting started with IAR Workbench* which is available free from the Elektor website.
We will assume that you have done all this and have opened the file *BlinkingLeds.c* and have linked it to your project, as can be seen in **Figure 3**.
The program (the source code) is compiled (translated into machine code) by clicking on:

At the bottom of the window we can see that there are no C language errors in the code and how much code and data memory we have used. Although there are no syntax errors in the program, the C compiler cannot, of course, tell us whether the program operates as it should! This we have to check for ourselves!
In the code we can see words such as *BIT1* (binary for 0 ... 010), *P1OUT* (outputs of port 1) and *WDTCTL* (control register for the watchdog timer, this will reset the processor if the program gets stuck). The definitions for these words are in the header file *msp430x20x2.h*. This also contains all the features of the processor that we are using, such as the addresses of the ports, memory size, special registers for the timers, clock generator, etc.
With the statement *P1DIR = 30* (=2+4+8+16) the correct bits in the port direction register are set High so that the port pins for the four LEDs (P1.1 through P1.4) are set to outputs. A port pin which is configured as an output can supply up to about 5 mA, sufficient to drive an LED directly!

With the instruction *P1OUT = 255* (binary 11111111) we make all eight bits of port 1 High. Only the port pins to which the LEDs are connected will go High. The other pins do not go High because they are not configured as outputs.

Structure of the BlinkingLeds program

As already noted, the statements between the braces of the 'main' function are executed sequentially. If this were the only option then our program would be very long. In C we can also make program loops and program jumps: with the statement *while(condition = true)* all the code between the braces is repeated until the condition is no longer true. Here we use *while(1)*, were 1 means 'true' (0 means 'false'). The while-loop is therefore repeated forever (or until the power supply is disconnected or the reset pin of the processor is activated). In addition we also see a *for(…,…,…)* loop, with which we let the processor count from 0 to 65535 (this is the largest positive number that we can represent with 16 bits; 2^{16}–1). This loop is added twice to create a software delay of about 2×0.5 seconds, so that we can clearly see that the LEDs are flashing. For this we declared the variables i and j, where j is a temporary variable, the memory location of which is available to be reused for other variables (this reduces the amount of the — limited — RAM that is used).

After this brief explanation we continue with IAR Workbench to 'flash' the program into the microcontroller by clicking on C-Spy:

We assume that the board is connected to the USB port via the MSP-eZ430 USB interface board and all settings are configured according to the document: *Getting started with IAR Workbench*.

We now arrive in the debug mode and can manually run through the program step by step and watch the values of the variables at the same time (**Figure 4**).

We can open a Watch window by selecting Watch in the View menu and adding the variables 'i' and 'j' in the dashed rectangles.

You can experiment for yourself with Single-step-mode, the RUN mode (we can now see the red and green LEDs flash!), Break (the next statement which is ready to be executed is shown in green) and stop using a reset.

Running through the for-loop in single-step mode gives little information and will take a very long time. We can change the variables 'i' and 'j' in the Watch window by clicking on their value and typing 65534, for example... and with only a few more steps we're out of the loop!

Masking of bits

The C language can do many things, but we cannot, for example, directly change a single bit to logic High (1) or Low (0)! For example, using the statement

P1OUT = BIT1; (or *P1OUT = 2*;) we can make the second bit High, the red LED D1 will turn on, but this will cause the other port pins to be Low! This could result in other important actuators such as an alarm or motor to be turned off or even on. We can solve this annoying problem by the masking of bits: If *P1OUT* has the value, for example, of 01…101 and we only want to make *BIT1* High and leave the other bits unchanged then we first use a logical-OR function with 00…010 and send the result to the port pins. With the OR function all bits remain the same, except *BIT1* which goes from 0 to 1 (if it was already 1 then it remains 1).

In the C language we can indicate these two operations as follows: *P1OUT (new value) = P1OUT (old value) | BIT1* (*|* is in C the bit-wise OR function). The C language is well-known for its concise notation, so it can therefore also be written shorter: *P1OUT |= BIT1*.

Another example of this compact notation: *i++* means: read the value of *i* from its memory location, add 1 and write the result to the original location (the original value is therefore lost).

The OR function is necessary for setting a bit (making it High). For resetting (0) we require the AND function (bit-wise AND; in C the symbol for this is *&*). Say we want to make the third bit Low. We need to make a mask with the inverse of 00...0100 and use this in an AND function. the bit-wise operator for inversion is the ~. The short notation therefore becomes: *P1OUT &= ~BIT2*.

Example: *P1OUT = 01010101*. We want to reset the last bit only. Use a mask that is the inverse of 00000001 (this is 11111110) and use this number in a logic AND function with the old value of the port: *01010101 & 11111110 = 01010100*.

Finally an interesting exercise: Change the program *BlinkingLeds.c* so that you obtain a running light where each of the LEDs turn on one after the other. Don't forget the for-loop to obtain a delay, otherwise the LEDs will change every few microseconds and it will appear that they are all on at the same time, because of the persistence of our eyes.

Try it for yourself!!

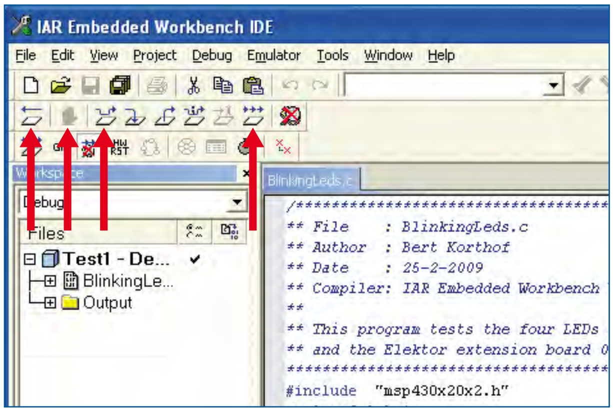


Figure 4.
In de debug-mode we can run through the program step by step and at the same time examine the values of variables.

The example program *BlinkingLeds.c* can be downloaded from the web page belonging with this article (www.elektor.com/081041) filed under number **081041-11**. The supplement *Getting started with IAR Workbench* can also be found here, filed under number **081041-W**.

(081041-I)

About the Author

Bert Korthof is a Lecturer in the department of Automotive Technology/Electrical Engineering at Rotterdam University.

Internet Links

[1] www.lysator.liu.se/c/bwk-tutor.html

[2] www.cprogramming.com/tutorial/c/lesson1.html

USB 3.0 SuperSpeed

José Luis Rupérez Fombellida (Spain)

The Universal Serial Bus (USB) was introduced in 1996 to allow easy connection of different devices to a computer by means of a fast serial link. Since then, USB has gradually become the widest used connection system for PCs, forcing at least two traditional connection systems, RS232 and Centronics, to take a distant back seat. USB has seen many speed increases and other enhancements over the years and release 3.0 seems to hold another promise.

Although USB has evolved since its introduction, its principal advantages have remained unchanged from the first version and can be summarized as:

- 'Live' connection / disconnection (hot swap). To connect or disconnect a USB device it is not necessary to switch off the computer.

Figure 1. The USB 3.0 logo.

- Bus Power: the USB can, in most cases, power devices.
- Plug and Play: the device connects to the USB and is almost instantly ready for use. In some cases, it is necessary to install a driver.

The USB Implementers Forum (USB-IF, [1]) is in charge of developing USB regulation in its entirety. USB-IF is formed by several companies: Hewlett-Packard Company, Microsoft Corporation, Intel Corporation, NEC Corporation, ST-NXP Wireless and Texas Instruments.
Today, the USB is so widespread that it has practically eliminated the parallel (Centronics) and serial (RS232) interfaces from our computers. The current USB version is 2.0, but very soon we will have version 3.0 with *SuperSpeed*. Although USB 3.0 devices and computers are expected on the market by the end of 2009 or the beginning of 2010, Windows 7 is not expected to support USB 3.0, at least in its initial version.

USB interface evolution

USB version 1.0 evolved to 1.1 and from there to version 2.0 (**Table 1**). Version 3.0 does not substitute 2.0 but rather complements it. USB 3.0 includes version 2.0 plus a feature called *SuperSpeed* (for practical purposes, a data transfer speed of 400 MBytes/s is expected). The impressive speed of USB 3.0 will help to rapidly transfer large amounts of data in devices like hard disks and high definition video cameras released on the market in the near future.
Another important advantage of USB 3.0 that it supports higher current supplied to devices ('Bus Power'). In addition, a computer with USB 3.0 will have complete USB 2.0 support and SuperSpeed on top of that. If we connect a

3.0 device we have SuperSpeed. If a 2.0 device is connected, the speed would be either Low Speed, Full Speed or High Speed.

USB 3.0 architecture

The USB architecture is arranged by *tiers*. **Figure 2** shows the first tier ('root') at the top of the architecture and below we find tiers 2, 3, etc.

At the first tier we only find the *host* (physically allocated to the computer) which is the bus controller having several *downstream ports* (DS PORT) to connect the USB *hubs* and the USB *peripherals*.

Peripherals are the USB devices (printer, hard disk, etc.) and they have an input *upstream* port (US PORT). Each peripheral can have more than one *device* internally. Devices are marked by *data origin* and *data destination*, and the transfer is between the host and the logical *function* or *functions* of each device along the *function interfaces*. We can think of a keyboard with associated card reader; this would be the peripheral. Inside this peripheral we have two devices: keyboard and card reader. In a logic-driven way, the host will communicate with the keyboard function and the card reader function by means of their interfaces.

Table 1. USB interface evolution

Version	USB 1.0	USB 1.1	USB 2.0	USB 3.0
Date	1996 / 01	1998 / 09	2000 / 04	2008 / 11
Speed	Low Speed 1.5 Mbits/s	Full Speed 12 Mbits/s	High Speed 480 Mbits/s	Super-Speed 5 Gbits/s

The hubs have an upstream input port which turns to the host or to a hub output. The hubs also have several *downstream* output ports to expand the BUS (they are the centre of the USB architecture stars). The hubs are situated on lower USB architecture tiers (tier 2, tier 3, etc.). They are special peripherals.

The terms 'port input' and 'port output' should be taken to refer to the position in the architecture since data can travel in both directions, i.e., upward and downward, via any port.
The number of devices is 127 maximum, while up to 5 hubs can be inserted between the host and a device. For this reason, on the last tier, # 7, there can only be devices but no hub. Upstream and downstream port connectors are different to avoid connection mistakes.

With USB 3.0 compatibility is guaranteed with previous releases thanks to its double bus architecture. This way, there is the possibility to run at Superspeed alongside 'older' speeds (like for USB 2.0). In **Figure 3**, an example is shown which includes a 3.0 host, a 3.0 hub and a two peripheral functions, one USB 2.0 and another USB 3.0. The topology of the double bus is also shown.

SuperSpeed and non-SuperSpeed (USB 2.0) connections are physically comprised together in the USB 3.0 cable.
A definite improvement in the USB 3.0 standards is that the dataflow heads to a correct device only, while USB 2.0 used to distribute the information across the entire bus.

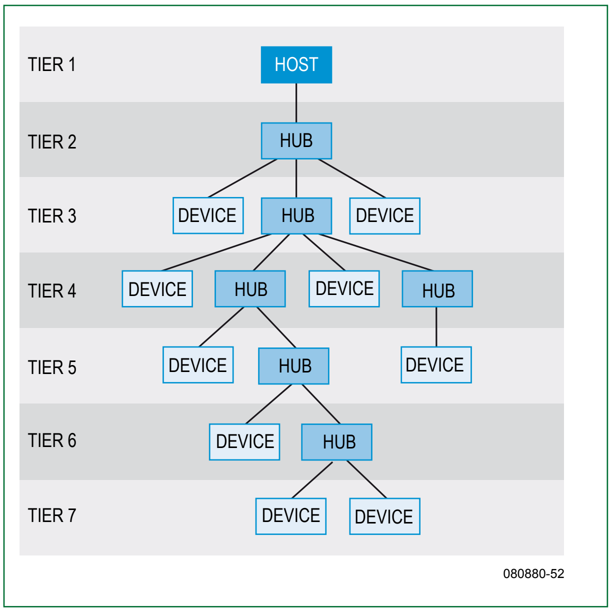


Figure 2.
USB 3.0 architecture.

The USB 3.0 host

For compatibility the 3.0 host (**Figure 4** and **Table 2**) comprises one SuperSpeed host and another non-SuperSpeed host (i.e. USB 2.0). Therefore the 3.0 USB bus can work simultaneously at SuperSpeed and non-SuperSpeed (USB 2.0).

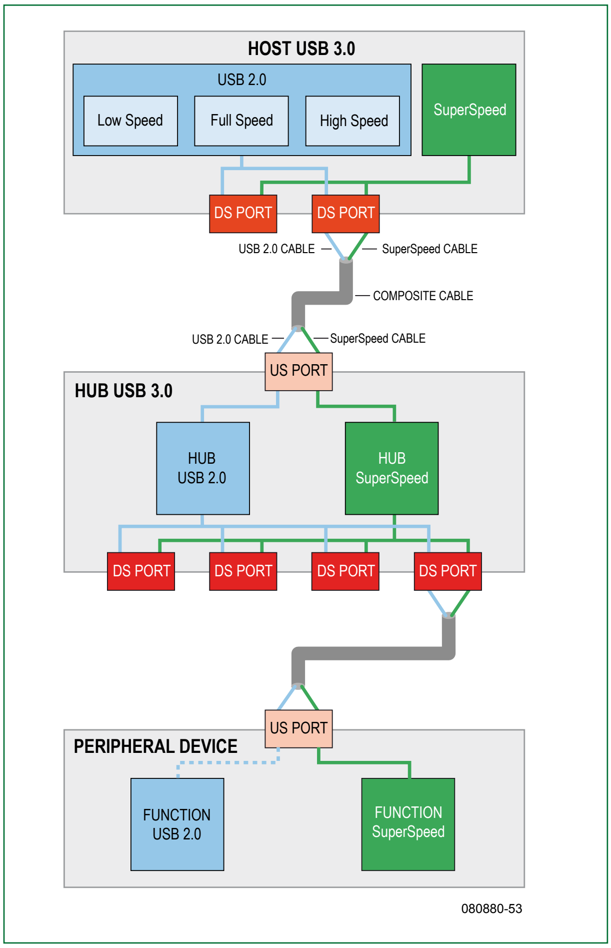


Figure 3.
The example shows the double bus topology.

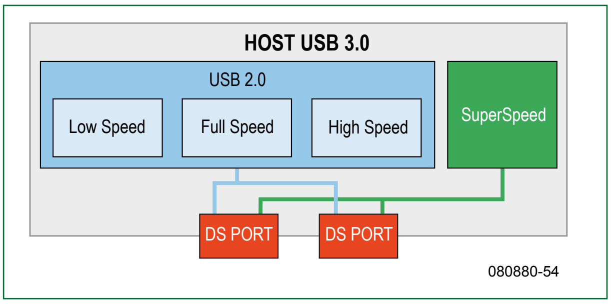


Figure 4.
USB 3.0 host.

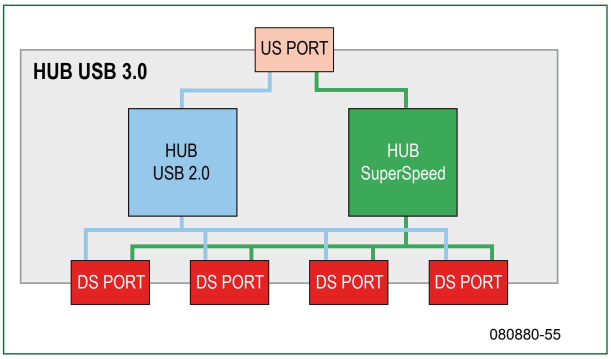


Figure 5.
USB 3.0 hub.

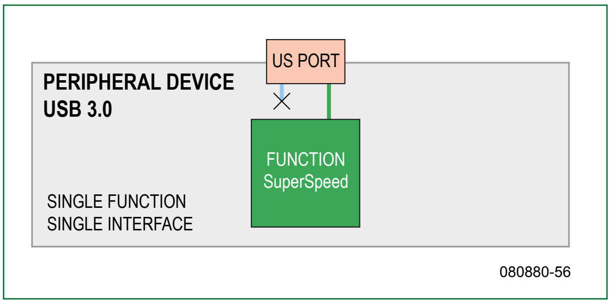


Figure 6.
USB 3.0 peripheral device

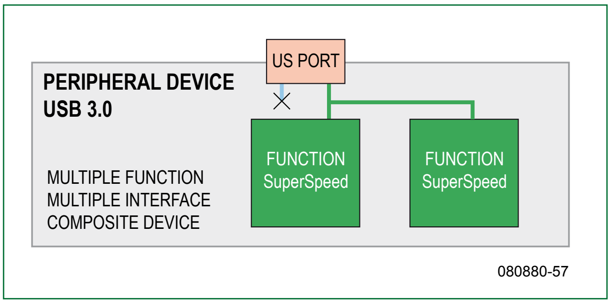


Figure 7.
USB composite device.

Table 2.

USB 3.0 Host	Meaning	Connection direction	Where can it be connected?
DS PORT	Downstream Port	Downward	Hub (US Port) Peripheral (US Port)

Table 3.

USB 3.0 Host	Meaning	Connection direction	Where can it be connected?
US port	Upstream Port	Upward	Hub (DS Port) Host (DS Port)
DS port	Downstream Port	Downward	Hub (US Port) Peripheral (US Port)

Table 4.

USB 3.0 Host	Meaning	Connection direction	Where can it be connected?
US port	Upstream Port	Upward	Hub (DS Port) Host (DS Port)

The USB 3.0 hub

Figure 5 and **Table 3** show that a 3.0 hub actually has two hubs inside, one for USB 2.0 and another for SuperSpeed. That way, the USB bus expansion obtained from the use of hub is compatible with SuperSpeed and non-SuperSpeed (USB 2.0).

The USB 3.0 peripheral

Peripherals with a single device can consist of one or more topologies (just one function, or several):

- a peripheral with a single function and a single interface constitutes a *single device* (**Figure 6** and **Table 4**).
- a peripheral with multiple functions and multiple interfaces constitutes a *composite device* (**Figure 7**).

There are also peripherals with more than one device. These devices are permanently connected to an integrated hub:

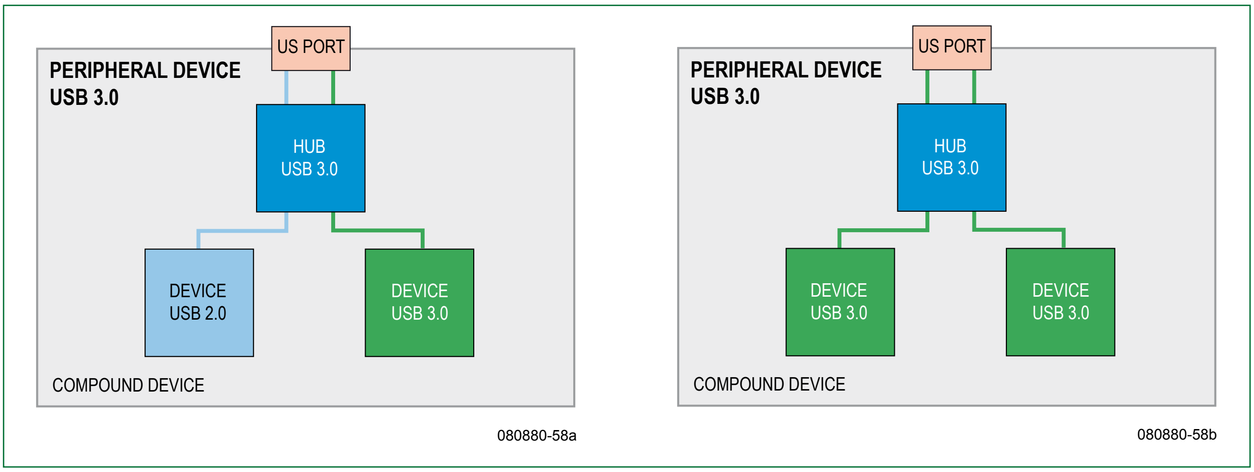


Figure 8.
USB compound device.

Table 5. USB 3.0 Cable		
Letter	**Comment**	**Conductor Name**
A	Jacket	–
B	Braid	Shield
C	Filler	–
D	UTP (unshielded twisted pair) signal differential pair	UTP_D-, UTP_D+
E	2 x SDP (shielded differential pair)	First pair: SDP1–, SDP1+ Second pair: SDP2–, SDP2+
F	Power	PWR
G	Ground	GND_PWRrt

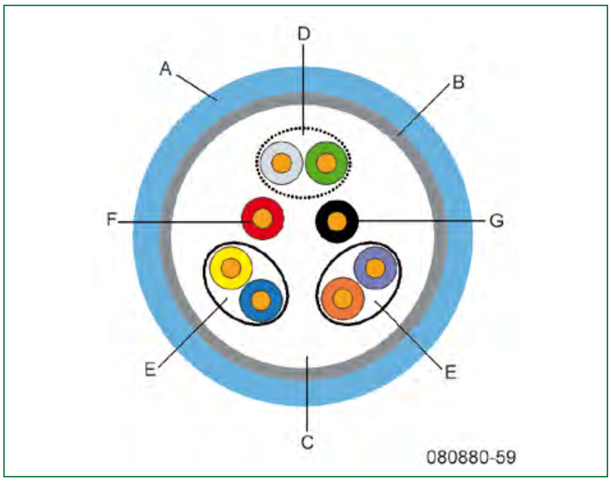


Figure 9.
USB 3.0 cable section.

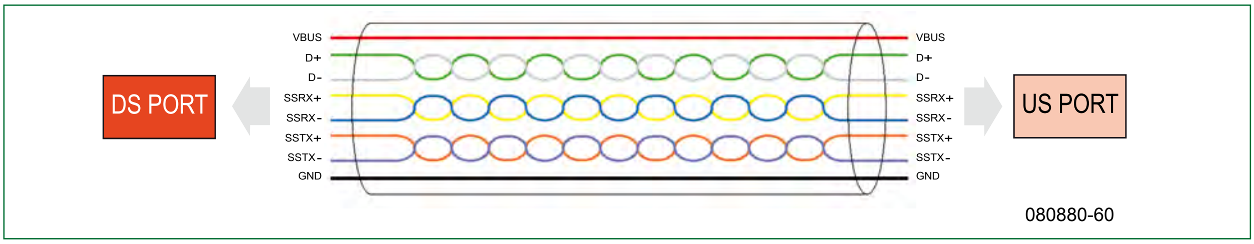


Figure 10.
USB 3.0 cable.

- a peripheral with multiple devices constitutes a *compound device* (**Figure 8**).

If a peripheral internally comprises a USB 3.0 function alongside a 2.0 function, these functions will not be able to function simultaneously.

USB 3.0 wires and connectors

USB 3.0 wiring has all the USB 2.0 conductors plus new ones for SuperSpeed — see **Figure 9 and Table 5**. The UTP pair belongs to USB 2.0 and the two SDP pairs belong to SuperSpeed**.** The positive and the negative wires already existed in USB 2.0 version and remain unchanged in version 3.0 (**Figure 10**).

The UTP pair allows *half-duplex* transmission: however the two pairs together allow *dual-simplex*, which is a great advantage. Data traffic can exist simultaneously in both directions. The nominal differential voltage on both SDP data pairs is 1 V_{pp}.

The USB 3.0 cable reveals a plug on each side. Depending on the plugs used, five different cable types exist (**Table 6**).

It can be seen that there are only two types of plug, A and B. Type A connects upwards (i.e. to DS PORT) and B, downwards (i.e. to US PORT). Besides, there is the standard size and 'micro'.

All the cable types, except for one, have an 'A' type plug at one end and a 'B' on the other, to connect the computer (host) with the peripherals or the hubs, and the hubs with other hubs or peripherals. The exception is the cable with an A type plug at both ends (it becomes something like a crossover cable). According to the USB 3.0 standards, this cable will be useful to connect one host with another (computer-to-computer link). This is something new because under USB 2.0, only two computers could be interconnected

Table 6. USB 3.0 cable types	
Type and size Plug 1	**Type and size Plug 2**
A–Standard	B–Standard
A–Standard	A–Standard
A–Standard	B–Micro
A–Micro	B–Micro
A–Micro	B–Standard

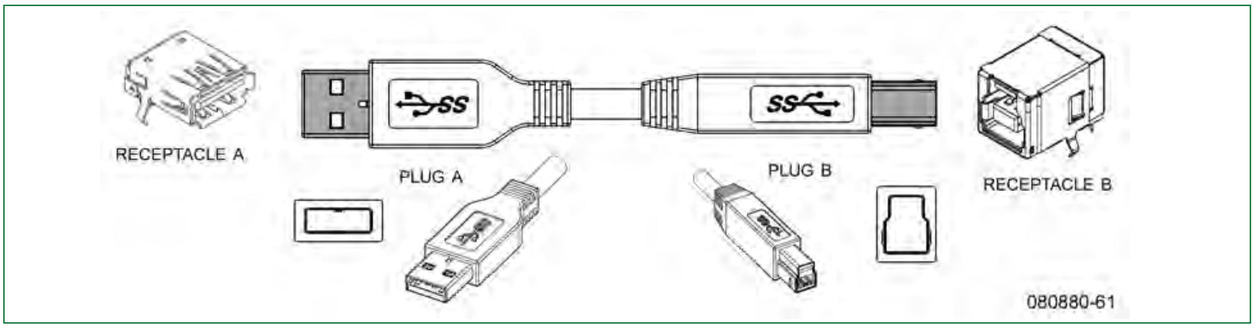


Figure 11.
USB 3.0 cable and connectors.

Figure 12.
The SuperSpeed logo.

by such a link (**Figure 11**).
The USB 3.0 standards mention that cables must fulfill predetermined electrical specifications — based on these, the maximum cable length can be assumed to be about 3 meters (10 feet).
The USB 3.0 plugs and sockets (receptacles) are different from USB 2.0 ones (though similar), due to more wires being connected. The cable plugs must be connected to the sockets.
The sockets can be A or B type, and standard or micro sized. We will typically find the A type in the computer (host) and in the hub output. The B type will be located in the peripherals and on hub inputs. The A or B type sockets allow both USB 3.0 and USB 2.0 cables to be connected (in the latter case, without SuperSpeed).

USB 3.0 bus power

As will be generally known to Elektor readers, the USB interface powers peripheral devices connected to it with a nominal 5 VDC (4 VDC minimum), see **Table 7**.
The USB standard employs the term *unit load* to express the amount of current carried:

- **USB 2.0**: one unit load equals 100 mA. If the current demand remains under one unit load, the current supply is guaranteed and it is a *low-current peripheral*. If it is higher — up to 5 unit loads (500 mA) — it is a *high-current peripheral***.** The host will determine if the bus is able to deliver that current, and will communicate status to the peripheral.
- **USB 3.0**: one unit load is 150 mA. If the current demand remains under one unit load, the current supply is guaranteed and it is a *low -current peripheral*. If it is higher — up to 6 unit loads (900 mA) — it is a *high-current peripheral***.** The host will determine if the bus is able to deliver that current, and will communicate status to the peripheral.

Table 7. Bus Power current

Type	unit load		mA	
	minimum	maximum	minimum	maximum
USB 2.0	1	5	100	500
USB 3.0	1	6	150	900

USB version 3.0 increases the supply capacity for peripherals, allowing many of these to rely on external supply, being powered directly from the USB bus. If a peripheral needs even more current, it has to provide its own power source, internally or externally ('self powered').

More about power

USB 2.0 or 3.0 can power peripheral devices (positive: VBUS, negative: GND), but in the USB 3.0 regulation there is an important novelty: a peripheral device can forward supply power to other elements. For that, the peripheral device uses a specially powered type B socket which includes all known signals (positive, negative, one UTP pair and two SDP pairs) but adds two new ones: *dpwr* and *dgnd*, with a 5 VDC nominal voltage between them and 1 A DC maximum current draw. This new supply is delivered by the peripheral device, not by the bus. If, for instance, we have a USB 3.0 printer with a powered B socket, instead of connecting with a cable to the USB bus (to the host or to some output from some hub), it will be possible to connect it to a wireless USB adaptor and power it at the same time. In this way the USB adaptor will receive its supply voltage from the printer and does not need another source. As a matter of course, the USB adaptor will have a powered B plug (i.e. a normal powered plug with two more terminals, to take the dpwr and dgnd lines presented by the printer socket).

'Pipe', 'endpoint', 'transaction'

A few words about USB lingo used by experts:

- *Pipe*: virtual data path between host and endpoint.
- *Endpoint*: the destination of each pipe. These are memory buffers to store multiples bytes. Physically they usually are memory registers or mere positions inside the devices. Endpoints are numbered from 0 to 15 (EP0, EP1, EP2, … EP15) each being an input or an output, from the host's point of view. All of them are optional, except EP0. EP0's input or output is used to access device configuration data. Since EP0 always exists, a pipe also exists between EP0 and the host called *default control pipe*.
- *Transaction*: this refers to the data exchange between the host and the endpoint from the device across the pipes. The endpoints, besides being an input or an output, are also classified as: *control*, *bulk*, *interrupt or isochronous*, giving rise to four types of transaction as summarized **Table 8**.

If a device input endpoint has to send data to the host, it notifies the host and starts the transaction. This protocol is called *asynchronous traffic flow* — new and better than the *polled traffic flow* within USB 2.0: the host is periodi-

Table 8.

Transactions	Control	Bulk	Interrupt	Isochronous
Typical use	Configuration	Printer Scanner	Keyboard Mouse	Audio Video
Essential	Yes	No	No	No
Corrective	Yes	Yes	Yes	No
Data flow directione	Bidirectional	I or O	I or O	I or O
Message type	Control	Data	Data	Data

cally checking if some device wants to send data to it. The poll is a worse technique because it adds to the traffic on the bus.

Descriptors on USB 3.0

The devices have certain *descriptors*, i.e. data lists which the host uses to configure and manage devices. The most important are:

- *Device descriptor*: there is only one. It includes general information from the device's VIP/PID (number pairs that identify the device) and number of different configuration ways shown.
- *Configuration descriptor*: one for each way of configuring the device. It includes specific information about the device, number of interfaces and maximum current consumption (in 8 mA increments). Remember that data transfer occurs between the host and the device function interfaces.
- *Interface descriptor*: one for each interface. It basically contains the number of Superspeed and non-Superspeed endpoints of this interface. In practice, each interface is a collection of virtual data paths (pipes), one for each endpoint.
- *Endpoint descriptor*: one for each SuperSpeed or non-SuperSpeed endpoint, it describes if it is input or output, and the kind of transaction (control, bulk, interrupt or isochronous), etc.
- *Superspeed endpoint companion descriptor*: one for each SuperSpeed endpoint. Specific to USB version 3.0: former descriptors are also found in USB 2.0.

Device enumeration under USB 3.0

A process called *enumeration* is launched when connecting a device to the USB bus. Several things start to happen:

- *Default Control pipe* is established between the host and endpoint 0 on the device. For the moment, the maximum current a device is allowed to draw is one unit load (150 mA).
- The host allocates an address to the device (1 to 127).
- The host reads the descriptor device by means of the default control pipe to know the VIP/PID and the number of possible configurations the device offers.
- The host reads the configuration descriptor by means of the default control predetermined pipe. It will have as many configuration descriptors as possible configurations offered by the device. The rest of the descriptors (interface, endpoint and Superspeed endpoint companion) are also read at that time as they are associated with the configuration descriptor. Using all this information, the host configures all the endpoint and establishes all the pipes.

According to the data read from the device, the host can respond to the device's supply current requirement, ranging from one unit load (150 mA) to 6 (900 mA).

(080880-I)

Internet Link

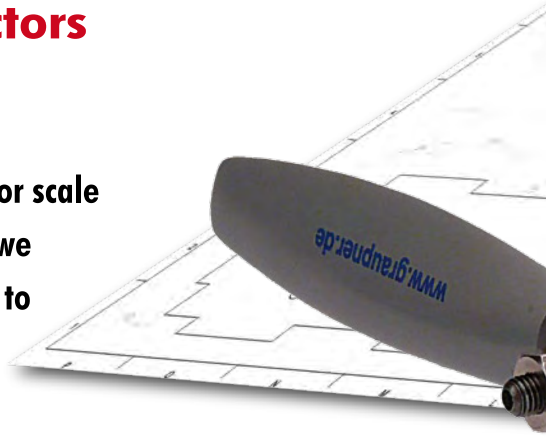
[1] www.usb.org

Automatic Running-in

for internal combustion model engines

Part 2: the test bench, actuators and detectors

Michel Kuenemann (France)



Last month, we began constructing a running-in bench for i/c engines for scale models with the description and wiring of the electronics boards. Now we need to build a chassis capable of housing our new boards, the engine to be run-in, and all the essential accessories.

Building the boards led us to make intensive use of the soldering iron and measuring instruments in our electronics lab.This month, the saw, drill, and screwdrivers are coming to the fore. To get the best out of these boards, it's vital to have a bench that is perfectly suited to this very specific activity of running-in model engines.

Throughout this article we'll be guiding you step-by-step through building your version of the bench. We have checked that the components used are readily available commercially. Most of them can be replaced without any problem by equivalents, depending on what you may already have, and your needs.

After describing the chassis of the bench, we'll tackle fitting, testing, and adjusting all the bench's actuators and detectors.

Before getting down to things, we strongly advise you to read the inset about the basic precautions relating to using model i/c engines.

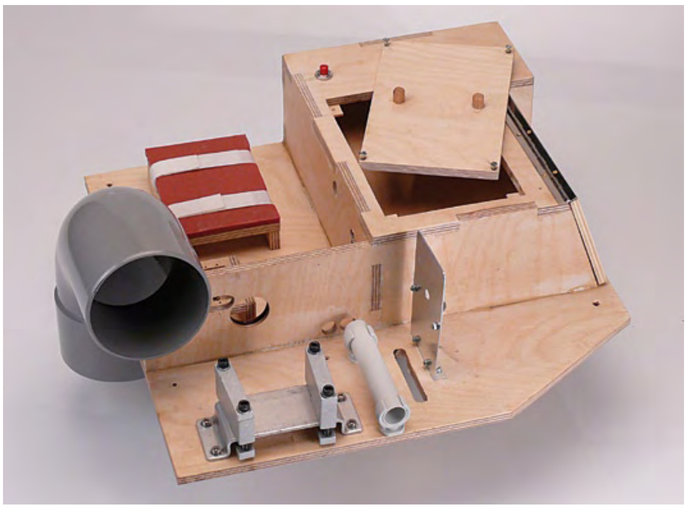


Figure 1. The prototype of our bench, without fittings.

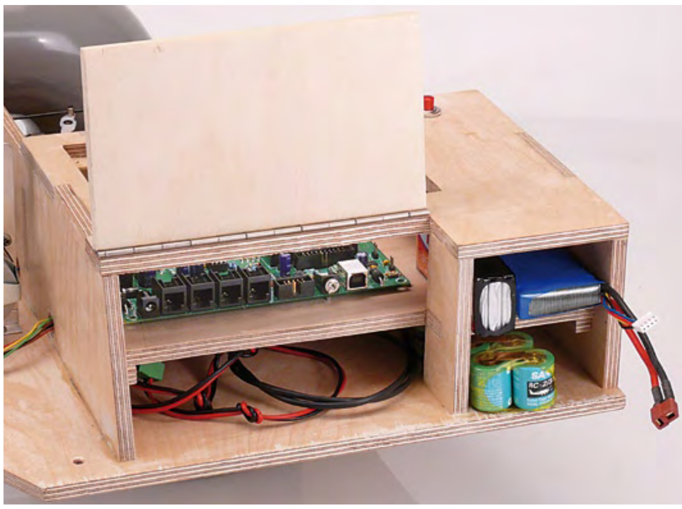
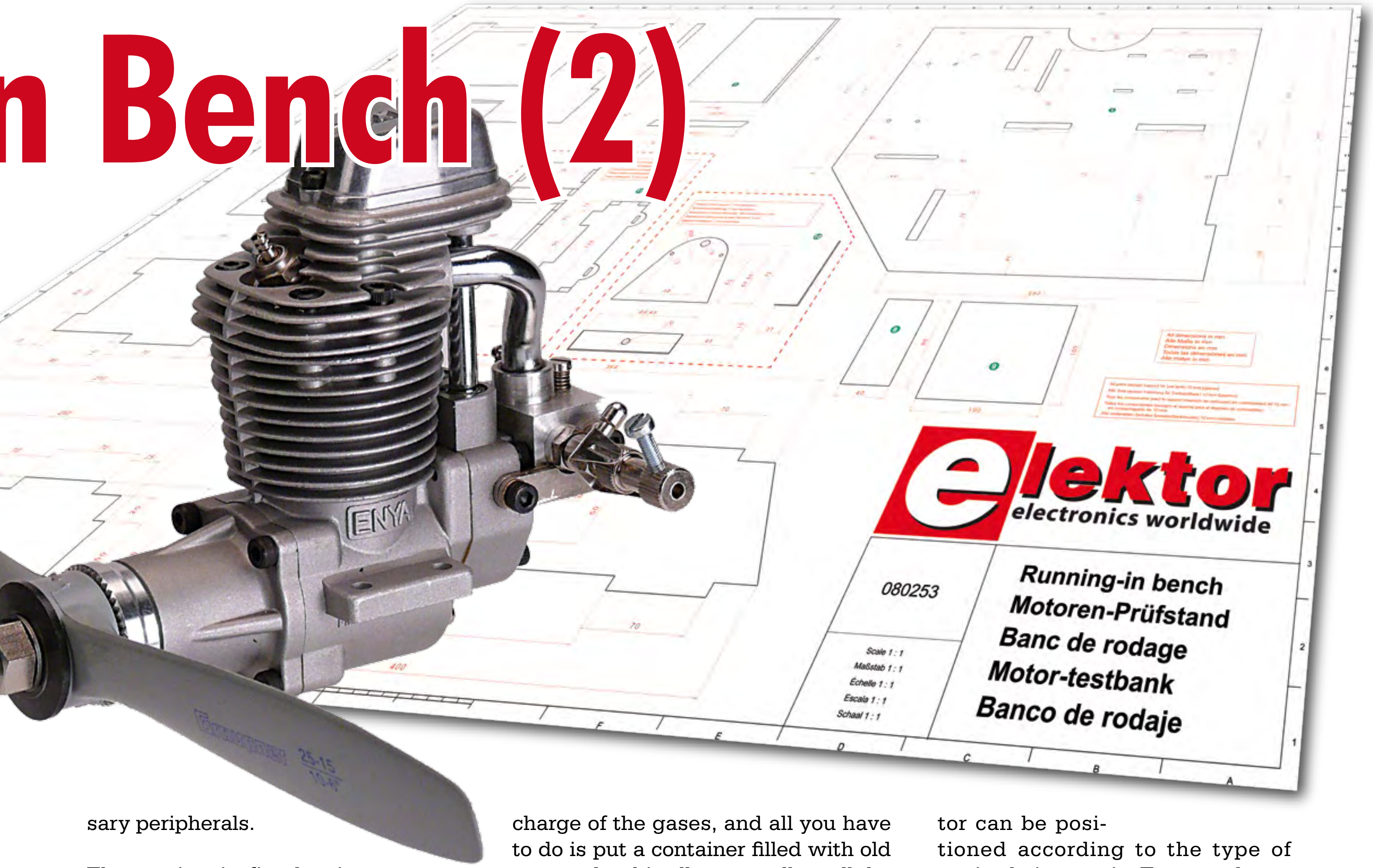


Figure 2. The compartments for the electronics, batteries, and cables.

The bench…

…has been specially designed for our application by experienced model enthusiasts (**Figure 1**) — the plans for this bench are available for download [1]. The base of the bench, the chassis, is made entirely from 10 mm plywood. The bench is compact enough that you can put it away on a shelf between two running-in sessions. This chassis takes the engine, the fuel tank, the electronics boards, and all the neces-

n Bench (2)



sary peripherals.

The engine is fixed using a robust aluminium mount specially designed for the purpose. You'll have no problem finding this sort of accessory in model shops or on the Internet. This mount is able to take most single-cylinder engines up to 20 cc. It is of course possible to make do without a mount of this type, but in all cases, make sure you have a solid fixing, and secure the engine fixings with thread locking compound (Loctite, etc.) or using self-locking nuts.

The fuel tank is slightly raised to meet the height requirements (see box). It may be necessary to adapt the fuel tank mount to the dimensions of your particular tank.

The strange plastic elbow on the left of the bench, you'll no doubt have guessed, is used to channel the oil-laden exhaust gases that the engine emits during running-in. This simple, cheap arrangement proved highly effective during our trials. The large diameter of the tube means that it doesn't in any way affect the discharge of the gases, and all you have to do is put a container filled with old rags under this elbow to collect all the oil given off and thus avoid polluting the environment, at the same time making cleaning up after each running-in session a lot easier.

To the right of the engine, there's enough space for the speed detector and for the richness-setting motor. Behind the engine, behind the partition we find a little mount, designed for a 'standard' size servo. This actuator can be positioned according to the type of engine being run-in. Two wood screws are used to fix this mount onto the bench's chassis.

A special two-level compartment is set aside for the electronics boards and possible batteries (**Figure 2**). The CBRM board [2] has its own 'tailor-made' space in the top of this compartment. Connecting the bench's detectors and actuators to the board really is child's play. A few judiciously placed holes, approx. 16 mm diameter, (see photo) let you thread the cables through easily with their connectors. The compartment below the board is designed for storing the cables between sessions. Amongst others, you'll probably want to store a USB cable, a DSC cable, the pocket terminal cable, and maybe a cigar-lighter plug. The second pair of compartments has been designed to hold the batteries that allow the bench to operate in a stand-alone fashion. In the top part, you can fit a 5- or 6-cell NiMH battery, or a 2s or 3s LiPo (Lithium Polymer) battery. A capacity of 1500 mAh

Let's talk safety!

Contrary to what their appearance might lead us to think, model engines are not toys. Their considerable power and the presence of the propeller makes them potentially dangerous, and every year there is a regrettably large number of serious accidents, particularly to people's hands. If you are new to modelling, seek advice and help from an experienced model-maker during your trials. They will be able to guide you and perhaps avoid an accident. If you are an experienced model-maker, but new to electronics, then ask the advice of an experienced electronics hobbyist who will help you build, test, and wire up your bench. And if you are familiar with both fields, then think like two people and check your board and all the electrical connections and mechanical tightening three times before carrying out your first trials. Under no circumstances may either the authors or Elektor be held liable for accidents following use of this board.

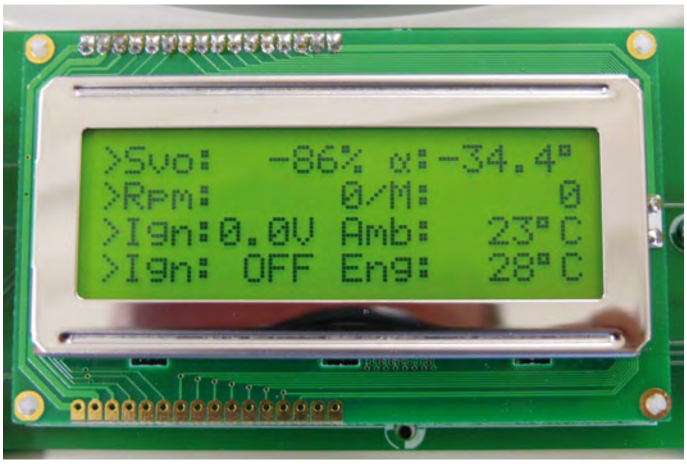


Figure 3. The display once the bench is powered up.

Figure 4. The engine mounted and fitted with a propeller.

is enough to operate the bench for several hours. The battery for heating the glow-plug or for the electronic ignition will go into the bottom compartment. Wrap your batteries in foam to wedge them in place and protect them from engine vibrations, thereby avoiding their falling off the bench.

Once the woodwork is finished, don't forget to apply a coat of cellulose varnish to your chassis to protect the wood from the fuel and the oil it contains. With the help of the photos, drill the holes for the cables, glue on the 100 mm diameter PVC elbow, and fit the engine and fuel tank mounts. Don't fix the servo mount for the time being.

Functional testing software

The next step consists of fitting, connecting up, and then testing the operation of the actuators and detectors. But before starting, 'flash' the CBRMtest_sensors.hex software [2], then fit the board to the bench. Connect up the pocket terminal, power up the board and check that the software starts up correctly. The terminal display should look like **Figure 3**.

The first line of the display shows the current position of the servo. By turning the encoder knob, the servo moves through an angle of 80° (from −100% to +100%). By pressing the push-button alongside the first line, the display changes and the encoder knob now acts on the stepper motor. The first line of the display now shows the current position of the motor and the position set for it to go to. The current position updates as the motor rotates.

The second line of the display permanently shows the engine speed.
The third line permanently shows the board supply voltage and the glow-plug supply voltage. As a safety precaution, if the board supply voltage drops below 6 V, the servo is set to the 0% position and the program hangs in this condition.

The start of the fourth line indicates the state of the glow-plug: on or off. The glow-plug can be turned on or off by operating the push-button alongside the fourth line. The end of the line shows the engine temperature.

Fitting the engine to be run-in

When seen from the front, most engines have their throttle controls on the left and the richness screw (or needle-valve) on the right. If this is not the case, it may be possible to turn the carburettor to achieve this situation. Fix the engine firmly to the mount, taking care that the exhaust comes within the discharge elbow (**Figure 4**). When everything is properly in place, connect a

Figure 5. The throttle servo in its mount.

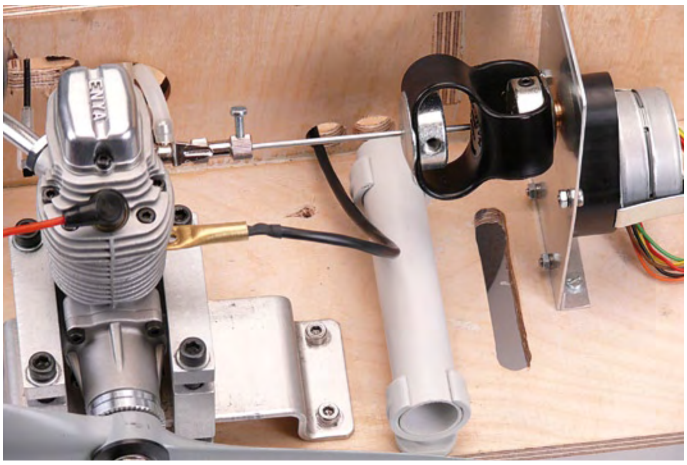


Figure 6. Connecting the stepper motor.

2 mm clevis with its rod to the throttle control and feed it through the oblong opening you've made in the partition behind the engine. It will also be very helpful to fit your engine with a new, good-quality glow-plug. You'll fit the propeller and its cone just at the moment of starting the tests.

Throttle control

Fit the servo into its mount using the rubber grommets and spacers supplied with the servo (rounded part downwards. Fit a piece of 'choc-block' to the servo rod (see **Figure 5**). This is probably the most effective and practical way of connecting the control rod to the throttle servo. Position and fix the mount in such a way that all the moving elements are correctly aligned and operate without any tight spots. Connect the servo to the board via K5, then set it to the neutral position (i.e. 0%) using the pocket terminal encoder. Position the servo rod in such a way that it is perpendicular to the servo, then fit the rod screw.

By turning the encoder knob, you'll be able to check that the control functions gently and operates over the whole travel of the throttle control for a control variation between approx. −100% and +100%. The butterfly valve should open as the control increases, by turning the encoder clockwise. If this is not the case, position the configuration jumper JP4 so as to invert the sense in which the servo acts. Adjust the position of the choc-block on the servo rod to arrive at a suitable travel.

Richness setting

The fuel mixture (richness) screw must be capable of being operated over several turns during the running-in. In this situation, a servo, whose travel is limited to around 120 degrees, i.e. a third of a turn, is not at all suitable for the task. What's more, the richness screw must be adjusted 'carefully and accurately', although the actual speed of this adjustment is not very critical. A single-pole stepper motor with reduction gearing meets these requirements perfectly. The type used, with 2400 steps per revolution, will operate the richness screw of your precious engine gently and accurately. The stepper motor is connected to the CBRM board via six wires. Look at the April 2009 article [2] for details of this connection. Instead of a long-winded explanation, **Figure 6** will guide you in building the bracket and coupling between the richness screw and the stepper motor. Check that the supply jumper JP11 is in the VHV position. Once fitted and connected, test the operation of your project using the encoder knob on the terminal. The motor should turn in the same direction as the encoder. Rotating the encoder anticlockwise should make the stepper motor turn in such a way as to open the needle valve. If this is not the case, correct the motor wiring. If the motor doesn't turn at all, check that jumper JP3 is not fitted. If it is fitted, remove it.

Glow-plug supply

Traditionally, model enthusiasts power their engine's glow-plug by means of a 'glow clip' — a sort of removable connector that works a bit like a syringe. This connector is hooked onto the engine by hand at the moment of starting. This method is perfectly suitable when you're starting before taking a flight or going for a lap of the circuit, depending on whether the model is a plane or a car. But in a running-in situation, the model-maker generally has to start the engine several times.

Repeatedly handling a glow clip close to a moving propeller is not very convenient and certainly dangerous. The CBRM board looks after powering the glow-plug for you, so you can leave the glow clip in place permanently, or replace it, as we have, with a little rubber connector specially intended for the purpose (**Figure 7**). A cheaper alternative to this connector is a simple electrician's choc-block, stripped of its insulation, which will connect the + pole of the glow-plug to its supply cable, which should have a cross-section of around 0.5 mm². Don't forget to connect the engine mount to ground, using an eyelet terminal and wire of the same gauge as the wire to the glow-plug. These two wires will be connected to connector K11 on the board. The glow-plug supply battery (not more than 2 V!) will be connected in the same way to connector K13. Make sure you observe the power source polarity correctly!

A polarity error won't cause any damage, but the glow-plug will be powered all the time, which is **very dangerous**, as the engine may start unex-

Basic precautions

Using i/c model engines does require some basic precautions.

Mechanical mounting of the engine

It is vital to make provision for a sturdy, reliable mechanical mount, as these engines vibrate a lot and produce a tractive force that can reach several tens of newtons. This point is particularly important, as it's not hard to imagine the damage and injuries that an engine fitted with its propeller could cause if it came adrift from its mounting at full speed! ***Don't use vices or G-clamps for holding the engine.***

Fuel supply

The bench's fuel tank must be designed to hold methanol-based fuel and include a pressurization point. The fuel tank must be positioned in such a way that the level of the engine carburettor is half-way up the fuel tank. What's more, you should take care to fit the fuel tank as close as possible to the engine, to minimize the length of piping. An unsatisfactory fuel supply will cause difficulties in starting and erratic running of the engine. This type of fuel tank, holding around 500 ml, is readily available form model shops. They'll also be able to supply the silicone 'hose', the pump for filling the fuel tank, and of course, the right sort of fuel for your dear little gem…

Exhaust

Model engines running on methanol operate with a fuel containing around 20% oil. This oil, mainly unburnt, gets into the exhaust gases. No need to point out that unless certain 'health' precautions are taken, this oil ends up invading everything around the exhaust outlet, with the unpleasant consequences that can be imagined. To make matters worse, during running-in, the engine operates with a very rich mixture, increasing the emission of oil very significantly.

Figure 7. Detail of the glow-plug.

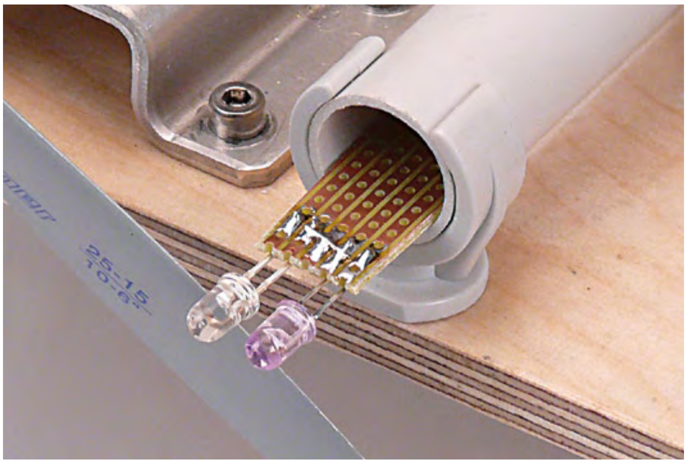


Figure 8. The speed detector, consisting of a phototransistor and an infrared LED inside a PVC tube.

pectedly while you are priming it! Test the proper operation of the glow-plug several times using the pocket terminal. To do this, you can temporarily connect a glow-plug to the connector and ground and check that it glows and goes out clearly according to your commands.

When the glow-plug is on, LED D15 on the CBRM board lights. If the LED doesn't light, check that jumper JP3 is not fitted. If it is fitted, remove it. The glow-plug power system must be totally reliable, or else the running-in sessions will become a real nightmare!

Speed detector

The speed detector consists of a phototransistor and an infrared LED. Fit these two components side by side on a small, rectangular piece of prototype board, just the right width to fit inside the 16 mm PVC tubing (**Figure 8**). Depending on the ambient lighting, the extra (invisible) light provided by the LED may not be needed, or may actually be a nuisance. By crimping both a 3-pin connector and a 2-pin connector to the end of the cable, you can choose whether or not the LED is powered, according to which connector is connected to the CBRM board A 5 cm length of PVC tubing provides effective mechanical protection for this detector.

Once it is in place, wave a sheet of white paper rapidly in front of the detector. The terminal display should indicate a speed of a few hundred RPM, varying. Tip: The 'camera' function of your mobile phone will let you see the infrared light from the LED. If it is not visible in the form of a white dot on the screen of your phone, check the polarities and quality of the wiring to the detector and LED.

Engine temperature detector

The KTY81-210 temperature detector is easy to use, as it comes in a standard 2-pin TO92 package, and is thoughtful enough not to be polarised. After connecting it to a 2-core cable and insulating the joints with heatshrink sleeving, plug the detector onto connector K17 and test it by checking the plausibility of the temperature it shows on the terminal. If you grasp it with your fingers, the temperature indication should change. Then cut off a short length (approx. 3 cm) of 5 mm inside diameter brass tubing. Flatten one end of the tube and drill it with a 3 mm hole. Check that you can easily fix this bit of tube under one of the engine block screws (**Figure 9**) to ensure very good thermal contact. Once the mechanical

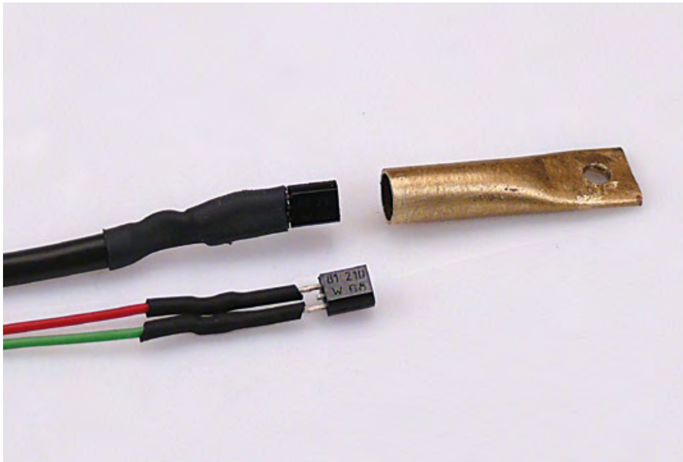


Figure 9. The temperature detector and its brass fixing.

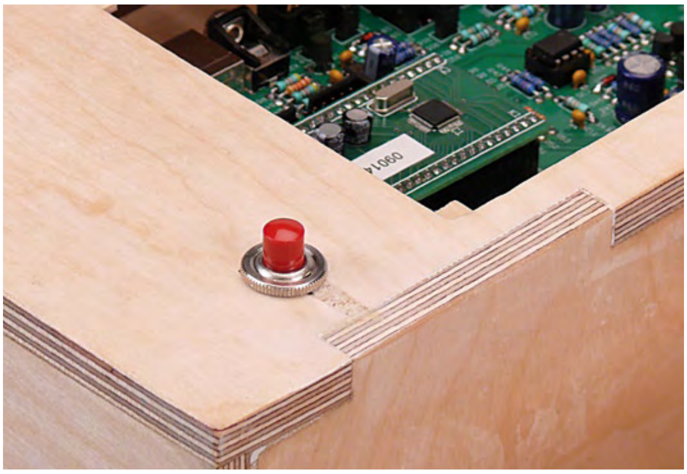


Figure 10. The emergency stop button needs to be accessible!

assembly is finished, insert the detector all the way into the tube and stick it in place with epoxy resin.

Emergency stop push button

It's vital to fit our magnificent bench with a control that will let us shut off the engine throttle control quickly in the event of a problem. Cutting the power to the bench is not a good idea, as the throttle servo will stay in its last position at the moment of losing power and the engine will continue to run.

Figure 11. A solid panel of wood lets you fix the bench onto trestles.

The system we've adopted has the merit of being simple and effective. The emergency stop button (refer to **Figure 10** for fitting) is simply connected in parallel with the CBRM board reset button, via connector K4. As soon as you release the button, the microcontroller will restart, and will lose no time putting the throttle servo into the 'throttle closed' position, thereby stalling the engine. You should use a simple normally-open (NO) push button, sturdy enough to withstand 'beefy' pressing. A locking industrial-type 'emergency stop' button will not be suitable for this use, as it locks into the contact closed position and the microcontroller won't reboot until the button is unlocked, which is unacceptable here. Check that this button works properly before carrying out your first trials with an engine running.

Mounting the bench

We recommend mounting your bench on a solid panel of 19 mm thick chipboard so you can rest the bench on two trestles during your trials. You can use four sturdy G-clamps to hold the bench firmly on the trestles.

First trials with an engine

After you have checked several times that all the detectors and actuators of your new bench operate correctly, we recommend getting used to it by doing some trials with an engine that's already been run-in. The terminal will let you control its settings manually without 'sticking your fingers in'. So handy!

To be continued...

Next month, we'll be rounding off this 3-part article with the description of the automatic program.

(081187-I)

Acknowledgements

The author would like to thank Guillaume and Dominique Dobler for designing and building the mechanical part of the bench.

Internet Links

[1]. www.elektor.com/081187

[2]. www.elektor.com/080253

USB On-the-Go, OLED and capacitive touch pad

Microchip Starter kit for the PIC24F family

Clemens Valens (Elektor France)

The new evaluation board for the PIC24F 16-bit microcontroller family is supplied in a DVD box and has a very comprehensive set of features. It has an integrated programmer and debugger, a capacitive touch pad, a small OLED display, a processor and, above all, no fewer than three USB ports!

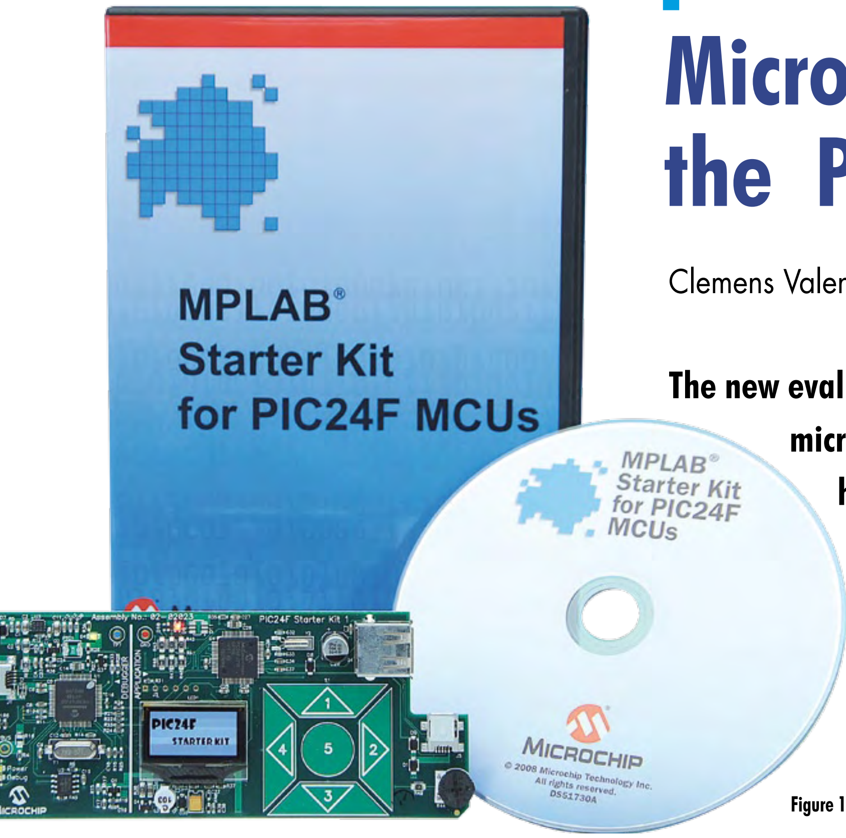


Figure 1. The PIC24F starter kit.

The new starter kit for the PIC24F microcontroller family comprises only a few parts. In the DVD box that contains everything, we find the evaluation board in pink bubble wrap, a USB cable and a CD-ROM. The installation of this kit is just as uncomplicated as its contents, the CD-ROM goes in the CD-ROM drive of a PC and the supplied programs will be installed. The evaluation board is now connected with the USB cable to the PC and the job is complete. One thing you will have to observe, is that there are two mini USB connectors on the board and it is important that you use the correct one. The one you need to use is on the section of the board which is labelled *DEBUGGER*. The CD-ROM supplies the well-known software tools from Microchip, such as the integrated development environment MPLAP IDE and the student version of the C-compiler MPLAB C30. This version is fully functional for the first 60 days. After that the functionality is limited, but that does not affect the results of the previous 60 days. According to the documentation, the evaluation board can only be used in combination with the version of MPLAB (V8.11) supplied on the CD-ROM. In addition, these tools only work under Windows (XP Professional in our case) so that MAC or Linux users will need to use a Windows simulator or emulator.

Hardware

Before we start to experiment, we will take a closer look at the board itself. As already mentioned, this board consists of two parts: a programmer/debugger with (mini) USB port, based on a PIC18F67J50, and the actual application, based around a PIC24FJ256-GB106. We are mostly interested in this second part. Here we find a very small (15 × 25 mm) OLED display with a resolution 128 × 64 pixels, a capacitive touch pad with five buttons, an RGB LED, a potentiometer and a further two USB connectors, a miniature male connector (type mini B) and a standard female connector (type A). These two USB connectors are positioned in such a way that they cannot be used at the same time. The board obtains its power supply either from the USB connector on the debugger part or from the mini-USB connector on the other side.

The most interesting aspect of this board, as you will already have realised, are the USB connections. The microcontroller conforms to the USB 2.0 On-the-Go standard (see **inset**) and the board is supplied with a C library which contains everything you need to develop USB applications (both OTG and standard).

Another interesting part of this board is the capacitive touch pad. The processor has an interface specifically for this type of keyboard built in, which makes the implementation of such a keyboard much simpler.

To complete the board, Microchip have added a small OLED display with

accompanying graphics library (in C), which makes it very quick to implement a graphical user interface.
In fact only an MP3 decoder is missing, otherwise you would have been able to make your own iPod!
The only drawback of this board is that it is not possible to control anything with it. There are no expansion connectors and there is also no breadboard space. But okay, this is really a starter kit and not a development kit.

Software

If the board is connected correctly it will start without problems. Two green LEDs in the debugger part will turn on, the OLED display will initially turn completely white, which is then followed by a welcome message. The three-colour LED (RGB) turns on with such a brightness that it is painful to the eyes. Fortunately it turns off again at the start of the demonstration.

This demonstration, incidentally, is quite impressive. At the top of the initial screen there is a menu with four options and showing the date and time. This menu offers the options of *Flash Drive*, *Utilities*, *Demos* and *Games*, and the navigation is done with the aid of the capacitive touch pad. After selecting the *Flash Drive* option, the board asks you to insert a USB stick. Once the stick has started up, a scroll window appears which contains all the names of the folders and files that are on the stick and it is possible to browse the contents of the USB stick.
The *Utilities* option offers the possibility of setting the date and time (with the + and – buttons), to calibrate the capacitive touch pad (this becomes noticeably more sensitive) and to start a test for the board.
From the *Demo* option there is a choice of three different demonstrations. In the RGB LED screen the brightness of the three colours can be adjusted individually with arrows. The *Graph* demonstration shows a moving curve, which can be influenced using the potentiometer on the board. The speed of the scrolling movement can be adjusted with the capacitive touch pad. The *Capture* demonstration looks a bit like *Graph*, except that the curve does not move and the values are stored on the USB stick. Unfortunately this demo stops after a few times, the file that is created on the USB stick is empty and cannot be deleted, even when using a PC. Another USB stick, 8 GB in size, which already contained a few files, caused an error message *"cannot open file"*.
Finally, the Games option in the main menu offers three intuitive graphical games: *ShuBox*, *Shapelet* and *Blaster*. The complete source code for the demonstration program is available on the CD-ROM.
On the CD-ROM we also find the *MPLAB Starter Kit for PIC24F MCUs*, which during installation splits into MPASM Suite, MPLAB ASM30, MPLAB C30 and MPLAB IDE. In addition a folder is created with three libraries: graphics, USB and memory card. The source code (in C) is also supplied, as well as the documentation and even a few tools, such as for example one which will convert bitmap-files (BMP) and font files (FNT, TTF and OTF) into hex files that can be used by the graphics library.
After starting MPLAB the board is quickly recognised. The project that allows the demonstration software to be changed and then compiled again is found quickly, and the compilation is without errors. Downloading to the controller takes about ten seconds, after which the board will start up again and run the program.

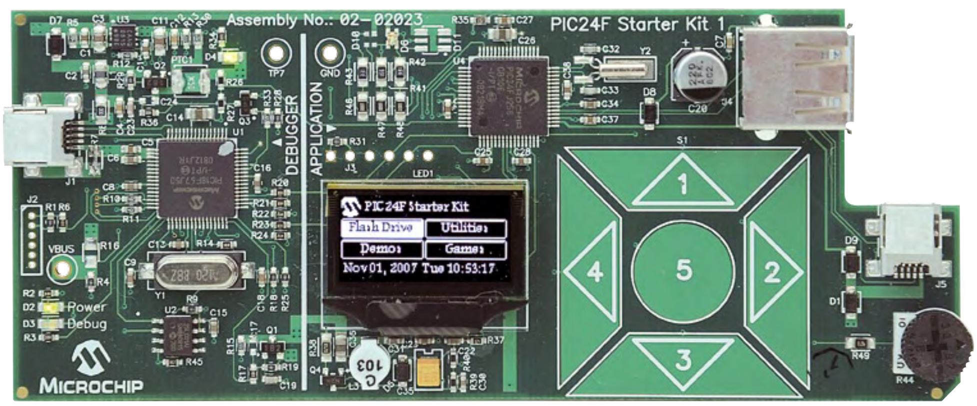


Figure 2. The board in operation, in the middle is the miniature OLED display.

Conclusion

The PIC24F starter kit is very comprehensive, solidly constructed and very quick to get started. Compared to the Explorer 16, also based on a PIC24F and familiar to Elektor readers, this board is much easier to use. That is where the comparison ends however, because these boards are clearly targeted at different applications; The Explorer 16 is intended more for electronics engineers while the board described here is mainly for software engineers.

(080927-I)

Technical Features PIC24F Starter kit DM240011

- **Microcontroller PIC24FJ256-GB106**
- **USB 2.0 OTG**
- **256 KB flash memory**
- **16 KB RAM**
- **OLED display 128x64 pixels**
- **Capacitive touch pad**
- **RGB LED**
- **Potentiometer**
- **Integrated programmer/debugger**
- **Includes MPLAB, ASM30, C30 and source code**

Internet Links

www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en535092

USB On-the-Go (OTG)

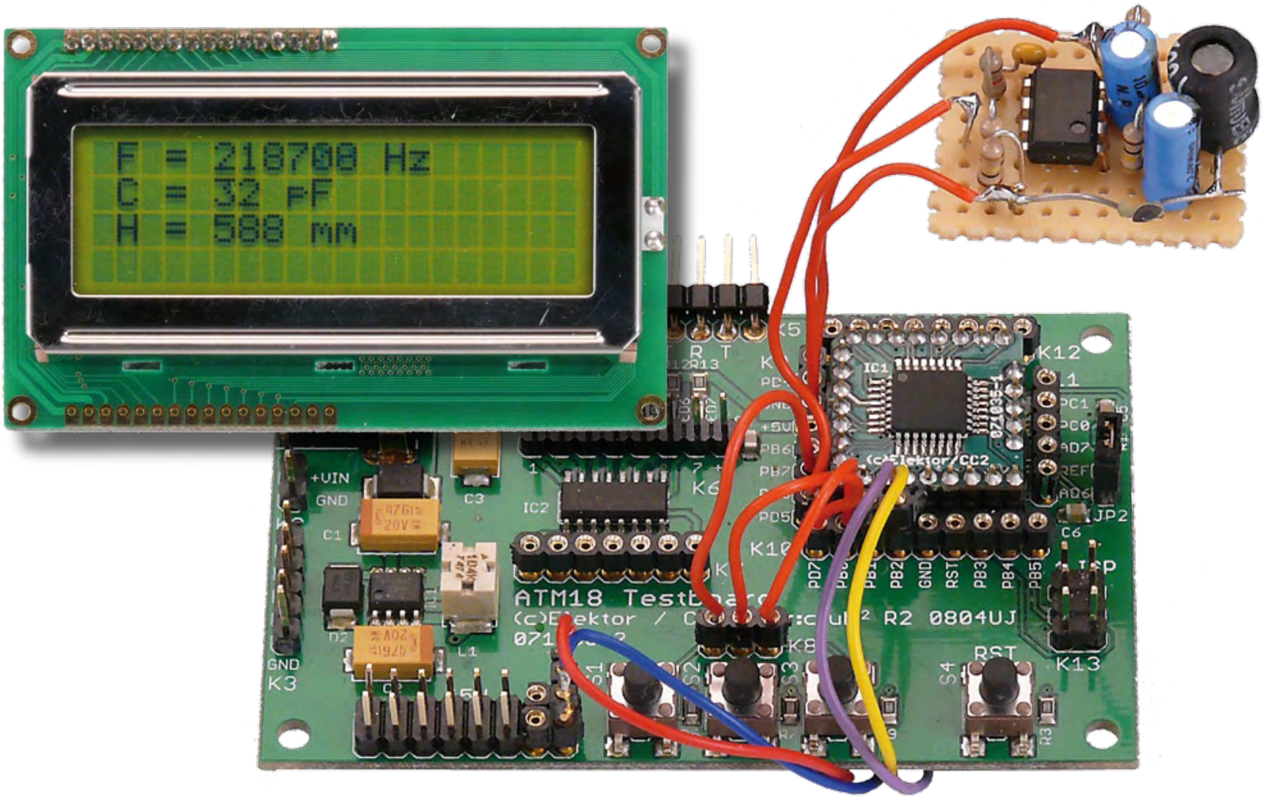
USB On-the-Go is an extension to the USB standard. A device that conforms to this standard can take the function of both a USB-host as well as a USB-slave, and can change between these two while in use. Two USB-OTG compatible devices can communicate with each other without the need for a separate host. This is the case, for example, with digital cameras that can send pictures directly to a printer, or store directly to a hard disk.

Brim Full

Capacitive liquid-level measurement

Wolfgang Rudolph (Germany), Rudolf Pretzenbacher (Austria), and Burkhard Kainka (Germany)

Electronics enthusiasts are sometimes a breed apart. Most people simply look at a bottle when they want to know how full it is, but we want to measure it.



Of course, it doesn't have to be a bottle. Situations that involve measuring the level of a liquid stir the creative juices and foster true acts of genius, and there are countless applications for liquid-level sensors, ranging from rain barrels to heating-oil tanks.

We're sure that our readers can come up with many other situations where the liquid-level sensor described here can be put to good use. However, let's first consider the question of how to measure a liquid level accurately and reliably.

Measuring methods

A wide variety of measuring methods are used. Many lavatory cisterns have a float valve that first reduces the inflow of water when the float rises to a certain level and finally stops it completely. In

Table 1

Inductor specifications

(vertical package
with moderate rated current)

Manufacturer: Fastron;
type number 09 P-103 J-50

Dimensions: Ø 9.5 mm, height 14 mm,
lead pitch 5 mm

Inductance: 10.0 mH (at 20 kHz)

Self-resonant frequency (SRF): 0.41 MHz

Rated DC current: 90 mA

Resistance: 35.0 Ω

Tolerance: ±5 %

Q (min): 70

this case, the float is not only the sensor but also the actuator, which controls the valve via a lever mechanism. Although this is a very reliable principle, it can't be used to measure the liquid level. The same principle was used in the past (and is sometimes still used) to measure the fuel level in petrol tanks of cars. In this case, the float moves the wiper of a potentiometer instead of actuating a valve. This variable resistance forms part of a voltage divider that drives a milliammeter, which indicates how full the tank is. In some cases, the accuracy of this gauge leaves a lot to be desired.

Nowadays a wide variety of modern measuring methods are used in many different situations. They include hydrostatic and differential pressure measurement, conductivity measurement, light absorption measurement, transit time measurement using ultrasound, distance measurement using microwaves, and even transit time measurement using radar pulses.

From an electronic perspective, capacitive measurement is also interesting. This method involves measuring the change in the capacitance between two electrodes. If these electrodes are located in a container with a liquid that covers them more or less depending on its level, the capacitance of this 'capacitor' changes accordingly. The capacitance depends on the dielectric constant ε of the liquid, and it increases as the level of the liquid rises.

Capacitive sensing

You've probably guessed that this is the method we intend to use here. After all, we're used to working with capacitors. However, it's not as simple as it seems at first glance. We have to do a bit of maths first. This article is based on a capacitive liquid-level sensor built by Rudolf Pretzenbacher, which uses a simple but remarkably stable oscillator for the sensor circuit and an AVR microcontroller for the signal processing. His liquid-level gauge provided the inspiration for this ATM18 article, and it delivers truly astounding results. This setup can be used to measure capacitances in the range of nanofarads (nF) to femtofarads (fF). In case you've forgotten, a femtofarad is 10^{-15} F or a thousandth of a picofarad. How can such high sensitivity be achieved? The answer is that the 'sense capacitor' in the liquid is one of the frequency-determining components of a resonant loop, which in turn is part of an oscillator circuit. If an object to be measured is brought in the vicinity of the capacitor, the resonant frequency of the loop changes. The more the capacitance of the capacitor is increased by the object, the lower the resulting frequency. The task of the microcontroller on the Elektor ATM18 board is to measure the frequency and then calculate the value of the capacitance from the measured frequency and the known value of the inductance.

This sounds quite simple, but there are still a few details to be sorted out.

Oscillator

The oscillator circuit can affect the resonant loop due to its own capacitance or as a result of excessively strong coupling. To keep this effect as small as possible, the resonant loop should have a high quality factor (Q) and the excitation level should be kept low. It is also important to choose a suitable inductor.

In this case, we decided on a fixed inductor made by Fastron. This inductor (type number 09 P-103 J-50; available from Reichelt and other sources) has an inductance of 10 mH, a DC resistance of 35 Ω, and a self-resonant frequency of 410 kHz. This means that it has a remarkably low stray capacitance of 15 pF. In addition, it has a specified Q factor of 70 (max.). Its characteristics are listed in **Table 1.**

The higher the Q factor of a resonant loop, the lower its damping. A Q factor of 70 means that the amplitude of a 'free' (damped) oscillation is reduced by a factor of e after 70 cycles, which can be seen very nicely on an oscilloscope. The damping results from the resistive losses in the wire and the magnetic losses in the core. A resonant loop with an inductance of 10 mH and a capacitance of 6300 pF has a resonant frequency of 20 kHz, and the inductive and capacitive impedance are both 1260 Ω. The ratio of this impedance to the DC resistance (35 Ω) yields a theoretical Q factor of 36, which means that the resonant impedance of the circuit is 45 kΩ (1260 Ω × 36). The Q factor and the resonant impedance increase as the capacitance is reduced and the frequency rises. For a high Q factor, we have to aim for a high L/C ratio. At around 3000 pF and 30 kHz, the calculated value of the Q factor is approximately 70. The core losses increase at very high frequencies, which causes the Q factor to drop. However, the oscillator circuit has an even larger effect, since a resonant loop with a high resonant impedance is especially sensitive to external influences.

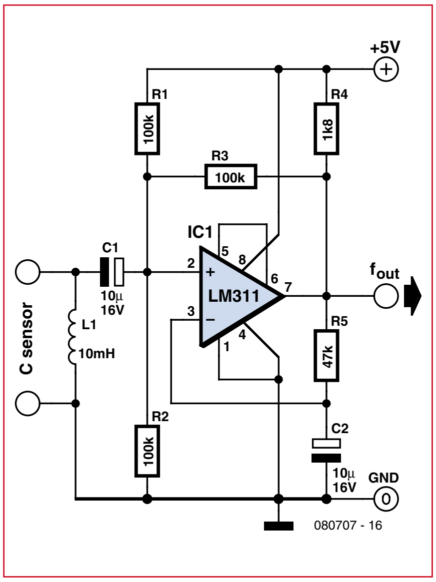


Figure 1. Schematic diagram of the oscillator used for capacitance measurement.

Figure 1 shows the oscillator circuit used here, which is built around an LM311 comparator. It compares the input voltage with a reference voltage and converts the sinusoidal signal from the resonant loop into a squarewave signal at its output. This signal excites the resonant loop via a feedback resistor. A voltage divider at the non-inverting input of the comparator provides a voltage equal to half the supply voltage. The inverting input is fed by a comparison voltage obtained by integrating the output voltage. As a

Listing 1

Capacitance measurement

```
Config Timer0 = Timer ,
   Prescale = 64
Config Timer1 = Counter ,
   Edge = Falling , Prescale
   = 1
On Ovf0 Tim0_isr
On Ovf1 Tim1_isr
Enable Timer0
Enable Timer1

Do
  Ticks = 0
  Enable Interrupts
  Waitms 1100
  Disable Interrupts
  Lcdpos = 2 : Lcdline = 1 :
   Lcd_pos
  Lcdtext = "Freq = "
  Lcdtext = Lcdtext +
   Str(freq)
  Lcdtext = Lcdtext + " Hz
   "
  Lcd_text
  Print Freq;
  Print " Hz"
  C = Freq / 10000000
  C = 1 / C
  C = C * C
  C = C / 39.48
  If Pinb.0 = 0 Then C0 = C
  C = C - C0
  Print Fusing(c , "#.###");
  Print " pF"
  Lcdpos = 2 : Lcdline = 2 :
   Lcd_pos
  Lcdtext = "Cap ="
  Lcdtext = Lcdtext +
   Fusing(c , "#.###")
  Text = Fusing(c , "#.###")
  Lcdtext = Text
  Lcdtext = Lcdtext + " pF
   "
  Lcd_text
  Waitms 10
Loop


Tim0_isr:
  '1000 µs
  Timer0 = 6
  Ticks = Ticks + 1
  If Ticks = 1 Then
    Timer1 = 0
    Highword = 0
  End If
  If Ticks = 1001 Then
    Lowword = Timer1
    Freq = Highword * 65536
    Freq = Freq + Lowword
    Ticks = 0
  End If
Return

Tim1_isr:
  Highword = Highword + 1
Return
```

result, the operating point of the oscillator is set automatically, and it starts reliably and produces a symmetric square wave at the output.
With regard to the effect of the oscillator circuit on the resonant loop, the main consideration is the resistor values. The voltage divider formed by the two 100-kΩ resistors loads and thus damps the resonant loop with an effective value of 50 kΩ. There is also the resistance of the negative feedback resistor (100 kΩ) divided by the effective voltage gain. As a result, stable oscillation is possible with sensor capacitance values of up to 100,000 pF (or more). The open-circuit frequency is approximately 350 kHz, which yields an effective capacitance of around 20 pF. The inductor accounts for 15 pF of this, while the input capacitance of the LM311 and the stray circuit capacitance add another 5 pF.
If you use an oscilloscope to view the signal on the inductor, you will see an amplitude of approximately 1 V at the highest frequency and a somewhat distorted sinusoidal waveform. This means that the excitation level could be reduced even further. However, with increasing sensor capacitance the amplitude decreases noticeably and the signal becomes more sinusoidal. The oscillator still works at 100 nF, with a frequency of 4.9 kHz and a signal amplitude of 0.1 V. It stops operating suddenly somewhere above this figure.
The next issue to be considered is frequency stability. The fact that the circuit only contributes 5 pF to the capacitance of the resonant loop is in itself favourable. This leaves us with the difficult question of the temperature dependence of the inductance. The only way to answer this question is to perform experiments. To make a long story short, we can say that the stability of the prototype version built on stripboard in the Elektor labs (**Figure 2**) is sufficient to achieve a sensitivity of 0.001 pF, or in other words 1 fF (1 femtofarad – what an uncommon term!). Incidentally, frequency measurement is not the limiting factor. At 350 kHz and 20 pF, a change of 1 Hz corresponds to a capacitance change of only around 0.1 fF. However, the effective constancy is somewhat lower.

Frequency measurement

Now we come to familiar ground. Frequency measurement was already described in instalment 4 of the Bascom AVR series (Elektor December 2008). The counter input is T1 (PD5), and the frequency in hertz can be obtained directly with a gate period of 1 second. It is sent directly to the PC at 9600 baud, without any correction or window dressing. All that's left is to convert the frequency into capacitance. We use a single-precision variable for this. The conversion formula must be broken down into individual operations in Bascom. Here you have to ensure that the intermediate values do not become too large or too small, since this would degrade the accuracy. This means that the sequence of the operations is somewhat important. The 10 mH of the inductor is expressed as a factor of 10,000,000. The underlying reason for this is to arrive at a value in picofarads at the end. If comparative measurements indicate that the actual value of the inductor is slightly different, such as 1% higher or lower, this is the place to make the correction. The inductor has a rated tolerance of 5%, which means that the capacitance can be measured with a potential error of approximately 5%.
The open-circuit capacitance C_0 is around 20 pF. Of course, the exact value depends on several factors, including component tolerances, PCB construction, and perhaps even the type of solder that is used, since the dielectric

Body capacitance

If you move your hand close to the oscillator (Figures 1 and 2), you will see the measured capacitance change by a few femtofarads, even if no sensor cable is connected. We measured the following approximate results at various distances between the board and our hand:

5 cm	0.005 pF
4 cm	0.009 pF
3 cm	0.020 pF
2 cm	0.040 pF
1 cm	0.100 pF

This is interesting from a physics perspective. The phenomenon of body capacitance is both familiar and notorious among radio hobbyists. If a DIY receiver is not adequately screened, it is often possible to detune it slightly by moving your hand toward it. Some people make handy use of this effect for fine tuning when receiving SSB signals.

Musicians who use Theremin instruments also take advantage of body capacitance.

constant of solder flux can have an effect on the order of a few femtofarads. The only solution to this is to perform a zero-point calibration.
Nothing could be easier: when the user presses a button connected to port B0, the current zero-point capacitance C_0 is measured and stored. This is anyhow necessary, because if you use a cable to connect the sensor it can easily contribute another 10 pF. Consequently, we measure and store the zero offset before making the actual measurement, and this way we obtain the best possible accuracy
The measured values are output in two different ways: via the serial interface and on the familiar LCD with its two-wire interface. At first this was a bit too much for the LCD routine, which didn't want to cooperate with the timer interrupts. The problem was found to arise from passing variables to the subroutines, and it was cured by declaring all variable as global. In addition, the timing was improved to make data transfer even more reliable (see **Listing 1**).
Now the program displays the current frequency and the capacitance. This enables us to make some experimental measurements of temperature stability. For example, you can warm the inductor with your hand and observe the change. With a temperature increase of approximately 20 °C (to around 30 °C), the measured capacitance increased by approximately 0.15 pF. This means that if your objective is to measure the value of an unknown capacitor, the temperature is scarcely important. However, if you actually want to measure capacitance with an accuracy of a few femtofarads, you must first allow the oscillator to stabilise for a few minutes and then make a zero-offset reading. The measured value changes by less than 5 fF over the course of several minutes.

Their hand movements alter the frequency of an oscillator and thus change the audio frequency in a smooth, continuous manner.

You can try this for yourself with this oscillator. Connect a copper-plated board in Eurocard format (100× 160 mm) to act as the sense electrode. This adds approximately 17 pF to the capacitance of the resonant loop, and the frequency drops to around 260 kHz. This is in the long-wave radio band, and you can pick up the signal on a radio. With a bit of luck, you can find a long-wave broadcast signal that interferes with the oscillator signal to produce a beat frequency. Then you can start making music, assuming you have the knack.

All the neighbourhood cats will probably run for cover, but that shouldn't stop you from trying out the effect and learning to understand it, even if you'll never compete with Theremin virtuoso Lydia Kavina, a great-niece of the inventor of the Theremin. The most effective variation in capacitance, around 0.1 pF, occurs at a distance of around 5 cm due to the relatively large size of the sense electrode.

Capacitance measurement

People who play around with RF circuits almost always have something to measure, such as a variable capacitor. Before a true radio hobbyist tosses an old radio in the bin, he at least salvages the variable capacitor, since they are not so easy to come by nowadays. Naturally, you have to measure the salvaged part to know what you actually have. If it has a range of 8 pF to 520 pF, it's brilliant.
You can also measure unknown SMD capacitors, variable-capacitance diodes, the input capacitances of FETs or valves, and cable capacitances. You can even determine the length of a cable by measuring its capacitance. For example, suppose you have a partially used roll of coax cable and you want to feed it down a disused chimney. Before you start, it's a good idea to know whether it's long enough to reach the bottom. We've all heard enough stories about cursing men on high roofs.
This question is easily answered with our capacitance meter. The capacitance per metre is stated on the data sheet. For example, popular 50-Ω RG58 cable has a capacitance of 100 pF/m. If you don't have a data sheet, you can simply measure the capacitance of a known length, such as 1 metre, to determine the number of picofarads per metre. Once you know this value, you can easily calculate the cable length from the measured cable capacitance (cable capacitance divided by capacitance per metre yields cable length in metres). The fact that the cable also has an inductance doesn't matter, since the measuring frequency is much less than the quarter-wavelength frequency. For example, at 100 kHz the wavelength is 3 km.

Liquid level measurement

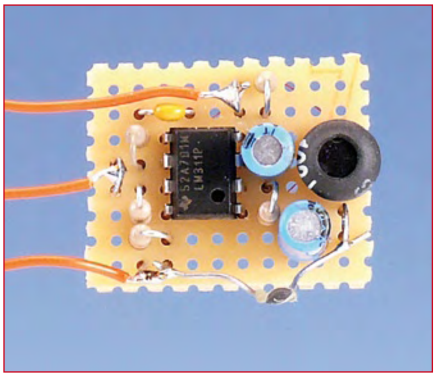


Figure 2. Prototype version of the oscillator, built on a piece of perforated circuit board.

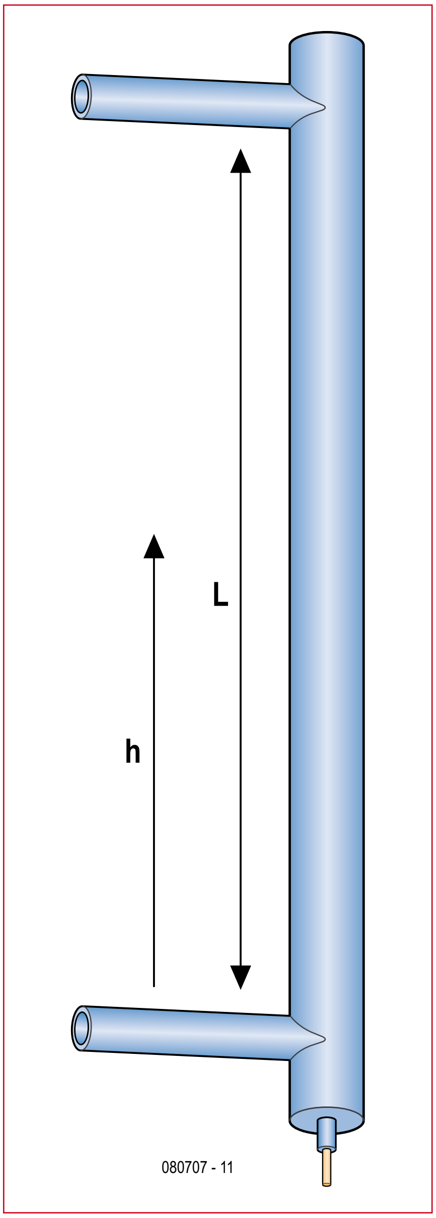


Figure 3. The liquid-level sensor is a tube with an insulated inner electrode that forms a cylindrical capacitor. Here L is the length of the active portion of the tube (wrapped with aluminium foil) and h is the height of the water in the tube.

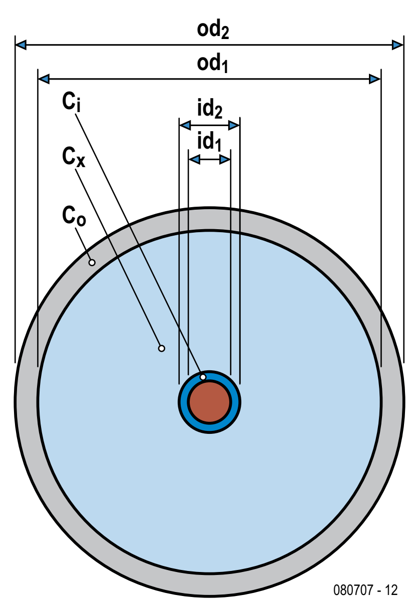


Figure 4. The concentric capacitors of the sensor tube structure.

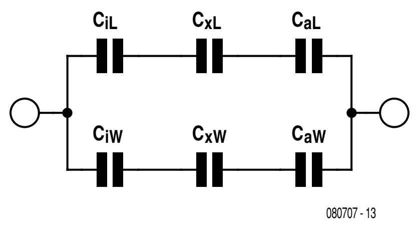


Figure 5. The equivalent circuit of the sensor tube.

To make our liquid-level sensor, we fitted a small Plexiglas (polycarbonate) tube with two connection stubs. A length of polyethylene-insulated hookup wire was stretched through the tube and centred as well as possible, and then both ends of the tube were sealed watertight (**Figure 3**). The conductor of the hookup wire must be fully insulated (galvanically isolated) from the space inside the tube. Then we wrapped the length of the tube between the two stubs with aluminium foil applied as uniformly as possible and attached a bare connecting lead to the aluminium foil (held in place by electrician's tape). The bare lead and the end of the hookup wire protruding from the tube form the terminals of our sense capacitor.

A cylindrical capacitor is a rotationally symmetric form, so its capacitance can be calculated rather accurately by using the following formula if the length is much greater than the diameter:

$$c = \frac{2 \cdot \pi \cdot \varepsilon_0 \cdot \varepsilon_r \cdot l}{LN\left(\frac{od}{id}\right)}$$

ε_0 = dielectric constant of vacuum and air (8.854×10^{-12} As/Vm)
ε_r = relative dielectric constant (material constant)
L = cylinder length
od = diameter of the outer electrode (here od2)
id = diameter of the inner electrode (here id1)

If we combine the constants and convert metres to millimetres, we obtain the following formula:

$$c = \frac{0.0556 \cdot \varepsilon_r}{LN\left(\frac{od}{id}\right)} \cdot l \quad \text{pF/mm}$$

If a cylindrical capacitor consists of several concentric layers, each layer forms a separate capacitor (here C_o, C_x, and C_i). The total capacitance is then determined by the series connection of the individual capacitors (**Figure 4**). If we divide the cylindrical capacitor into a portion filled with water or another liquid (C_W) and a portion filled with air (C_A), the total capacitance of the tube is $C_T = C_W + C_A$ (parallel connection), with the portion filled with water having a length h and the portion filled with air having a length L – h. The equivalent circuit of this arrangement is shown in **Figure 5**.

The relative dielectric constant (ε_r) of air is 1.0, while the relative dielectric constant of water depends on the temperature and ranges from 55 to 88 (approximately 83 at 10 °C). The dielectric constant of transparent plastic is around 3.0 (polystyrene and polycarbonate) or 3.2 (acrylic), and the dielectric constant of wire insulation is around 2.3 (polyethylene) or 4 to 5 (polyvinyl chloride).

This is excellent for our intended measuring applications because it means that there will be a rather large difference between the values of the capacitance Cx in air and in water.

The capacitances in the air-filled portion of the tube are:

$$CiA = \frac{0.0556 \cdot 2.3 \cdot (l-h)}{LN\left(\frac{id2}{id1}\right)}$$

$$Cxl = \frac{0.0556 \cdot 1 \cdot (l-h)}{LN\left(\frac{od1}{id2}\right)}$$

$$CoA = \frac{0.0556 \cdot 3 \cdot (l-h)}{LN\left(\frac{od2}{od1}\right)}$$

Table 2

Sensor tube data (for Figure 6)

Standpipe outside diameter: 12 mm

Standpipe inside diameter: 8.5 mm

Standpipe length: 300 mm

Inner electrode conductor diameter: 0.4 mm

Inner electrode outside diameter: 0.6 mm

Standpipe tube dielectric constant: 3.0

Inner electrode dielectric constant: 2.3

Electrolyte dielectric constant: 83

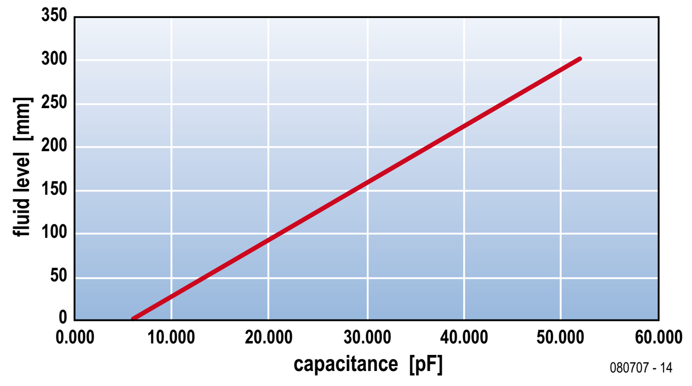


Figure 6. The capacitance increases linearly with the liquid level.

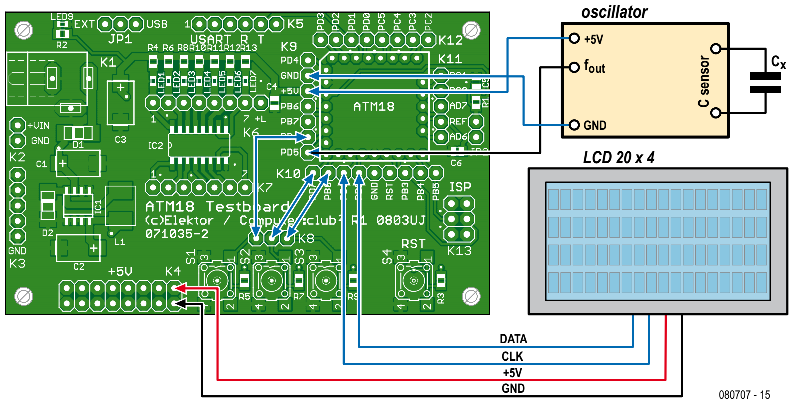


Figure 7. Wiring diagram of the Elektor ATM18 board for the liquid-level gauge.

while the capacitances in the water-filled portion are:

$$CiW = \frac{0.0556 \cdot 2.3 \cdot h}{LN\left(\frac{id2}{id1}\right)}$$

$$CxW = \frac{0.0556 \cdot 83 \cdot h}{LN\left(\frac{od1}{id2}\right)}$$

$$CoW = \frac{0.0556 \cdot 3 \cdot h}{LN\left(\frac{od2}{od1}\right)}$$

If you use a spreadsheet program to calculate and plot the relationship between the total capacitance and the water level, you will discover that it is fully linear if you use a fixed dielectric constant for water. **Figure 6** shows the capacitance as a function of liquid level for a standpipe sensor with the dimensions given in **Table 2**.

Now we can use our standpipe sense capacitor and an inductor with a more or less known value to form a resonant loop, measure the resonant frequency, and use the well-known resonant-loop formula

$$f0 = \frac{1}{2 \cdot \pi \cdot \sqrt{(L \cdot C)}}$$

to calculate the capacitance of the standpipe and thus determine the height of the water in the standpipe.

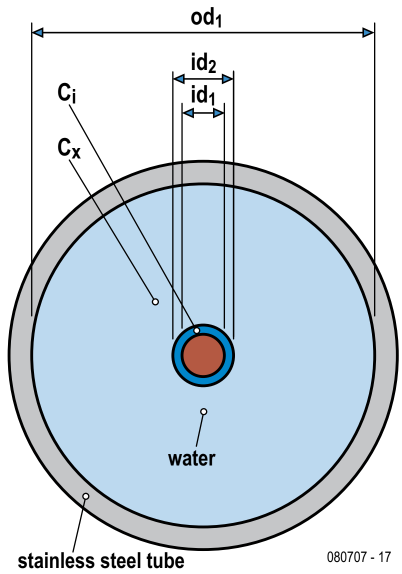


Figure 8. Simplified sensor construction using a stainless-steel or copper outer tube and an insulated brass tube as the inner electrode.

We first measure the capacitance Cmin with the standpipe empty (h = 0) and the maximum capacitance Cmax with the standpipe full (h = L), after which we can use the straight-line formula to calculate the height:

$$h = \frac{L \cdot (C_{measured} - C_{\min})}{C_{\max} - C_{\min}}$$

Here the mechanical accuracy of the construction and the accuracy of the reference inductor do not matter, and the absolute accuracy of the frequency measurement, the presence of parasitic capacitances, and the dielectric constants of the materials used to construct the sensor are equally irrelevant.

The oscillator module (**Figure 2**) should be located as close to the sensor as possible in order to minimise the parasitic capacitance of the cable and reduce the effects of nearby objects on the sensor cable capacitance.

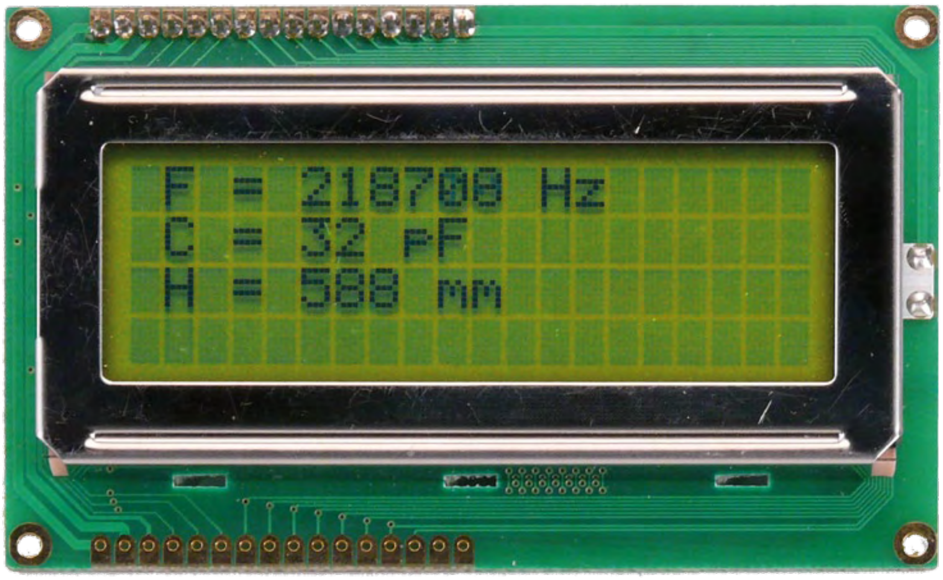
Software

The Bascom project *Level.bas* also uses the serial interface and the LCD. In addition to the frequency and the capacitance, it shows the liquid level in millimetres on the display. A pair of buttons connected to PD6 and PD7 can be used for calibration, with the

Listing 2

Calibration and calculation of the liquid level

```
Hmin = 0.0
 Hmax = 300.0
 Getminmax
 If Cmax <= Cmin Then
   Cmin = 7.0
   Cmax = 52.0
 End If
…

Sub Calclevel
 'ensure that: Hmax>Hmin and
   Cmax>Cmin
 If Cap < Cmin Then Cap = Cmin
 K = Hmax - Hmin
 D = Cmax - Cmin
 If D = 0 Then D = 0.01  'avoid
   division by zero
 K = K / D
 D = -k
 D = D * Cmin
 Y = Cap * K
 Y = Y + D
 Yfix = Y
End Sub

'Calibrate Minimum Value
Sub Calibmin
 Print "Minimum Calibration"
 Bitwait Pind.7 , Set
 Cmin = Cap
 Print "Cmin" ; Cfix ; " pF"
 Eadr = Eadrcmin
 Writeeeprom Cmin , Eadr
End Sub

'calibrate Maximaum Value
Sub Calibmax
 Print "Maximum Calibration"
 Bitwait Pind.6 , Set
 Cmax = Cap
 Print "Cmax" ; Cfix ; " pF"
 Eadr = Eadrcmax
 Writeeeprom Cmax , Eadr
End Sub
```



calibration values being stored in EEPROM. The default values assign a height of 0 to a capacitance of 7 pF and a height of 300 mm to a capacitance of 52 pF. If you adjust the liquid level to a height of 0 mm and press the first button (PD7), the measured capacitance is copied to Cmin and stored in memory. After this, you can fill the sensor tube to the 300-mm level and press the second button (PD7) to copy the corresponding value to Cmax. This data is held in non-volatile memory, so it is available the next time you switch on the instrument (see **Listing 2**).

If the parasitic capacitance of the cable (approximately 33 pF) is taken into account, the measured values are amazingly close to the theoretically determined values. From this we can conclude that a method based on purely theoretical calculation (without calibration of the minimum and maximum levels), and taking the temperature dependencies of the electrolytes into account, could be implemented with a reasonable amount of effort.

As already mentioned, the simple approach only works if you assume that the dielectric constant of the electrolyte (in this case water) remains more or less the same after calibration. The error due to electrolyte temperature variation depends on the dimensions of the sensor tube, and with the prototype arrangement it is approximately 1 mm per 20 °C.

If this is not acceptable, you will have to measure the temperature of the electrolyte as well and use a table to determine the actual dielectric constant. Unfortunately, the simple calibration procedure is no longer feasible in this case, and the liquid level must be determined using the theoretical formulae. With this approach, the accuracy of the sensor tube construction, the exactness of the dielectric constants of the tube insulation and the insulation of the centre electrode, and the accuracy of the reference inductor and the frequency measurement are very important for obtaining good results. In addition, the parasitic capacitance of the connecting cable must be measured exactly.

Choice of materials

A wire with polyethylene (PE) insulation is a better choice for the inner conductor than one insulated with polyvinyl chloride (PVC) because the dielectric constant of polyethylene has a very small range of variation and lies between 2.28 and 2.3. A good way to obtain such a wire is to remove the sheath and braid from a length of coax cable. If the dielectric is transparent, it is solid polyethylene with $\varepsilon_r = 2.3$. Naturally, you can also use a glass tube (ε_r range: 6 to 8) for the sensor.

It's even easier if you can allow the electrolyte to make electrical contact with a sensor electrode and the electrolyte is electrically conductive (which is the case with normal water). In this case the electrolyte acts as the outer electrode of the capacitor (see **Figure 8**). Here again there is a linear relationship between the capacitance and the liquid level. The temperature dependence of the electrolyte is largely irrelevant as long as the conductivity of the electrolyte is much greater than the conductivity of the insulation of the inner electrode. This is always the case with tap water.

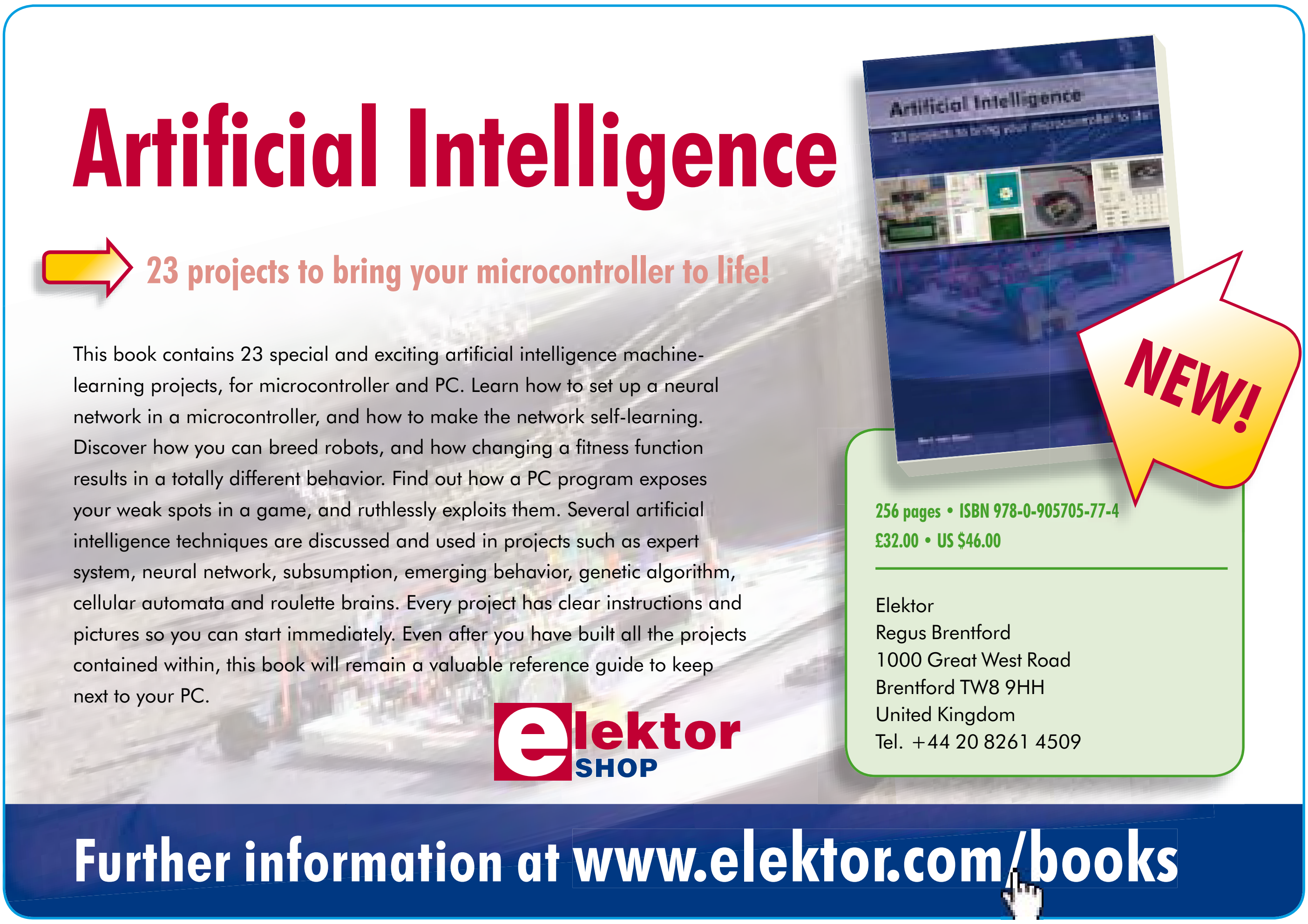
Constructing the sensor is a bit tricky in this case because the inner electrode cannot be clamped at both ends. The best approach is to use a thin brass tube (from a DIY shop) and insulate it with heat-shrink tubing so the brass does not come in contact with the electrolyte. Now the trick is to devise brackets that hold the inner tube and the outer tube of the sensor (the outer tube can be made from stainless steel or copper) such that they are accurately concentric. Depending on the diameter of the outer tube, an arrangement using plastic champagne corks with a hole drilled through the centre is reasonably effective. Don't forget to also drill a vent hole.

(080707-I)


ANDRE LAMOTHE'S
XGAMESTATION
LEARN STEP-BY-STEP HOW TO BUILD
AND DESIGN YOUR OWN VIDEO GAME CONSOLE!
Design inspired by the
Atari 800/2600, Sinclair ZX Spectrum,
Apple II & Commodore 64!
INCLUDES:
WWW.XGAMESTATION.COM


PCBs in
24 hours
from £30.00
www.PCBTrain.com
Tel: 01635 40347
Newbury Electronics Ltd


Prototype & small series PCB specialists
EURO CIRCUITS
Instant online pricing and ordering
Low order-pooling prices - 1–8 layers
Full options service On demand - 1-16 layers
Deliveries from 3 days
Stencil service
New PCB prototype service
- 2 boards in 5 days
- No tooling charge
- Low PCB-Proto prices
E.g. 2x 160x100 mm: 2 layers 49€ each*
4 layers 99€ each*
- Immediate online ordering
- No minimum order charge
*excluding transport and VAT
PCB PROTO
Call us: 020 8816 7005 Email: euro@eurocircuits.com
www.eurocircuits.com
NEW show – Earls court, stand H31, 16-18 June


Artificial Intelligence
23 projects to bring your microcontroller to life!
This book contains 23 special and exciting artificial intelligence machine-learning projects, for microcontroller and PC. Learn how to set up a neural network in a microcontroller, and how to make the network self-learning. Discover how you can breed robots, and how changing a fitness function results in a totally different behavior. Find out how a PC program exposes your weak spots in a game, and ruthlessly exploits them. Several artificial intelligence techniques are discussed and used in projects such as expert system, neural network, subsumption, emerging behavior, genetic algorithm, cellular automata and roulette brains. Every project has clear instructions and pictures so you can start immediately. Even after you have built all the projects contained within, this book will remain a valuable reference guide to keep next to your PC.
elektor SHOP
NEW!
256 pages • ISBN 978-0-905705-77-4
£32.00 • US $46.00
Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom
Tel. +44 20 8261 4509
Further information at www.elektor.com/books

V & I Calibrator

Have faith in your measurements

Dr. Thomas Scherer (Germany)

It's difficult to be sure that your digital multimeter (DMM) is taking accurate measurements especially if it's a few years old. This handy calibrator gives full scale reference levels of both voltage and current, designed specifically for the scale ranges used by DMMs.

DMMs which claim to have a basic accuracy better than 1% can these days be found for less than £20. Even instruments with better than 0.5 % accuracy are selling for less than £100.

At the other end of the scale you can find low-spec 'no name' digital multimeters for just a few pounds at 'bargain basement' outlets and jumble sales. You may have doubts about the accuracy of these instruments. Even the better known brands do not give any figures regarding long term accuracy. Periodic recalibration is recommended. You can of course blindly trust the display readings but as they say 'confidence comes with calibration'. This calibration circuit is small enough to find a space on any workbench and will facilitate speedy and precise multimeter (re)calibration.

Voltage reference devices

The basic requirements for the calibration circuits are that it must supply a known stable DC reference voltage level. The multimeter to be calibrated is connected to the reference supply and adjustments are made to its calibration preset (see **Figure 1**) until the displayed value is the same as the known voltage level. Both the DC volt-

Figure 1. A simple low-cost multimeter with the back removed to reveal the calibration preset (red arrow).

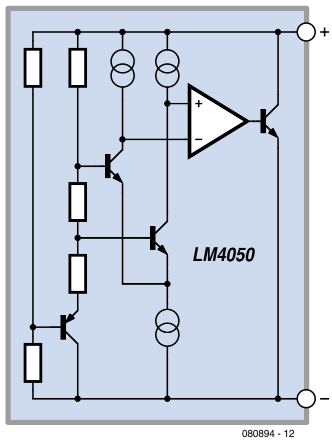


Figure 2. Block diagram of the LM4050 precision zener.

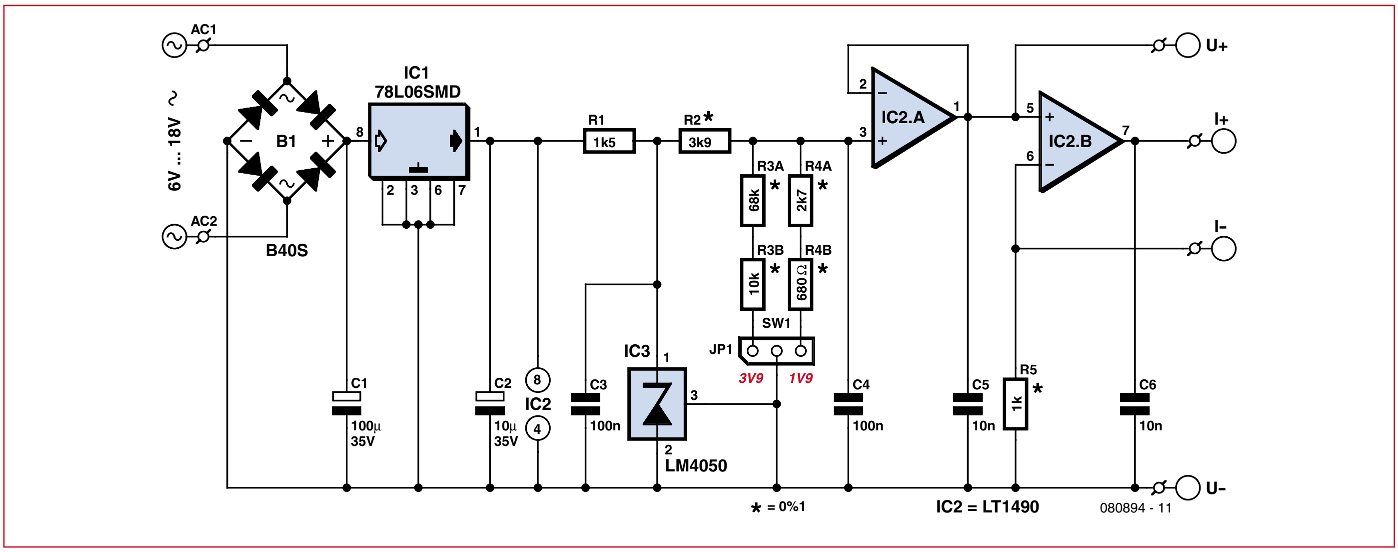


Figure 3. The volts calibrator has a voltage regulator, voltage reference, switchable voltage divider networks and a dual op amp used as a buffer and voltage/current converter.

age range and current range can be checked. The most important component in this circuit will undoubtedly be the device which generates the stable reference voltage. Fortunately there are a number of suitable ICs available which offer good precision at low cost. The LM4050 from National Semiconductor [1] was eventually chosen for this application. The LM4050A-4.1 acts as a high-precision zener diode producing a 4.1 V reference with an accuracy of 0.1%, the device is priced at around £2.00 and is available in a two-pin SMD-SOT23 package. The internal block diagram of the LM4050A is shown in **Figure 2**.

In its simplest form the calibration circuit can be made by connecting the LM4050 in series with a suitable resistor across a supply voltage (higher than the reference voltage). The multimeter to be tested then measures the voltage produced across the IC and its calibration preset is adjusted. The 10 to 20 MΩ input impedance of the multimeter will not impose a significant load on the reference voltage. OK, job done…I knew this was going to be a short article.

Reference voltage level

That certainly would do the job but to make a more useful universal calibration device requires a bit more thought and planning. High precision voltage reference ICs are available with a range of fixed reference voltage outputs. The part number suffix usually indicates the reference voltage. A typical range of standard voltage references would include 1.024, 1.200, 1.240, 2.000, 2.048, 2.500, 3.000, 3.300, 4.096, 5.000 and 10.000 V, none of which are ideally suited to our needs here. Reference levels of 1.000 V, 2.500 V and 5.000 V are fine for analogue multimeters but for digital multimeters it is necessary to produce a level just below full-scale. To find the optimum voltage reference we need to look more closely at the way DMM scale ranges work.

Technical Specification

- 0.1 % accuracy at 25°C
- Temperature stability: 50 ppm/°C
- Output voltage: 3.9 V/1.9 V switchable
- Output current: 3.9 mA/1.9 mA switchable
- Power requirements: 6 to 18 VAC or 9 V battery
- Current consumption: 5 mA

A standard 3½ digit DMM can display readings in the range from 0 to 1999 while a better 3¾ digit device can show 0 to 3999. Using a meter with manual range selection and a reference value of 2.000 V (or 4.000 V for 3¾ digit) will cause the DMM display to indicate an overflow. This is not too much of a problem; you can turn the calibration adjustment on the DMM until the display is on the point of alternating between full scale and overflow. The DMM will then be calibrated with sufficient precision. Meters with automatic range select however will switch up to the next range, so a reference level of 2.000 V (or 4.000 V) will be displayed as 02.00 V (or 04.00 V). The resulting reduction in measurement resolution caused by the loss of a decimal place amounts to 1/200 = 0.5% (or 1/400 = 0.25%). This would significantly reduce calibration precision; the reference voltage has an accuracy of 0.1%.

Testing various makes of autoranging DMMs showed that they do not all behave identically as the measured voltage approaches full-scale. Some of them switch up to the next range at 1.950 V (or 3.950 V) while others display up to 1.999 V (3.999 V) before they switch. This influences the choice of reference voltage level, it was found that all the meters tested remained stable in the lower range (giving best resolution) with a voltage level of 1.900 V (or 3.900 V). The resulting resolution at this level now shows an improvement to the more acceptable figure of 1/1900 = 0.053% (or 1/3900 = 0.026%).

Precision circuitry

The circuit diagram (**Figure 3**) shows that precision resistors are used in two voltage divider networks to derive the 1.900 V and 3.900 V voltage levels from a reference voltage of 4.096 V produced by the LM4050A-4.1 (IC3). An AC mains adapter with an output voltage in the range of 6 to 18 V will make a suitable supply for this circuit. Regulator IC1 produces the 6 V supply voltage. Full wave rectifier B1 on the input ensures that the polarity of the

supply voltage is not important. The circuit draws low current so for occasional use a 9 V battery is also a suitable power source.
Jumper JP1 is used to select between the two different output levels to cater for both 3½ and 3¾ digit meters. The current through R1 is 1.27 mA. The current through IC3 is either 1.22 mA or 0.71 mA depending on the position of JP1 but both of these values lie around the middle of the optimum operating curve specified in the IC3 data sheet.

It can be difficult to find a supplier who stocks the complete E96 series of 0.1% tolerance resistors so R3 and R4 are both made up of two resistors in series. This allows resistor values from the more popular E12 series to be used and also makes it easier to select resistor combinations to give exactly the right output voltage.
The resulting reference voltage is now filtered by C4 and connected to the noninverting input of IC2A. This Rail-to-Rail Input and Output op amp features a low input offset voltage typically less than 0.2 mA. It is configured here as a buffer for the voltage reference [2]. The second op amp in the package (IC2B) with the help of R5 is configured as a voltage to current converter. The position of JP1 not only switches the output reference voltage but also switches this precise reference output current generator between 1.900 mA and 3.900 mA. This feature allows you to check the meter's current measurement accuracy; it is often the case that the current ranges are less accurate than voltage ranges.

C5 and C6 are used to attenuate any RF signals which may be picked up by the circuit. The author's home lab is situated quite close to several sources of radio frequency interference which can sometimes be troublesome.

Construction and use

IC3 is only available as an SMD outline so with the exception of the precision resistors and both electrolytic capacitors all the remaining components use SMD packaging. The finished PCB is just 40 mm square (**Figure 4**). To make it easier to mount the SMD components R1 and C3 to C6 use the larger 1206 package outline.
The author recommends using the following procedure to mount the ICs and B1: firstly tin just one of the pads on the PCB where the IC will be fitted. Move the IC into position over the pads, clamp it down tightly with one fingernail then using the other hand bring the soldering iron tip in contact with both the tinned pad and IC leg until a joint is formed. Once cool the IC will now be correctly fixed in position. Now after double-checking the IC orientation, solder the remaining leads. Lastly check that you have not accidentally created any solder bridges between pads.

Once the board is fully populated and you have carried out a careful visual check of your soldering handiwork it is time to test the circuit. Connect the supply input pins to the output of an AC mains adapter capable of supplying 6 to 18 V (the circuit consumption is less than 10 mA) or alternatively use a 9 V battery. A DMM can now be connected between 'U+' and 'U–' where either 1.900 V or 3.900 V can be measured. Connect the multimeter leads to 'I+' and 'I–' and switch the range to DC current to measure 1.900 mA or 3.900 mA.
Calibrating the DMM is just as simple: Open-up the DMM case and identify the scale calibration preset (**Figure 1**). Select the correct scale on the DMM and switch it on. Connect the calibrator output to the DMM inputs and adjust the preset until the DMM display value is correct. It is usually sufficient to calibrate just one range, all the measurement ranges are normally linked by cascaded resistor networks and it is very difficult to make any individual changes. Once the voltage range is calibrated the current reference can be used to test the DMM current reading. It is usually not possible to make any adjustment to the displayed current value. You will at least get an indication of how accurate the current readings are and how far the meter readings can be trusted. This also applies to the AC measurement ranges of voltage and current which are not calibrated. To calibrate resistance ranges there are no prizes for guessing that a precision resistor of value 1.8 k or 3.9 k can be used as a reference.

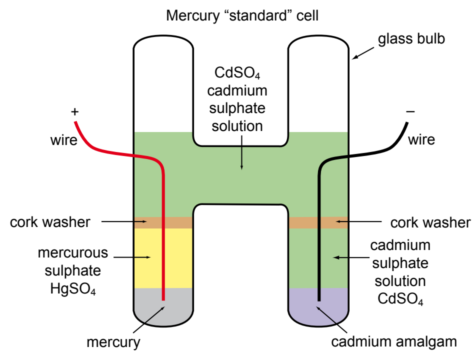
Variations

The circuit will also work with a 5 V voltage regulator in place of the 6 V version used for IC1. In this case it will be necessary to reduce the value of R1 to 820 Ω. This modification will however prevent the circuit from supplying the reference current to test the current ranges. A current of 4 mA produces a voltage drop of more than 400 mV. The author has a 3¾ digit DMM in his possession which (curiously) produces a voltage drop of just over 1 V at this level of current. In this case the supply voltage to IC2B will be too low. The component values given in the circuit

The Weston standard cell

Reference voltage sources have traditionally been called 'standard cells'. Similar to a battery, they use a combination of galvanic materials to produce a precise reference voltage which is relatively stable and temperature independent. The Weston cell (1893) was the work of the American physicist Edward Weston (1850–1936) and was adopted as the international standard for EMF (electromotive force) in 1911.

Like all galvanic elements the cell has two electrodes suspended in an electrolyte solution. The cathode is mercury and the anode is a cadmium/mercury amalgam while the electrolyte is a solution of cadmium sulphate (see illustration). The Weston cell produces a nominal voltage of 1.01865 V at 20 °C. It has a very low temperature coefficient of less than 10^{-4} V/°C.



The photo at the beginning of the article shows a cell which was made in the second half of the last century. According to the label it produces a voltage of 1.0193 V with an accuracy of 0.1 %. This is better than we have achieved here with our low-cost silicon alternative but it has to be said that our version is less toxic, more robust and much more versatile.

COMPONENT LIST

Resistors
R1 = 1kΩ5, SMD R1206
R2 = 3kΩ9, 0.1%
R3A = 68kΩ, 0.1%
R3B = 10kΩ, 0.1%
R4A = 2kΩ7, 0.1%
R4B = 680Ω, 0.1%
R5 = 1kΩ, 0.1%

Capacitors
C1 = 100µF 35V, radial electrolytic
C2 = 10µF 35V, radial electrolytic
C3,C4 = 100nF, SMD C1206
C5 = 10nF, SMD C1206
C6 = 1nF, SMD C1206

Semiconductors
B1 = B40S, SMD bridge rectifier, 40V/1A
IC1 = 78L05SMD, SO08
IC2 = LT1490, SO08
IC3 = LM4050A-4.1, SOT-23, e.g. Farnell # 1468851

Miscellaneous
JP1 = 3-way SIL pinheader, lead pitch 2.54mm (0.1") with jumper
PCB, # 080894-1

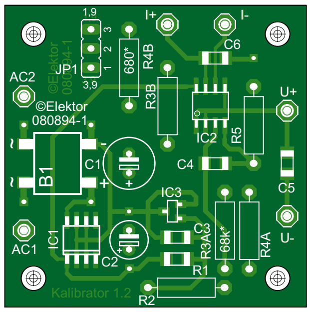


Figure 4. At 40 mm x 40 mm the Volts Calibrator PCB is very compact.

diagram will be suitable to cater for the majority of situations. If you really want to cover every possible case you can use an 8 V regulator for IC1, R1 will now need to be 3.3 kΩ and a minimum AC input supply of 9 V will be required.

In many cases a less precise reference is acceptable; the B version of the LM4050 can be used here. It has a precision of 0.2 % but the price difference between the two chips is only a matter of a few pence. Alternative op amps for IC2 (the LM4050) are the OPA2343 from Burr-Brown or the AD822 from Analog Devices.

The circuit can be fitted into an enclosure; a single pole changeover switch can be wired to the pins of JP1 to replace the jumper. Those of you who would prefer the volts calibrator to produce the more traditional reference values of 1 V and 1 mA can use a value of 750 Ω for R3A and 510 Ω for R3B. Both 0.1%, of course.

(080894-I)

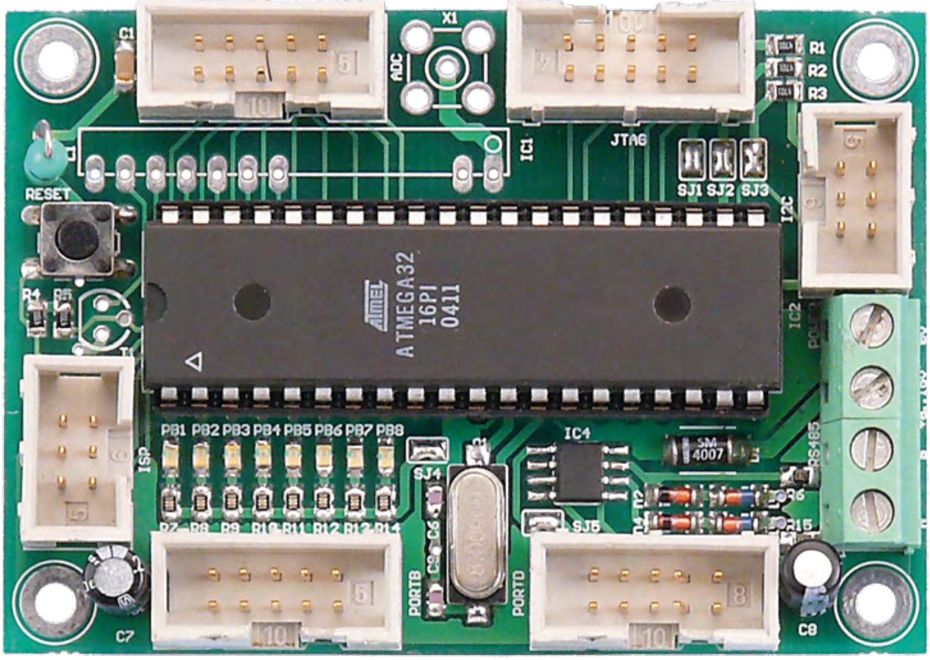
Internet links & Literature

[1] www.national.com/mpf/LM/LM4050.html

[2] www.linear.com/pc/productDetail.jsp?navId=LT1490

Pocket Calculator Control Interface

Communicating with a TI-83(Plus) or TI-84 Plus



Koen Kempeneers (Belgium)

Programmable calculators used to be exotic, but now they are standard equipment in almost every school or polytechnic. The TI-83 (Plus) and TI-84 Plus are especially popular. Both models have a 'link port' for exchanging data. If you connect the right hardware to this port, you can do much more with it.

Every technician or engineer has a pocket calculator somewhere on their desk. Although the internal processes of these calculators are hardly suitable for control tasks, some of them can be used to build a nice robot.

All TI graphic calculators have a 'link port', which takes the form of a 2.5-mm (0.1") stereo headphone socket that enables the unit to exchange data and programs with other units and send commands to a calculator-based laboratory (CBL) system or a calculator-based ranger (CBR). However, these machines can do even more. The interface described here gives the calculator access to 16 digital I/O lines and four analogue inputs linked to 10-bit converters.

Link port

The headphone connector has two signal contacts and a ground contact (GND). However, it uses a non-standard communication protocol specially devised by TI for this purpose. Fortunately, the protocol is remarkably simple. In the quiescent state, the measured signal level on each communication line is around 3.3 V. When data is to be sent, the transmitter pulls one of the two lines to 0 V, and the receiver then pulls the other line to 0 V in acknowledgement. To send a '0', the transmitter pulls D0 (the white lead of the cable) to 0 V first, and to send a '1' it pulls D1 (the red lead) to 0 V first.

It is also important to know the order in which the individual bits leave the link port. Instead of the usual practice of sending the LSB first, with the link port the MSB is sent first.

Technical specifications

- 32 ESD-protected inputs/outputs including connectivity for I²C, AVR-ISP, JTAG, RS485 and general I/O.
- Supply regulation using 7805
- Firmware all in C
- Simple to program

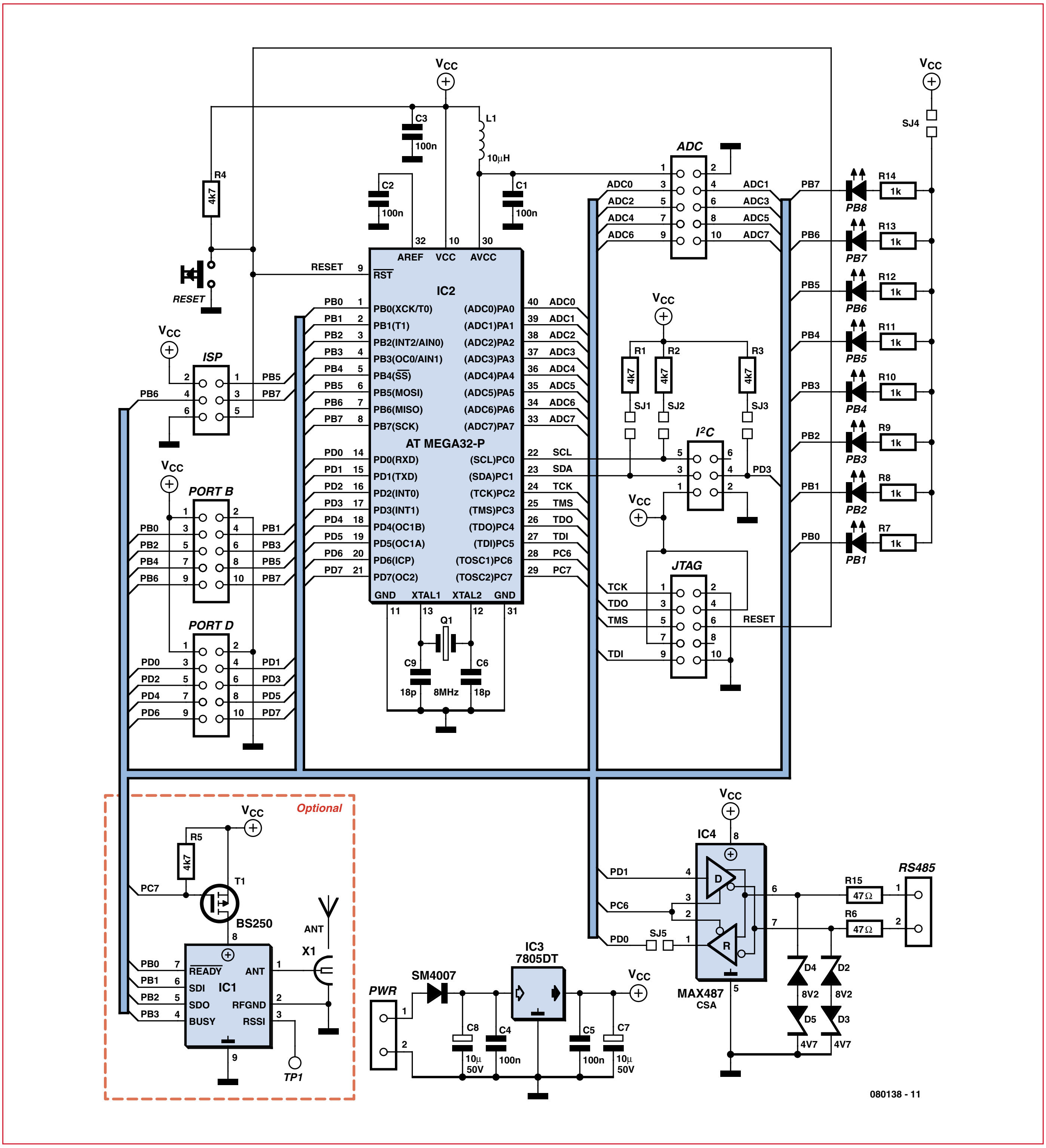


Figure 1. Schematic diagram of interface board.

Table 1. LED values	
LED	**Weight**
LED1	2^0 (1)
LED2	2^1 (2)
LED3	2^2 (4)
LED4	2^3 (8)
LED5	2^4 (16)
LED6	2^5 (32)
LED7	2^6 (64)
LED8	2^7 (128)

Interface

The hardware used here was not designed specifically for this project. The author had already developed it for another project, but he realised that it could be put to good use for this project as well. The interface hardware is simplicity itself (see **Figure 1**). The ATMega32 has everything necessary in house (and even more). The four ports of the ATMega32 provide a total of 32 ESD-protected discrete I/O lines. In addition, the microcontroller has ample internal memory, so the program can be extended as desired.

The schematic diagram of the interface board is fairly standard. The microcontroller is accompanied by supply voltage decoupling capacitors, decoupling capacitors for the analogue input supply voltage, and a crystal with the associated capacitors. There is also a wealth of connectors that can be used to connect sensors and actuators to the interface board. You have your choice of a standard I²C port, a 6-way AVRISP connector, and a JTAG port. The latter port can be used to debug the application software in the microcontroller if you want to add your own code. The servo connectors and an TTL/RS232 transceiver round out the picture.

The hardware for the link port implementation on the interface board requires only two I/O lines (PC0/SCL and PC1/SDA) and two pull-up resistors (R1 and R2). The interface board also has several LEDs that can be connected via SJ4 and used to simplify program testing.

The RS485 port is protected by zener diodes that divert excessively high or low voltages to ground. The supply voltage is stabilised by a 7805 voltage regulator. A standard AC mains adapter rated at 8 to 16 V is adequate for powering the circuit.

The firmware for the ATMega32 was written entirely in C, and the compiler (AVR-GCC, which is available free of charge and integrates seamlessly with AVR Studio [1]) is perfectly adequate for the task. The microcontroller can be programmed relatively easily using AVRDude [2].

Writing a program

In order to use the interface board with the calculator, you also need a program that runs on the calculator. You can easily write a suitable program yourself. The procedure for writing the 'Blink' program is described in the inset.

Let's have a brief look at how this program works. The first line performs the initialisation. Here the variable A is assigned the value '0'. The second line marks the start of a loop. The calculator executes the statements between this line and the 'End' label as long as the condition after 'While' is satisfied. In this case the condition is 'getKey=0', which means as long as a key is not pressed.

The third line of the program causes the LEDs to blink. If the value of A is '1'. the first LED lights up after the *Send* command, and if the value is '0', it goes dark. The fifth line generates a short delay, as otherwise the LED would blink too fast.

Exchanging data

You use the *Send* command to send data to the interface. In principle, you can send any type of variable, but the interface only responds to real numbers, lists of real numbers, and character strings.

The possible syntaxes for the *Send* command are:

- Send(variable_name (A - Z))
- Send({*n0*,*n1*,*n2*, ... ,*nx*})
- Send(Str*[0 - 9]*)

If an integer is sent to the interface, it is transferred to the direct 8-bit I/O port (LEDs), regardless of the name of variable used to store the number in the calculator. The range of values that can be assigned to this port is 0 to 255. Each LED represents a particular weight (power of 2); the weights of the individual LEDs are listed in **Table 1**.

The *Send* command can be used to send 8-bit data to the interface at high speed. The syntax for sending a command is the same:

- Send({Cmd,arg1,arg2, ... ,argx})

Here 'Cmd' stands for the name of the specific command the interface is supposed to execute. The commands supported by the interface are listed in **Table 2**. Here you should note that in order to maintain the general-purpose nature of the interface, it does not have a mechanism to check the number of arguments or the validity of the arguments. If the interface suddenly stops responding after an incorrect command, it must be reset.

You use the *Get* command to fetch data from the interface. Only real numbers can be fetched from the interface. The syntax for *Get* is thus very simple:

- Get(*variable_name (A - Z)*)

If *Get* is not preceded by *Send*, the status of the 8-bit digital I/O port PB0: PB7 is read (the port connected to the LEDs).

Construction

At first glance, the only components you see on the board are a lot of connectors and a large IC. If you look more closely, you also notice the tiny SMD components (see the PCB layout in **Figure 2**). The double-sided board has

Table 2. Interface commands

Cmd	Function	Arguments	Example
0	Sends a value to a special function register (SFR) of the interface	SFR address, value	Send({0,55,A})
1	Enables the interface to read the value of an SFR in response to the following Get command	SFR address	Send({0,55})
2	Sends a value to a 16-bit special function register (SFR16)	SFR16 address, value	Send({0,55,A})
3	Enables the interface to read the value of an SFR16 in response to the following Get command	SFR16 address	Send({0,55})

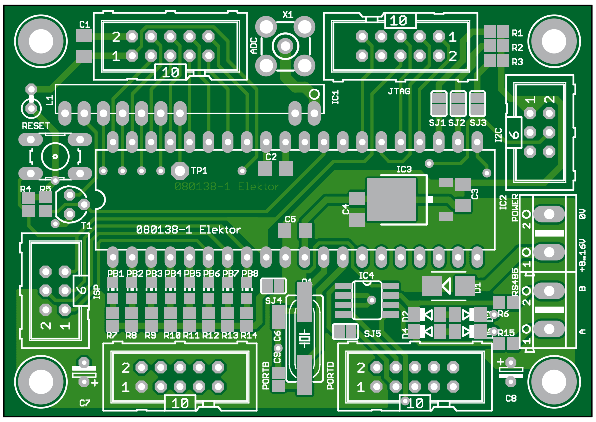


Figure 2. As you can see from the component layout, the IC takes up most of the board area.

COMPONENT LIST

Resistors
R1-R5 = 4kΩ7 (SMD 0805)
R6,R15 = 47Ω (SMD 0805)
R7-R14 = 1kΩ (SMD 0805)

Capacitors
C1-C5 = 100nF (SMD 0805)
C6,C9 = 18pF (SMD 0805)
C7,C8 = 10µF 50V

Semiconductors
D1 = SM4007
D2,D4 = 8.2V zener diode
D3,D5 = 4.7V zener diode
IC1 = not fitted
IC2 = ATmega32-P, programmed
IC3 = 7805DT
IC4 = MAX487CSA
PB1-PB8 = LED (SMD 0805)

Miscellaneous
L1 = 10µH
Q1 = 8MHz
PORTB,PORTD,JTAG,ADC = 10-way boxheader
ISP,I2C = 6-way DIL pinheader
POWER,RS485 = PC terminal block, lead pitch 5 mm (0.2")

components on only one side. When assembling the board, it is best to follow the usual practice of starting with the SMD components. In particular, the components *beneath* the IC must be fitted first. Even if the IC is mounted on a socket, which is actually a good idea, these components are difficult to reach. Be sure to bridge all the surface jumpers (SJ1 to SJ5) with a drop of solder.

After the SMDs are fitted, it's time for the other standard components (there aren't many), and finally the headers and PCB-mount terminal strips.

As the hardware was not originally designed for this project, it is not necessary to populate the entire PCB. The components in the dashed area of the schematic diagram (**Figure 1**) do not have to be fitted (they are intended for a different project).

Conclusion

The Web references listed below include a link to an interesting site with handy information on TI calculators [3]. It provides tutorials, example programs, and access to a lively user community.

For the proper configuration of the fuse bits when programming the microcontroller, refer to the Engbedded website [4]. The firmware for programming the microcontroller can be downloaded from the Elektor website [5].

(080138-I)

Internet Links

[1] www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

[2] http://meuk.spritesserver.nl/projects/avr_stuff/avrdude-gui_v0.2.1.zip

[3] www.ticalc.org

[4] www.engbedded.com/cgi-bin/fc.cgi

[5] www.elektor.com/080138

Writing a program

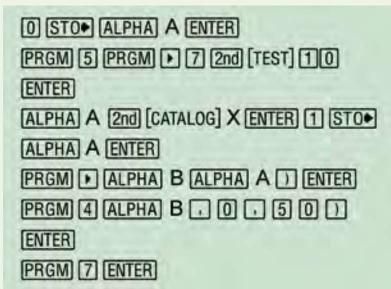
To begin writing a new program, press the PRGM button and then press => twice. You will then see the following screen:

Now press ENTER to start the editor. First enter the name of the new program, and then start entering the program.

The following program causes a LED to blink on the I/O interface:

```
:0->A
:While getKey=0
:A xor 1->A
:Send(A)
:For(B,0,50)
:End
```

For first-time programmers, the key combinations necessary to enter this program are listed below:



If everything has been entered correctly, this is what you should see on the calculator:

XMEGA Revealed

First impressions of the Atmel ATXMEGA1281A1

Benedikt Sauter and Dr. Thomas Scherer (Germany)

Atmel's 8-bit AVR microcontrollers lost their novelty value rapidly to become an established cornerstone of many Elektor projects over the years. The new addition to this family, the XMEGA series, has raised the stakes, elevating these 8/16-bit microcontrollers to a new level of system performance. Elektor author Benedikt Sauter has already sampled an AVR XMEGA.

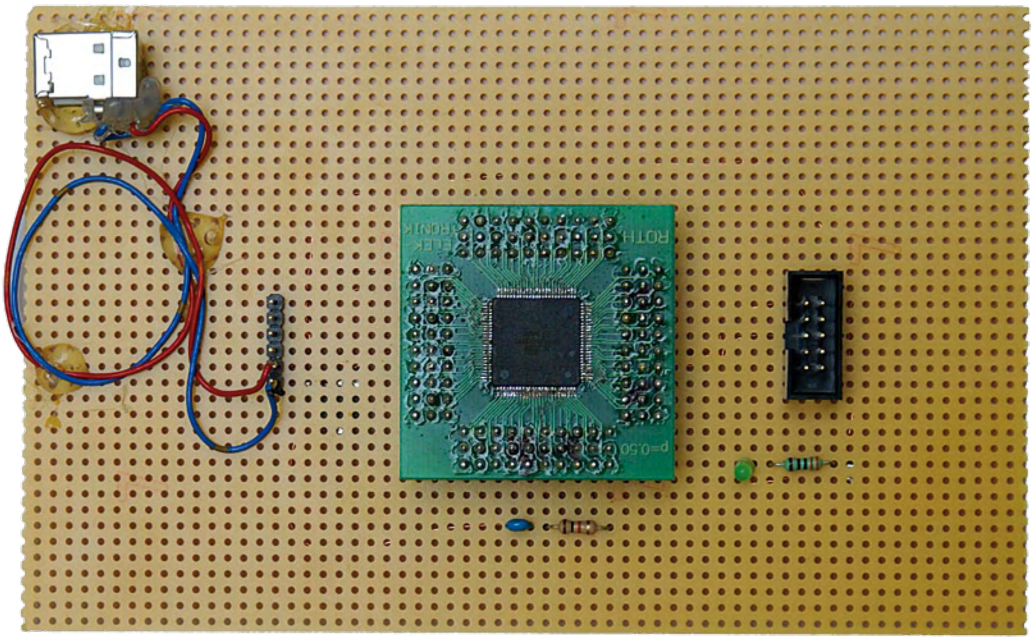


Figure 1. Perf board with XMEGA mounted on adapter panel together with USB and programming interfaces.

Many Elektor readers know and love the 8-bit AVR microcontroller family from Atmel. This is hardly surprising in view of the wide availability of these products, the straightforward data sheets, the free software provided (debugger, compiler, linker, programmer, etc.) and not least the popular and effective BASIC compiler BASCOM-AVR. Even though these 8-bit controllers suffice for very many applications (indeed, the majority), the point must come some time when their number-crunching capacity is finally exceeded. This is the stage where we must find an equally simple and effective solution. Making the transition to a 16- or 32-bit controller family with an unfamiliar development environment is obviously not going to be a straightforward process. Does Atmel's XMEGA product offer an easy way out?

The XMEGAs

For nearly a year now Atmel has indicated that small-scale production was due to start shortly. Originally the new controllers were to be ready to go on sale in the spring of 2008; the company's latest statement (October 2008) is that "Samples of the XMEGA devices are available now with flash memory density from 64 kB to 256 kB in 64 to 100-pin packages.". A bird in the hand is always worth two in the bush, and the author has been fortunate enough to get hold of samples of these still scarce chips and test them in practice. So that's how you, the *Elektor* reader, come to be in a position now to form an impression of what the XMEGA is really like and can see the kind of potential that lurks within these new chips.

The XMEGA family appears—as far as has been stated up to now — a follow-on development of the ATmega series. To this extent it is more of an evolution then, albeit with genuinely new capabilities. It is based (although not unconditionally) on broader bus structures, since the XMEGAs occupy an intermediate position between 8- and 16-bit controllers. The underlying characteristics and new features are set out in an **inset** in this article.

Plenty of improvements have been made under the bonnet but all of them on the basis of an 8-bit AVR kernel: faster clock speeds, up to 16 MBytes of external memory, a DMA interface and an intelligent event system for cutting processor time all generate a demonstrable uplift in performance in return for a comparably modest current consumption. But perhaps the most important aspect is the close relationship with the previous AVR series: the 'old' software (compiler, linker, debugger etc.) can all still be used — and there is already an XMEGA version of BASCOM [1]! The learning curve and transition pains are really not that great then.

Quick test

The test sample was an ATXMEGA128A1 [2], a controller with no less than 78 I/O lines in a 100-pin TQFP package. To construct our test set-up we first soldered the controller to an adapter board, which was then attached to a piece of perf board (**Figure 1**).

Very little more is needed for this test bed; only a power-on-reset-switch (a 10-kΩ resistor and a 100-nF capacitor suffice for this) and a stabilised power supply.
At this stage we encounter one of the important differences from the familiar AVR microcontrollers. The XMEGAs cannot be powered from a 5 V supply; instead they must be fed with a voltage between 1.6 and 3.6 V. The best solution is an integrated voltage regulator like the LM1117-3.3V, using two electrolytics naturally. This is the easy way of producing 3.3 V from the 5 V available on a USB connector.

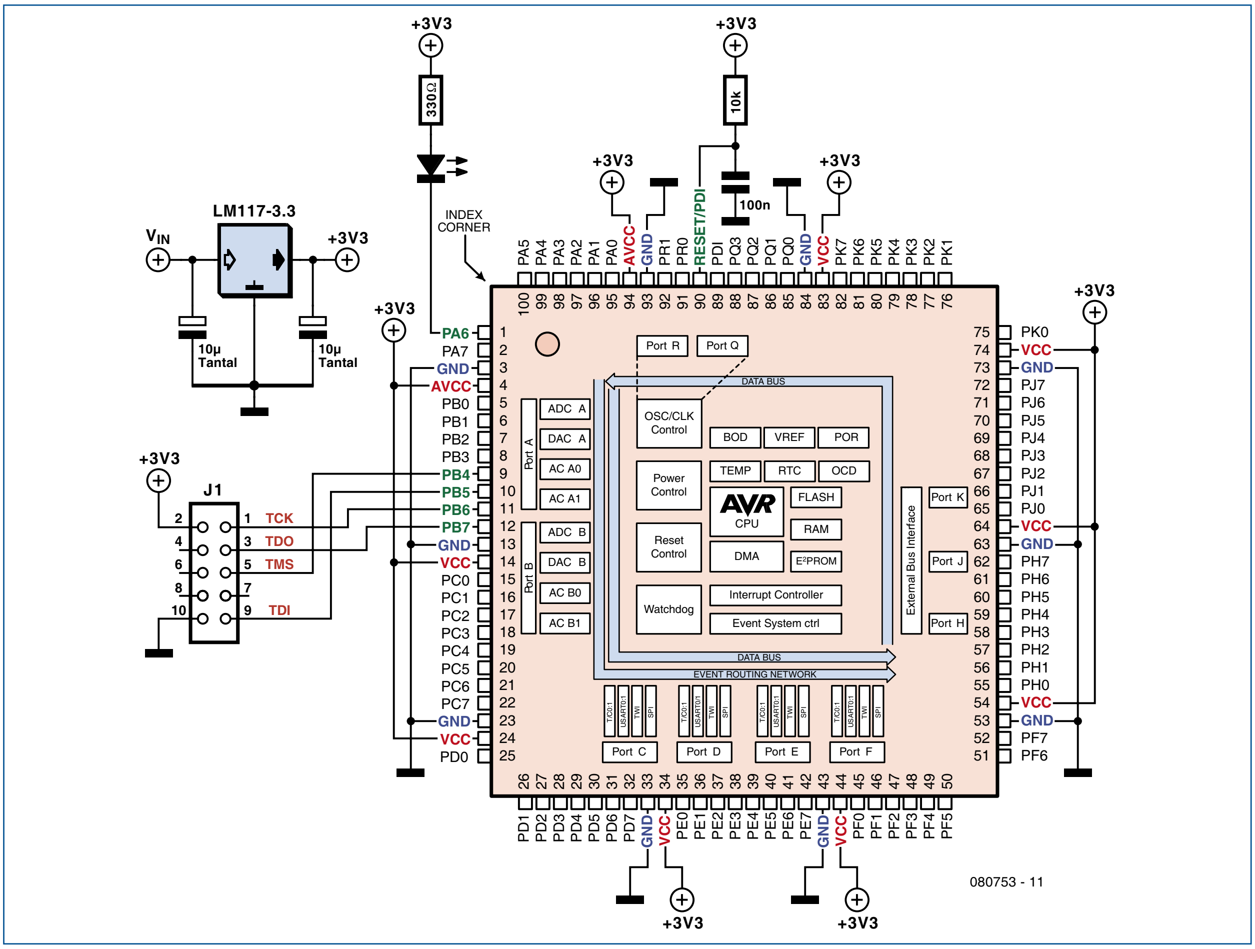


Figure 2.
Only a few components are needed to construct a test bed for XMEGA controllers.

An external crystal or ceramic resonator is not required, since several oscillators are provided internally, unlike on previous controllers, and the clock speed is high (up to 32 MHz). For loading programs into the internal flash memory of the XMEGA we use a JTAG interface. The pin assignments of the 10-way 'bathtub' connector must naturally match those of the JTAGICE Mk II programming adapter, in order that this can be used. The STK600 starter kit available from Atmel can also be put to use here. Another possibility here is to use the technique already much loved with the ATmega in the form of a boot loader via a simple serial interface, since the XMEGAs make use of the same boot mechanism. Additionally there is a bus similar to the ISP on the existing ATmegas, which will probably be supported by the low-cost programming adapters such as the AVR ISP Mk II and the AVR Dragon.
An LED (with series resistor) provides a simple output indicator on our perf board 'XMEGA Tester', the circuit of which is shown in **Figure 2**.

Software

As with the AVR controller you can find the complete 'tool chain' (compiler, linker, IDE, programmer and debugger) for the XMEGA free of charge on the Internet. Note that the two separate software packages need to be installed together: firstly WinAVR [3] (compiler complete with linker etc.) and secondly the development environment, AVR Studio [4] (editor, project control, integration of compilers, debugger, etc.). It is best to install the Open Source pro-

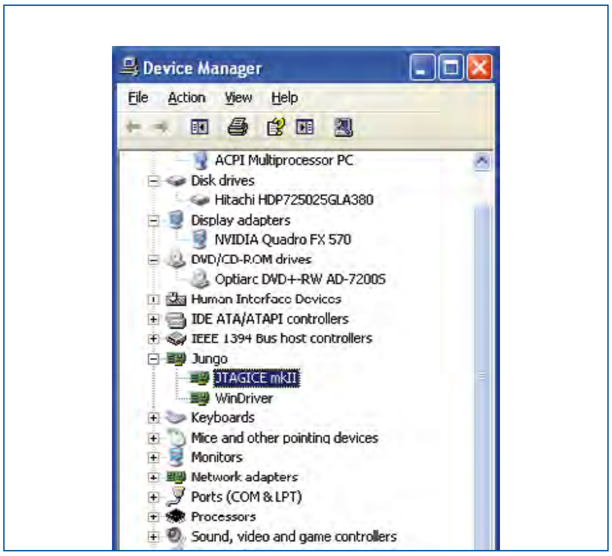


Figure 3. Successful entry of JTAGICE Mk II into the system control.

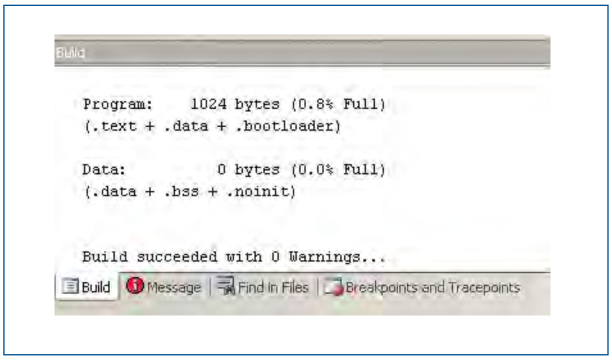


Figure 4. The status window of AVR Studio following a compiler run.

Listing 1. LED Flasher test program

```
#INCLUDE "AVR_COMPILER.H"
#INCLUDE "PORT_DRIVER.H"
#INCLUDE "CLKSYS_DRIVER.H"
#INCLUDE <AVR/IO.H>
#INCLUDE <UTIL/DELAY.H>

INT     MAIN( VOID )
{
   /* ACTIVATE INTERNAL OSCILLATOR */
   CLKSYS_PLL_CONFIG(
        OSC_PLLSRC_RC2M_GC, 30 );
   CLKSYS_ENABLE( OSC_PLLEN_BM );
   CLKSYS_PRESCALERS_CONFIG(
        CLK_PSADIV_1_GC,
        CLK_PSBCDIV_1_2_GC );
   DO {} WHILE ( CLKSYS_ISREADY(
        OSC_PLLRDY_BM ) == 0 );
   CLKSYS_MAIN_CLOCKSOURCE_SELECT
        ( CLK_SCLKSEL_PLL_GC );

   /* PORT A IS OUTPUT */
   PORT_SETPINSASOUTPUT(
        &PORTA, 0XFF );

   WHILE(TRUE){
        /* 1 SECOND PAUSE */
        _DELAY_MS(1000);
        /* CHANGE VALUE OF PORT A */
        PORT_TOGGLEPINS(
               &PORTA, 0XFF );
        }
}
```

ject WinAVR (actively supported by Atmel) first. Note that only XMEGA-compatible versions (20080411 onwards) should be used.

After WinAVR comes AVR Studio. Here too only XMEGA-compatible versions (4.14 and higher) are suitable. The USB drivers needed by JTAGICE Mk II are installed along with WinAVR, so long as you tick the corresponding boxes (don't forget, this is important!). After running up JTAGICE Mk II for the first time you can call up Windows Installer to search for suitable drivers. To check how the installation went, just look at the entry in Windows Explorer, see **Figure 3**.

After this, as soon as you boot up AVR Studio, it should automatically offer you the opportunity to create a C-Project. If so, WinAVR has been found and integrated successfully.

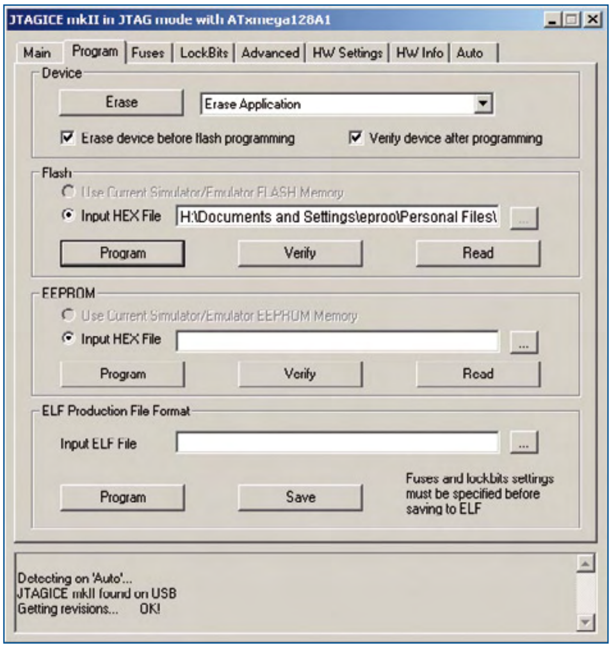


Figure 5. The dialogue window of AVR Studio for selecting the flasher hex file.

Commissioning

At start-up AVR Studio offers you the choice of retrieving an existing project or starting a new one. Logically your first test needs to be to launch a new project and assign it a project name. The next task is to select the controller (here we want ATXMEGA128A1) and the programming and debugging interface (in this case JTAGICE Mk II with USB connection).

At this stage it can happen that AVR Studio may want to introduce new firmware into JTAGICE Mk II. In this situation just follow the software requirements. If everything is still in order the internal signature of the chip can be selected by selecting 'Read Signature' using the tab 'Main'. If this doesn't work you should look for errors as follows:

- Is the power supply voltage correct?
- Are the VCC and GND pins connected the proper way round?
- Are the JTAG connections correct?
- Is the Reset line connected to the JTAG socket via TRST?

Once the process of reading out of the signature functions properly, then downloading software into the controller is no longer a problem.

'Flasher' test program

For our first (admittedly simple) test it will suffice to make an LED flash on and off. The source code (written in C) of the program shown in **Listing 1** can also be downloaded from location [5].

After the typical introductory Include statements the program first activates the internal oscillator and the prescaler and then in the second block defines Port A as output. In the endless loop that follows, Port A, to which the LED is connected, is alternately set Low and then High. And that's it. The program is interpreted with a click on the Compiler icon or with a key press on F7. If no errors are indicated in the status window (see **Figure 4**) the result can be transferred into the controller via the programmer.
After successful compilation the programmer window appears (see **Figure 5**). The Hex file that results from the compiler run is found normally in the 'debug' folder of the project directory. Once the file is transferred into the controller the program starts there immediately and in our test, hurray, the LED really did start to flash!

The verdict

Working with XMEGAs is no different from ATmegas or ATtinys and is just as straightforward, with no unpleasant surprises. An attractive feature is the event system, which enables simultaneous processing of data between peripherals during operations without burdening the processor. In this way for example a timer can initiate directly measurement of an analogue value, without adding to the load of the CPU. Thanks to cryptographic functions such as AES and DES some new and very interesting application areas can be exploited. In practical terms the XMEGAs offer nothing but advantages over the preceding ATmega controllers, meaning that these new microcontrollers be recommended wholeheartedly to all AVR users looking for extra performance.

(080753-I)

Internet Links

[1] BASCOM for XMEGA:
www.mcselec.com

[2] Datasheet for the ATXMEGA1281A1:
www.mcselec.com

[3] WinAVR:
http://winavr.sourceforge.net/

[4] AVR Studio:
www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

[5] Test program LED Flasher:
www.elektor.com/080753 and
www.embedded-projects.net/myxmega.zip

The new XMEGAs — facts and data

The new XMEGA microcontroller represents a significant further development of the well-known 8-bit controller of the ATtiny and ATmega type. The computation module is fundamentally unaltered but has been expanded with some 16-bit operations. The upper clock speed has been raised and the processing performance has been optimised by a hardware multiplier among other things. It has also been enhanced with additional, improved and expanded peripherals as well as a properly configured event system.

Voltage range

1.6 to 3.6 V without limitations (the maximum clock speed of 32 MHz is feasible already at 1.6 V).

Clock speed

Maximum 32 MHz.

4 internal ULP oscillators: 32 MHz, 2 MHz, 32 kHz and 32 kHz

External crystals: 32 kHz and.0.4 to 16 MHz.

Internal PLL with factors up to 1:31.

Prescaler with factors from 1 to 2048.

Timer with maximum of 128 MHz.

Following reset the XMEGA starts at internal 2 MHz.

DMA

A total of four DMA channels make it possible for example to realise Interrupt-driven analogue to digital conversion without putting any additional load on the CPU.

Encryption

Integrated CPU-sparing hardware cryptography with DES or AES algorithms.

Memory

The XMEGAs provide integrated program memory up to a planned maximum of 384 KB. In addition up to 16 MB of external memory can be addressed.

ADC

The resolution of the A-to-D converter is increased to 12 bits. Increased number of channels and maximum sampling rate of 2 MS/s.

DAC

In the same way the XMEGAs are provided with integrated D-to-A converters with 12-bit resolution as standard. Up to one million transformations a second are feasible.

Timer

All timers are provided with 16-bit resolution. The number of times can be raised to eight.

Pins

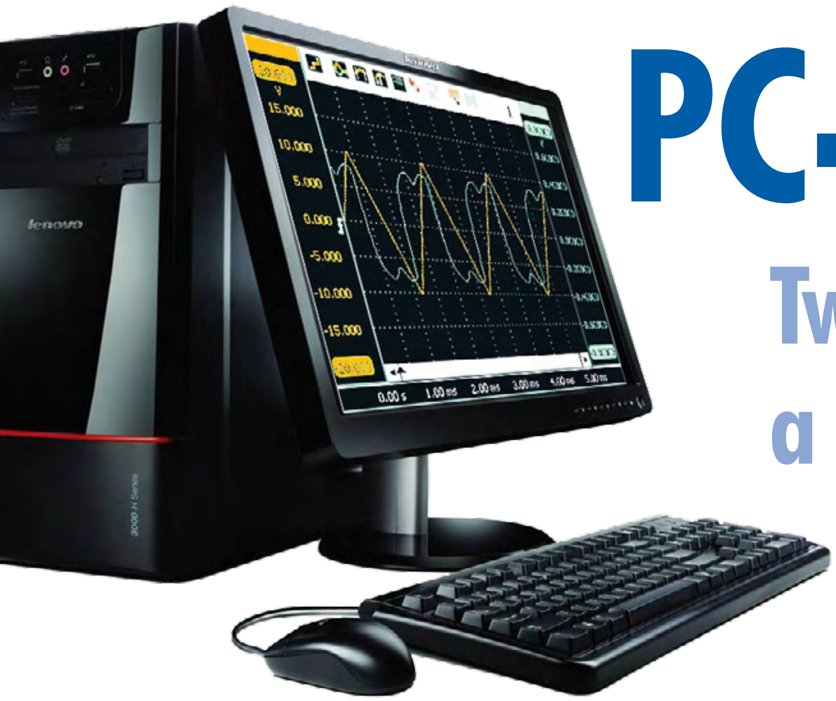
A greater number of more flexibly applicable I/O lines. An interrupt can be allocated to each pin. Output loading is restricted to
10 mA.

Interfaces

The integrated USART is, as with the SPI, fully duplex-capable. I2C now uses 10-bit addresses.

Interrupt and Events

Thanks to the integrated multi-level interrupt controller it is possible to prioritise interrupts to up to eight levels, which simplifies the more complex event-triggered applications considerably. In addition the CPU loading can be reduced by hardware routing.



PC-Based Ins

Two affordable USB o a built-in function ge

Harry Baggen (Elektor Netherlands Editorial)

A USB oscilloscope is a good alternative to a normal oscilloscope in a home lab. These instruments are handy and a good deal less expensive than a normal oscilloscope. In this article we examine a pair of two-channel units that also include a built-in function generator: the PicoScope 2203 and the Velleman PCSGU250.

USB oscilloscopes are available nowadays in all price ranges and quality levels. The idea behind these instruments is simple: why not use the computing power and display capabilities of a PC (which in most cases is already available) to display the signal waveforms? After all, a modern oscilloscope is not much more than a dedicated computer with a screen (not especially large in some cases) and a few buttons and knobs.

A USB oscilloscope module doesn't look very impressive on the outside: just a small box with a couple of BNC connectors on one end and an USB port on the other end for the link to the computer. As this module does not need controls and indicators, a powerful processor, a display or a power supply, it can be produced and sold at a much lower price than a full oscilloscope. Of course, you operate a USB oscilloscope with a keyboard and mouse, which is quite different from using a normal oscilloscope. This takes a bit of getting used to if you are accustomed to working with a 'real' oscilloscope.

For an electronics enthusiast who regularly puts together electronic circuits at home or a designer who sometimes works at home, a USB oscilloscope is a good, affordable alternative to a real oscilloscope.

For this test, we selected two inexpensive USB oscilloscopes that are also equipped with a built-in function generator, which gives you a complete 'mini-lab' on your bench. This is very handy for jobs such as measuring the frequency response of a circuit. Both units cost around 200 euros or the equivalent in pounds or dollars, which fits within an enthusiast's hobby budget. The main specifications of the two units are shown side by side in **Table 1**.

PicoScope 2203

The UK firm Pico Technology specialises in USB oscilloscopes and has a broad range of products in this category,

truments

scilloscopes with
nerator

including some with sampling rates of several gigasamples per second. The model we selected for our review, the 2203, is housed in a small blue-grey box slightly larger than a cigarette pack. There are three BNC connectors and an LED at the front, with a USB-B connector for the link to the PC at the back. The box is powered via the USB connection, so a separate AC mains adapter is not necessary.
In addition to the box, you receive a brief user guide, a CD-ROM with the software, a USB connection cable, and two probes that turn out to be of reasonably good quality. This is certainly not always the case with relatively inexpensive USB oscilloscopes.

After the PicoScope6 software and the drivers are installed, everything works smoothly. The layout of the user interface (**Figure 1**) takes some getting used to. It consists of a large display window for the waveforms, with narrow bars at the top and bottom for the controls. The settings for most tasks, such as selecting the horizontal or vertical scale, are configured using drop-down menus. This feels a bit strange at first if you're used to turning knobs on a normal oscilloscope. However, after a while it becomes fairly natural, and the overall layout remains clean and uncluttered with this approach.

The software provides three main functions: a two-channel oscilloscope, a spectrum analyser, and an arbitrary waveform generator (AWG). You can use the generator to produce a signal, feed it into a circuit or device, and then examine the signal in the circuit or at the output.
You can easily adjust the size of the PicoScope program window. The measured signal is normally shown in the whole window, but you can also display several windows at the same time. For example, you can show the measured signal along with its spectrum or show the two input signals in separate windows. In addition to the 'normal' oscilloscope display, there is a 'persistence' mode that imitates the operation of a storage oscilloscope, with several waveforms displayed superimposed using a sort of slow-decay effect, just like a storage oscilloscope with a long-persistence phosphor.

In the oscilloscope mode, the 8-bit resolution can be mathematically increased to as much as 12 bits. This gives the waveform a smoother appearance and makes everything look nicer, but you should bear in mind that it can cause certain details to be hidden. In practice, increasing the resolution by 1.5 to 2 bits proves to be enough to largely eliminate the originally visible effect of the sampling increments while still retaining all the relevant waveform details.
The built-in autoranging function for the input amplifiers and attenuators responds to changes in the input signal level quickly and reliably; there is rarely any need to change the settings manually.
All sorts of mathematical operations can be performed on the input signals, such as addition, multiplication, and sub-

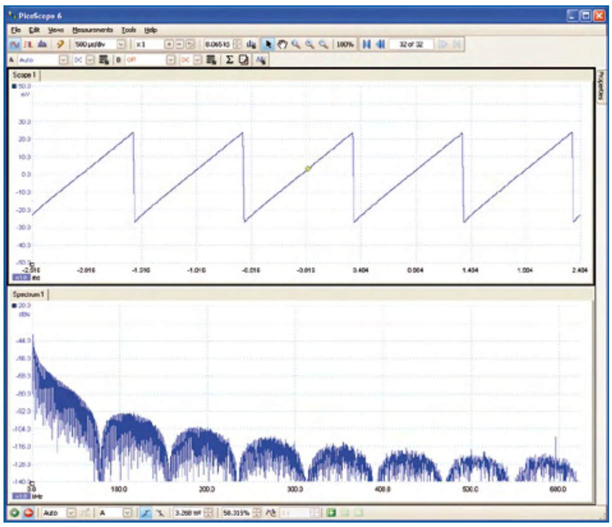


Figure 1. Screen shot of the PicoScope user interface. The number of windows is user-configurable.

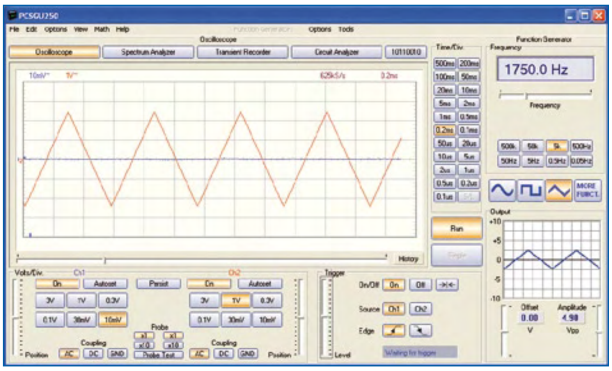


Figure 2. The user interface of the Velleman oscilloscope. Unfortunately, the screen size cannot be adjusted.

traction. You can even write your own formulas.
The spectrum analyser provides the most commonly used settings, such as the number of calculated points, selection of a measuring window (such as Blackman or Hamming), and a variety of scales. The FFT analysis runs especially fast on a modern PC, giving you the impression that you can monitor the composition of the measured signal in real time.
Finally, a few words about the user interface of the built-in generator. In contrast to the other functions, it is rather spartan. A button in one of the toolbars opens a small menu where you can set the frequency and select one of several waveforms (sine, triangle, sawtooth, etc. – most commonly used types are available). The output voltage can be adjusted in several steps from 125 mV to 2 V peak-to-peak, or you can enter a value directly, and the offset can be adjusted over a range of ±1 V. A sweep function is also available.
You can also use the generator to produce a user-defined waveform specified by a text file with a list of values. In addition, there is an 'Arbitrary' button that opens a separate window with a signal editor. It's very easy to use the mouse in this window to create a waveform or modify an existing waveform. This works very well.
All in all, the generator has a lot to offer, but the interface is not especially user-friendly and a separate mute button on the main toolbar would certainly be convenient.
In use, the PicoScope has a distinctly 'hands-on' feel. The display is clear and responds quickly — you almost feel like you're working with a normal oscilloscope. The built-in generator works well and produces very nice waveforms. In practice, this unit is a good alternative to a normal oscilloscope.

Velleman PCSGU250

Velleman supplies a large range of electronic products, but it is primarily known for its own electronic kits and instruments. The PCSGU250 is the smallest member of a new family of recently introduced USB oscilloscopes.
The housing has modern styling and is approximately twice as large as the PicoScope box. It is designed so it can stand upright next to a monitor or computer. Unfortunately, this makes it a bit unstable when a probe or set of probes is attached, since they sometimes get tugged in use. To help prevent the unit from tipping over, the designers fitted a sheet of lead at the bottom of the enclosure, and a triangular support for attachment to the rear of the enclosure is included.

The box also contains a USB cable, a short user guide, a mini-CD with the software, a probe, and a Cinch/BNC adapter. The probe is made by the same manufacturer as the PicoScope probes, and there's nothing wrong with it, but it's a pity that there's only one in the box.
Here again, installation of the PcLab2000LT software and the USB drivers was trouble-free on our Windows XP machine (the software for both units is only suitable for Windows systems).
The user interface of the program (**Figure 2**) is entirely different from that of the PicoScope. Just as with the enclosure, it appears that a deliberate effort was made to depart from the standard design. Whether you like this is a matter of taste. The left-hand part (the larger part) of the screen is reserved for the oscilloscope function, with a display window (not especially large) surrounded by all the controls. The control panel for the built-in generator is on the right.
Almost all user settings are made using buttons. For instance, there are six buttons for each input to select the input sensitivity, 21 buttons for the time base, and several additional buttons for the trigger settings and a few other things. Although this may appear convenient at first glance, it makes the overall layout very cluttered and somewhat confusing.
Despite the fact that the generator portion is designed in the same manner, it is well organised and easy to use because it has only a limited number of buttons.
One of the first things you notice when using the program is that that the size of the program and display windows cannot be changed. On a modern high-resolution monitor, they are simply too small.
The software has roughly the same features as the PicoScope software, namely a two-channel oscilloscope, a spectrum analyser, and an arbitrary waveform generator (AWG). It also has some extras in the form of a Bode plot generator (for automatic measurement of frequency and phase characteristics) and a transient recorder for making measurements over long time intervals.
Large buttons above the display select the individual functions. This area also has a button to select a special display mode for digital signals.
When making measurements in oscilloscope mode, it's a good idea to start by pressing the Autoset button, which

Table 1. Main specifications.

	PicoScope 2203	Velleman PCSGU250
Oscilloscope:		
Max. sampling rate	40 Msamples/s (1 ch.)	25 Msamples/s
	20 Msamples/s (2 ch.)	
Input bandwidth	5 MHz	12 MHz
Resolution	8 bits	8 bits
Internal buffer	8 Ksamples	8 Ksamples
Input range	10 mV/div to 4 V/div	10 mV/div to 30 V/div
Max. input voltage	20 V	30 V
Time-base range	200 ns/div to 20 s/div	100 ns/div to 2000 s/div
Generator:		
Frequency range	DC to 100 kHz	0.005 Hz to 500 kHz
Internal clock	2 MHz	12.5 MHz
Resolution	8 bits	8 bits
Output voltage	125 mV_{pp} to 2 V_{pp}	100 mV_{pp} to 10 V_{pp}
Offset	–1 to +1 V	–5 to +5 V
Output impedance	600 Ω	50 Ω

causes the system to try to find usable settings for the X and Y scales and triggering in order to produce a clear waveform display. The autoset function does not work continuously; if the input signal changes significantly, you have to adjust the settings manually or press Autoset again. Several times during the test, I wasn't sure whether I was looking at the current input signal or the samples stored in the buffer. In most cases, this doubt was resolved by again pressing the Run or Autoset button.
You can call up markers and use the mouse to position them in the display window. The markers can be used to measure various signal parameters (this can be done with the PicoScope by clicking points in the display window).
Several mathematical operations are possible, but they are limited to the most essential.
Like the PicoScope, the Velleman unit allows the resolution to be increased artificially with a few mathematical tricks, but it is not clear whether both devices use the same method for this.

During several practical tests, the Velleman oscilloscope proved to respond rather slowly to changes in the input signal. It looks like some sort of internal pre-processing must take place before the signal is shown on the screen. This must be taken into account if you want to make a series of measurements at different points in a circuit.
The spectrum analyser is certainly usable, but not as detailed or fast as the PicoScope spectrum analyser. It also has somewhat fewer configuration options.
The Bode plotter is an especially convenient addition to the program. You can use it together with the built-in function generator to record a frequency or phase characteristic quickly and automatically. You can set the start frequency and frequency range, the step size, and even a start delay. This is very handy for quick checks.
The generator portion is quite nice. Practically every desired frequency in the output range of the generator can be entered directly or adjusted in fine steps using a slider. The shape of the output waveform can be selected with a few large buttons, which also enable or disable the output signal. The output voltage can be adjusted from 100 mV to 10 V peak-to-peak, with an adjustable offset range of ±5 V.
In addition, the generator can produce special waveforms such as sin x/x and user-defined waveforms. For the latter function, there is a small utility program that helps you compose the waveform and shows a preview based on the entered numerical values.

Internal affairs

Both units have a few features that I haven't described, but the main features have been described here.
Naturally, we were also curious about the hardware inside the boxes. **Figures 3 and 4** show the internals of the PicoScope and Velleman units, respectively. The designs appear completely different at first glance, but if you examine the components that are used, you will find quite a few similarities. In both oscilloscope modules, the 'intelligence' is housed in a Xilinx Spartan FPGA, although different types are used in the two units. The Pico designers chose an XC3S250E (250 kgates), while their counterparts at Velleman chose an XC3S50 (50 kgates).

In addition to the FPGA, there is a microcontroller that looks after USB communication. Velleman uses a PIC18F2450 for this purpose, while Pico uses a Cypress CY7C68013A with an 8051 core and high-speed USB. The component used for analogue-to-digital conversion, which is largely responsible for the price and performance of a USB oscilloscope of this sort, is the same in both cases: an Analog Devices AD9288. Both units have several relays (or reed relays) for selecting the input range and DC/AC setting.
From this, we can conclude that in both cases you receive hardware with quite respectable performance and processing power for a price of around 200 euros.

The choice

After you've had a chance to play with these modules for a few days, you have to answer the question: which one do you prefer?

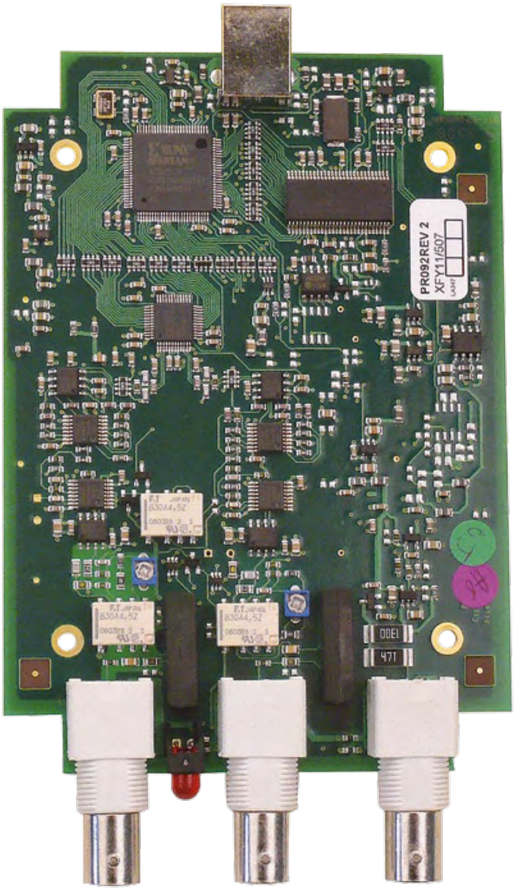


Figure 3.
The compact PCB in the PicoScope 2203 module.

As so often with such comparisons, the best solution would be a combination of the two. If I could, I'd like to have the oscilloscope portion of the PicoScope with the generator portion of the Velleman. Unfortunately this isn't possible, unless the two companies decide to join forces sometime in the future.

In terms of specifications, some differences between the two modules can be seen in both the oscilloscope and the generator portions, but they are not large enough to form the sole reason for choosing one or the other. The Velleman module has somewhat better specifications overall, especially the generator portion.

Nevertheless, for me the oscilloscope functionality is the most important aspect of a USB oscilloscope, and here the PicoScope is clearly better than the Velleman. It has fast response and a good autoranging function. In practical measurements, it works nearly the same way as a normal oscilloscope. Given this, I'm willing to accept the fact that the function generator software has a somewhat less convenient user interface (perhaps this could be addressed in a future software update?).

The Velleman oscilloscope is primarily attractive for audio enthusiasts, due to its built-in Bode plot generator.

All in all, although it's nice to be spoilt for choice, the trouble is, sometimes you have to make up your mind!

(090164-I)

Internet Links

PicoScope 2000 series:
www.picotech.com/picoscope2000.html

Velleman PCSGU250:
www.velleman.be/product/view/?id=377622

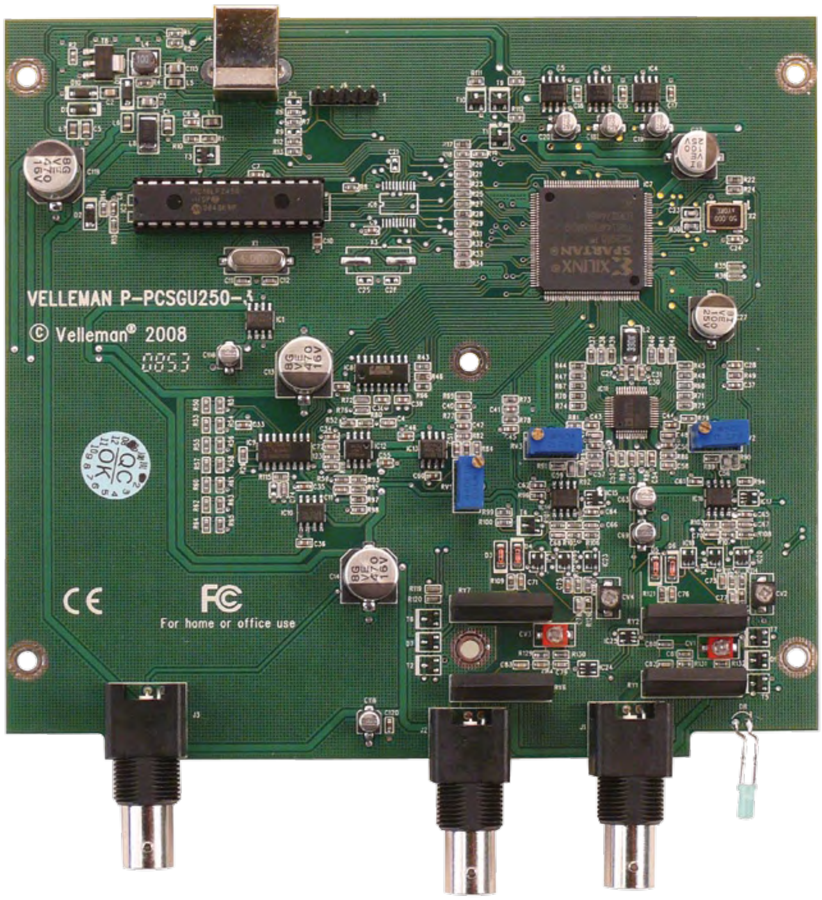


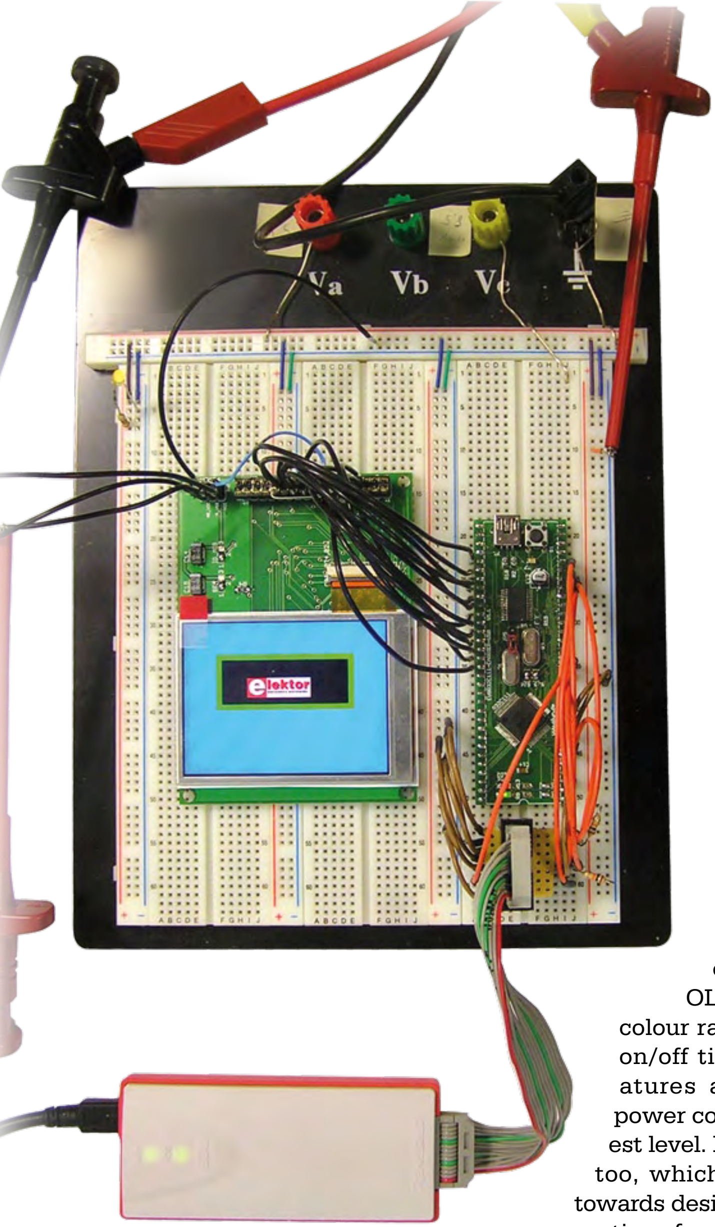
Figure 4.
The Velleman PCB has a similar design. Both circuits use a Xilinx FPGA.

R32C/111 goes OLED

From pure theory to proof-of-concept

Marc Oliver Reinschmidt (Germany) and
Martin Müller (Switzerland)

**OLEDs are hot news these days and rightly so, considering the new opportunities and applications they offer. It's not all plain sailing, however, since driving them by microcontroller presents developers with a number of challenges.
Continuing our series on the Renesas R32C, we trawl the theory to come up with a highly practical solution using the R32C carrier board.**



Organic LEDs — OLEDs for short — represent the latest in display technology and in the truest sense of the word offer 'visible' advantages over the now widespread TFT liquid crystal display (LCD).
Observers are struck immediately by the extremely broad angle of vision they offer at high contrast levels, thanks to self-luminous pixels that do not rely on auxiliary illumination. Besides an enormous contrast range, OLEDs also possess a broad colour range, extremely fast turn-on/off time even at low temperatures and manage to contain power consumption to a very modest level. Panel thickness is minimal too, which suits the current trend towards designs containing a high proportion of auxiliary electronics, such as touch-screens.

OLED vs. TFT

Figure 1 illustrates the construction of OLED displays in comparison to LCDs. Since OLEDs are made from organic material that produces its own pixel illumination there is no need for a backlight behind the entire display area. It's this feature that enables the low build profile and current-saver operation. As well as this, it means that unilluminated pixels are absolutely dark in OLEDs, whereas in LCDs unwanted light has to be masked. This function in LCDs — of either passing or suppressing light — works only in a rather rudimentary fashion. When we want the light to be visible we must reckon with some loss of light and when we are suppressing it, we must still contend with some residual light. With OLEDs the contrast ratio (from bright to dark) is better than in LCDs by some orders of magnitude.

Another factor is that switching light on and off in a liquid crystal display becomes more problematic as you approach low temperatures. Switching times increase considerably, which can lead to serious problems in some applications. With the OLED the deter-

mining factor is only the rising edge of the TFT transistor of around 50 μs. Furthermore we don't require any colour filters in the construction of OLEDs. The different materials employed light up directly in the colours red, green and blue. In this way we score not only improved efficiency but also colour intensity, particularly at reduced brightness.
The final improvement is that the pixels radiate light all directions. The viewing angle is thus unrestricted and the familiar 'toppling picture' effect of LCDs is unknown. The abolition of the optically unwelcome liquid crystal layer and colour filter — plus the improved contrast ratio — make a significant improvement to 'viewability' in brightly lit environments.

All these characteristics make high-performance designs very feasible, not just for these optical improvements but across the entire application. Taking battery life as an example, this can be improved, even when the display is in constant use — a long cherished desire of many developers and marketing people.
As with all new technologies, OLED displays bring changes as well as improvements, so right from the earliest stages of development designers have needed to be mindful of market-specific factors as well as purely technical aspects.
To fully exploit OLED technology all these characteristics have to be understood properly and factored into the design. We can deal with the purely hardware side rapidly: the only quirk needing a mention is the dual-polarity operating voltage of typically +4.6 V and – 4.4 V. The way OLEDs are driven is fundamentally similar to existing technologies, which we'll cover later on.

Summing up, what all this does mean is that how the display produces light must be considered from the very outset, with serious thought about 'what', 'when', 'where', 'under which conditions' and 'how bright really'. For OLEDs these factors are of vital significance, unlike the LCD with its always-on backlighting across the whole screen area and colours generated 'passively' by colour filters. Incidentally, the luminous organic material in OLEDs loses brightness over time in use — as do the illuminants in CRT screens and the LEDs in LCD displays. The current specification of usable life down to 50 per cent brightness loss has a bearing here as well. Direct comparisons are pretty meaningless, however, since with OLEDs the manner of use has a clear influence on these metrics. For this reason the specification for OLEDs is set out for mixed-mode operation using all pixels and all colours: measurement taken with hundreds of randomly selected images produce an average performance of 30 per cent, which is used as the basis for this specification.

For other types of usage we need to examine the situation more precisely; judicious operation to minimise excessive use may lead to significantly higher values. This is because unlike LCDs, it is only the active pixels in OLEDs that deteriorate in brightness.

Optimisation

A solution arises from the extremely high contrast available from OLEDs. If we considered poor viewing conditions (such as in direct sunlight) and wanted to state what which maximum brightness might be optimal, we could obviously 'turn down the wick' under normal conditions, without detriment to visibility of display and to the advantage of extended operational lifespan. By skilled design, for example by vertical scrolling or sideways

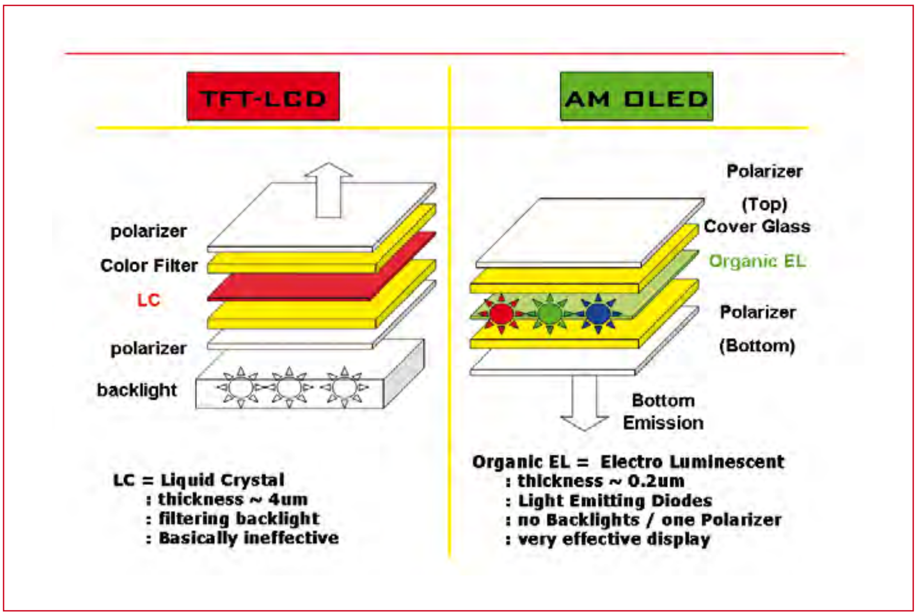


Figure 1. TFT LCD and AM-OLED (Active Matrix OLED) construction compared.

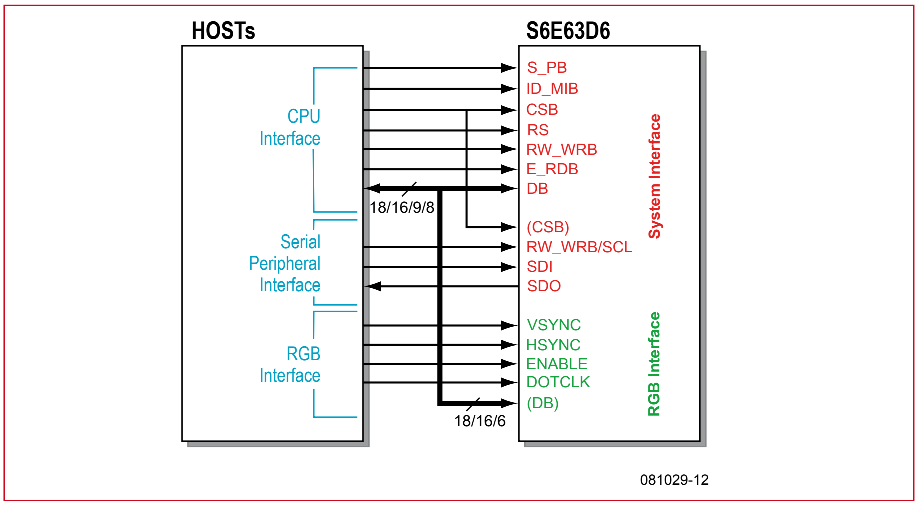


Figure 2. The S6E63D6OLED controller from Samsung is located 'on glass' integrally with the display and provides a variety of control interfaces.

shifting of static images, the number of pixels employed could be increased to increase the life span of individual pixels.

The resulting specification is thus directly dependent on the way we plan to use the display, meaning it could vary pronouncedly. The way OLEDs are influenced and behave under differing parameters is not always entirely linear, which is why Glyn has developed various sample set-ups to show up what users might expect under specified conditions of current consumption and lifespan. These can be used for operational specifications and also for interpreting test results.

OLEDs in operation

As already mentioned, the OLED requires a dual-polarity power supply. For this reason the test set-up shown has the display mounted on an OLED adapter board with the extra power supply components onboard. Special voltage regulators are used, developed for the particular requirements of OLEDs.

As **Figure 2** shows, the smaller displays (in our case a 2.4-inch OLED from CMEL) can be driven in a number of ways. In one case the display is provided with an SPI interface and in the other with a parallel data bus. In our case we are using the parallel input. A variety of bus widths is available. According to the depth of colour levels and data transfer rate required, we can select between 8, 9, 16 or 18 bits wide. To keep things simple, our sample set-

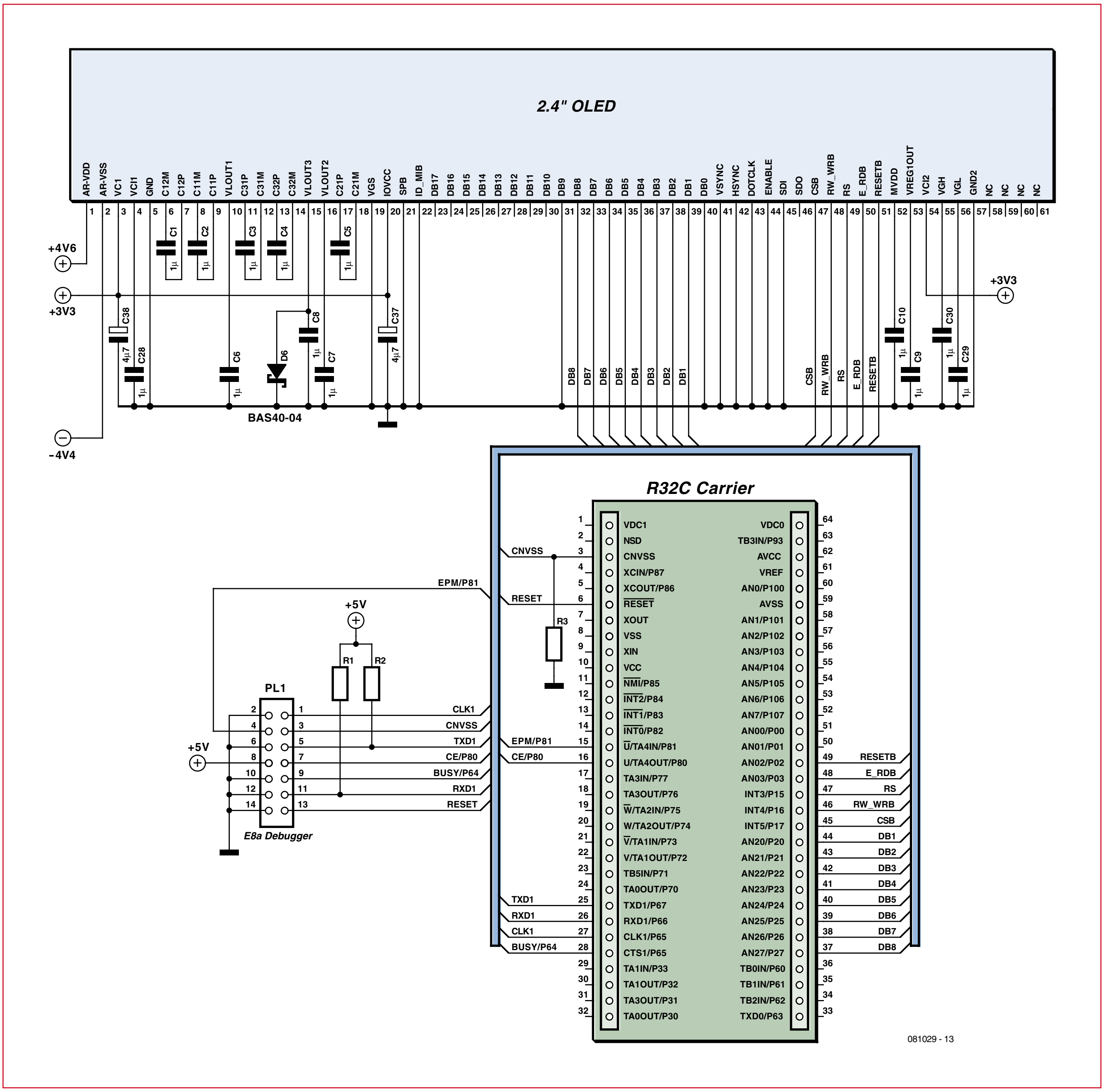


Figure 3. Circuit diagram showing the connections between the R32C carrier board, E8a debugger and 2.4-inch OLED display.

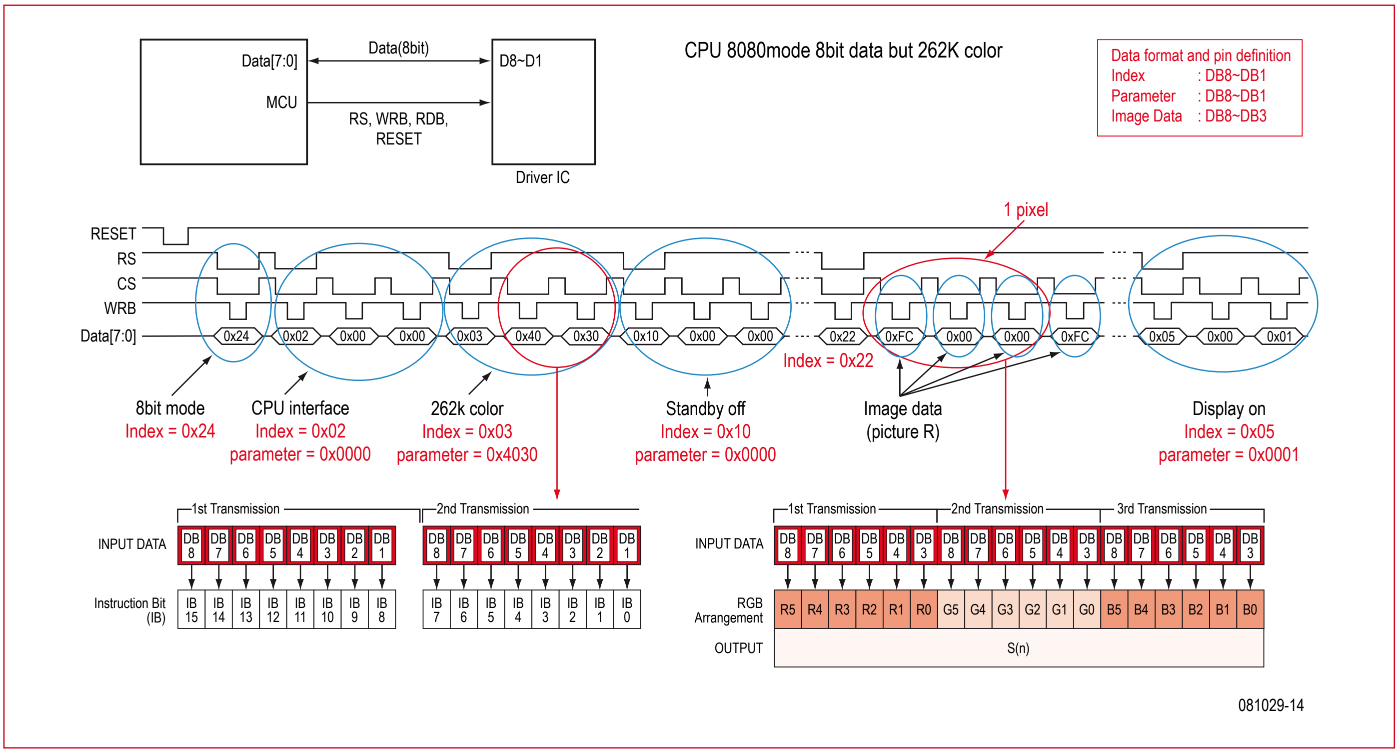


Figure 4. OLED display initialisation in 8-bit mode.

up employs 8-bit operation. Besides the data lines the display requires five additional control lines.

The schematic in **Figure 3** shows how the OLED and the E8a debugger are connected. The debugger has already been mentioned briefly in the previous article of this series. Using this cost-effective tool we can carry out debugging with much greater ease than with the serial KD100. The E8a is incorporated in the HEW, so we can view everything on a single screen. The E8a is applicable to all Renesas controllers of the M16C family, from the R8C right up to the R32C.

Getting back on-topic, the OLED requires just a few additional capacitors, which are already provided on the OLED adapter board from Glyn. This arrangement matches with the one on the R32C application board from Elektor that we are using and will be the subject of the next two articles (see inset 'Coming soon').

The power supply is not included in Figure 3, as there are many voltage regulators that can deliver a negative voltage of –4 V for V– to pin 2 of the OLED. That's it for the hardware need to control the OLED in our sample system.

Initialisation

To drive the display in 8-bit parallel mode we need to divide the address and data information, then transfer them sequentially. Taking for example the three primary colours red, green and blue, transferring them in 8-bit mode, there are differing configurations according to the number of colour levels to be defined (see **Figure 4**).

Our sample software transfers colours using 5-bit colour levels. For all three colours together this adds up to 15 bits per pixel.

The first step is to initialise the display, which requires the supply voltages to be powered up in a specific three-stage sequence.

We begin by activating the 3.3 V for the logic modules. Next we initialise the display and in the third step we connect the bipolar voltage for the pixels. Powering up the display without going through this three-stage sequence can damage it, since then all the pixels will light up in a white of undefined brightness, causing dangerously high current flow.

The initialisation routine is given in **Listing 1**. Only in the last line is the negative voltage regulator activated. Following successful initialisation the OLED will be black overall.

Rectangle and logo

Even if 'black is beautiful' we'd prefer to see something more interesting on the display. So let's look how easy it is to produce a simple rectangle in a single colour (see **Listing 2**).

The S6E63D6 display controller from Samsung, located on-glass integrally with the display, enables the use of 'frames'. Within this frame definition system we can use the 'auto-increment' function for the pixel addresses, so that all we need to specify is the parameters for the colours of the rows. This takes place in the lower loop, which defines the data by means of the function *Pixel_out(r,g,b)*.
We'll begin by defining the start and end points of our frame, using the functions *Index_out* and *Parameter_out*. These values need to be included in the corresponding register of the displays.

Just like the long-established alphanumeric dot-matrix displays, the drive commands are distinguished from the actual data by level and control line. A High level on the RS line specifies data

Listing 1

Initialisation of the OLED display

```
void init_S6E63D6_240X320_8Bit_80Mode(void)
{
 unsigned long i;

 Init;

 WRB=1;
 RDB=1;
 NCS_H;
 NRESET_L;
 NRESET_H;

 Index_out(0x24);
 Index_out(0x02);
 Parameter_out(0x0000);

 Index_out(0x03);
 Parameter_out(0x4120);    // 262k colour mode
   (3Bytes))  SS=1 0x4031

 Index_out(0x10);
 Parameter_out(0x0000);
 Index_out(0x05); // display on
 Parameter_out(0x0001);
 Index_out(0x22);
 clearscreen();
 Power=1; // switch -NCP5810- power supply on for
   display
}
```

Listing 2

Define frames and fill with colour

```
void OLED_RECT(uch HSA, uch HEA, uin VSA, uin VEA,
   uch r, uch g, uch b)
{
   unsigned long i;

   ulo x;
   /*** Set Window address ***/
   Index_out(0x35);
              //Start point VSA
   Parameter_out(VSA);


   Index_out(0x36);
              //Start point VEA
   Parameter_out(VEA);


   Index_out(0x37);
              //Start point VSA
   Parameter_out((int)((int)HSA<<8) | (int)HEA);
   //HSA/HEA - here y
//       for(i=0;i<0x500;i++) _asm("\tNOP");

   /*** Start address set ***/
   Index_out(0x20);

   Parameter_out(HSA);
   Index_out(0x21);

   Parameter_out(VSA);

   /*** Index write ***/
   Index_out(0x22);
//       for(i=0;i<0x500;i++) _asm("\tNOP");
   for(x=0;x<((ulo)VEA-(ulo)VSA+1)*(ulo)(HEA-
   HSA+1);x++)
   {
     Pixel_out(r, g, b);

   }
    Index_out(0x00);
}
```

Listing 3

Operating instructions

```
void Index_out(unsigned char wert)
{
DB_OUT;

DB=value;
RDB=1;
RS=0;
NCS_L;
WRB=0;
WRB=1;
NCS_H;
}
```

Listing 4

Display data output

```
void Parameter_out(unsigned int value)
{
DB_OUT;
RDB=1;
//high value Byte
DB=value>>8;
RS=1;
NCS_L;
WRB=0;
WRB=1;        // accept data
NCS_H;
//low value Byte
DB=wert;
NCS_L;
WRB=0;
WRB=1;       //accept data
NCS_H;
}
```

Listing 5

Display pixel output

```
void Pixel_out(uch r, uch g, uch b)
{
DB_OUT;
RDB=1;
DB=r<<2;
RS=1;
NCS_L;
WRB=0;
WRB=1;
NCS_H;

DB=g<<2;
NCS_L;
WRB=0;
WRB=1;
NCS_H;

DB=b<<2;
NCS_L;
WRB=0;
WRB=1;
NCS_H;

}
```

transfer, whereas a Low level indicates an instruction (see **Listing 3**).

This straightforward arrangement lets us transfer control command and parameters to the display. The 16-bit instructions are split into two bytes in the function shown in **Listing 4**.

This function enables data to be transferred to the display. As we are dealing with words of 16 bits, the high value byte is sent first and then the low value one. Changing level on pin WRB transfers the data from the display controller.

Finally here is the *Pixel_out* function (**Listing 5**). Each successive colour is passed to the display, after which the frame is initialised using the *OLED_RECT* function.

You can extract all other functions from the source code, which can be downloaded from the website under this article reference (www.elektor.com/081029).

Listing 6 gives the code you will require finally for creating the rectangle and a small logo on the display.

The leader photo shows the carrier board with E8a debugger connected and adapter board for the 2.4-inch OLED.

The logo on the OLED can be converted from a bitmap file into an array file with the aid of some free tools and then included direct into the source code.

The sample files can be downloaded from Glyn (www.glyn.de/r32c) as well. You can use either the Renesas development environment or the IAR Embedded Workbench from IAR with this project.

(081029-I)

Our authors

Marc Oliver Reinschmidt is an application engineer at Glyn's head office at Idstein, Germany, and has special responsibility for the M16C/R32C microcontroller family. In the following articles he will show how the universal application board developed by Elektor for the R32C carrier board can be programmed to act as an oscilloscope.

Martin Müller is a Field Application Engineer for display products and a specialist in the field of OLED displays. He is based at Glyn's Swiss office in Esslingen.

Coming soon

Elektor R32C application board with:

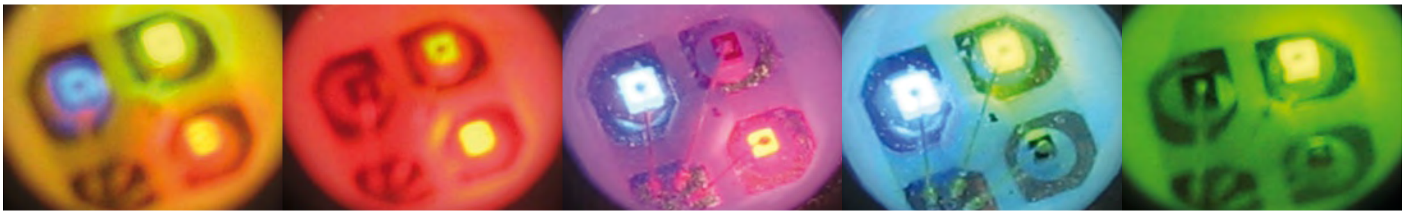
- 2.4-inch OLED display
- SD card reader interface
- I²C
- Slot for LAN module WIZ812MJ
- 2-channel oscilloscope input
- Rotary encoder with switch
- 4 LEDs
- 4 switches

Listing 6

```
/**************************************************/
/*
                        */
/*  FILE        :R32C_example.c      [R32C_sample.
   c]                   */
/*  DATE        :03.13.2009
                        */
/*  DESCRIPTION :main program file.
                        */
/*  CPU GROUP   :R32C111
                        */
/*
                        */
/**************************************************/

#include "sfr111.h"
#include "hwsetup.h"
#include "oled28.h"
#include "elektor.c"

void main(void)
{
unsigned long i;
   ConfigureOperatingFrequency();

   for (i=1;i<10000;i++);

   init_S6E63D6_240X320_8Bit_80Mode();

   {
   OLED_RECT(00,240,00,320,0,0,0xff);  //blue rect
   OLED_RECT(100,190,50,270,0,0xff,0);  //green
   rect
   OLED_RECT(110,180,60,260,0,0x00,0);  //black
   rect
picture(121, 151, 101, 200, elektor);

   while(1);
}
```

RGB LED Driver

High resolution colour control with the AAT3129



Fred Splittgerber (Germany)

Seemingly straightforward projects can turn into a 'money pit' or 'component graveyard' if you are not careful. This can easily come true if you intend driving colour LEDs in RGB mode with infinitely variable colour mixing and individual control over the brightness of each LED. Conventional control circuitry tends to produce quite bulky systems too. On the other hand, using a microcontroller and a specialised IC keeps the space footprint under control and eliminates all the uncertainties...

Specifications

- RGB LED driver module for universal application
- Straightforward serial control
- Operating voltage from 2.7 V upwards thanks to charge pumps
- LED maximum current (total) 40 to 180 mA
- 4,096 colours
- 16 stages of total brightness
- Low-interference operation at constant frequency
- Flicker-free illumination thanks to 1 MHz PWM frequency

Listing every possible application for infinitely variable control of individual RGB LEDs is an impossible task. What is not in dispute is the fact that the variety of RGB LEDs (one each in red, green and blue on a single carrier or in a single package) has risen significantly in recent years. Anyone planning to put these colourful semiconductor light sources to practical use needs to think carefully about the control electronics to be used.

RGB control

The rules covering LEDs in general apply also to RGB LEDs, the most fundamental being that LEDs need powering with constant current rather than constant voltage. This is because the threshold voltages of LEDs are strictly temperature-dependent and without constant current, stable operation is impossible. Simple logic indicates that achieving infinitely variable (step-free) current setting requires the use of infinitely variable current sources. If energy saving is important, then the recommended approach is to use switched constant current sources with adjustable duty cycles.

An important characteristic of RGB LEDs to note is that as a result of their physical structure, red, green and blue LEDs display differing forward voltages, ranging from less than 1.5 V for red LEDs up to nearly 4 V for blue ones. Without some kind of intelligent switching arrangement it's obvious that significant energy losses will arise if your driver circuitry provides the same voltage for R, G and B LEDs (which will be far too high for the red ones). Pulse-width modulated current sources are totally unsuitable, especially in battery powered applications. But before you bash your brains in looking for suitable solutions based on switching regulators, take it easy. Industry has already come up with a solution for this problem and embedded it in silicon.

AAT3129

As well as its switching regulators and other power supply ICs, the firm

Analogic Tech has lately brought out a whole range of chips intended to simplify the operation of all manner of LEDs. And with the IC AAT3129 [1] every control problem that might occur with RGB LEDs has been eliminated with a single chip.

The IC has a serial digital control input and integrated charge pumps with factors of 1, 1.5 and 2, enabling it to operate with supply voltages from 2.7 V to 5.5 V. Among other features are built-in logic for avoiding thermal overload and — important for battery operation — a standby mode with current consumption typically less than 0.1 μA. In operation the IC draws around 1 mA. Maximum current for the LEDs — shared across all three LEDs — can amount to 180 mA. LED brightness is set individually in 16 logarithmic stages each, producing in total $2^{4 \times 3} = 4,096$ different colours. On top of this there are 16 steps of overall brilliance.

The IC operates at a clock rate of 1 MHz and with 12 pins and dimensions of just 2.4 × 3.0 × 1 mm it is extremely compact. The only external components required are four small 1 μF ceramic capacitors. A functional diagram is given in **Figure 1**. All we need to complete the circuit is a small microcontroller to provide the AAT3129 with data.

AS²Cwire

Data for the AAT3129 is presented in the Simple Serial Control (S²C) protocol, AS²Cwire [2]. The S²Cwire™ single-wire interface offers a very straightforward control technique for programmable power IC devices, using just a single wire. Data is transmitted as a series of negative-going pulses having a length of between 50 ns and 75 μs. Between pulses the level remains High for up to 500 μs. Greater values are treated as separator signals between pulse trains (see data sheet [1]). Sequences with 16 to 21 pulses are interpreted as addresses for the registers R, G, B, T (total intensity) and M (operational mode) (see **Table 1**).
The sequence that follows afterwards with 1 to 16 pulses is the actual data. To summarise, the address follows a High level of >500 μs, after which comes the data to be transmitted. Whether a value is to be executed straightaway or synchronised only once all the colour values have been defined, depends on the value placed in the M register.

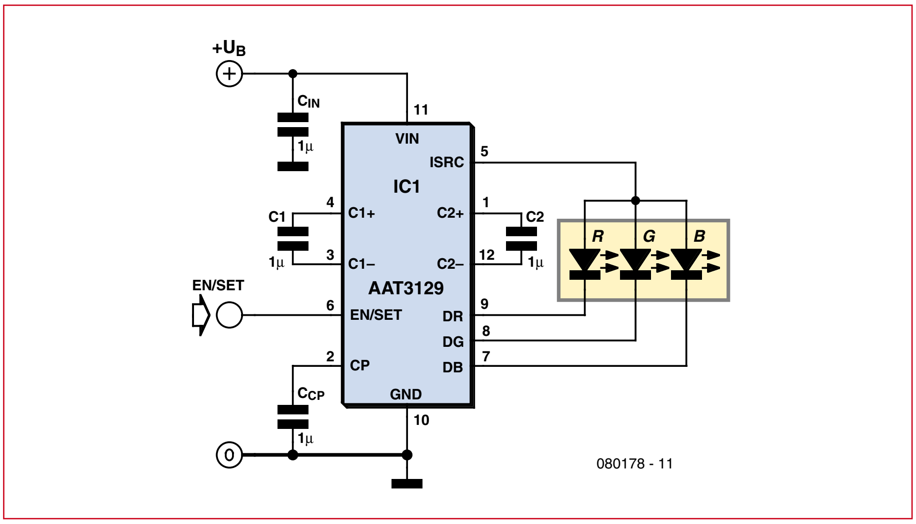


Figure 1. Block diagram of an RGB LED driver using the AAT3129.

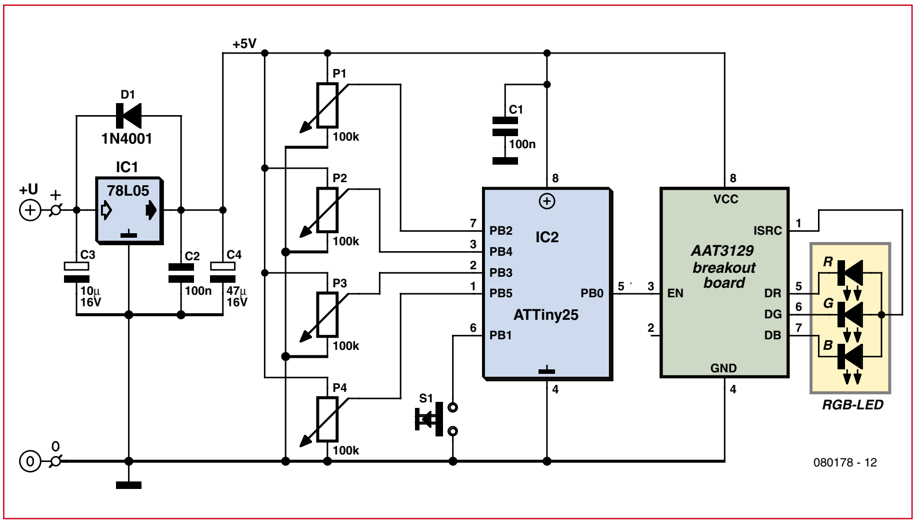


Figure 2. Control diagram with microcontroller and connector for the breakout board.

Control driver

In order that we can select the colours and the overall brightness easily with rotary or slider pots, we need to use another small microcontroller with a built-in multi-channel A/D converter. This transforms the analogue potentiometer values into corresponding digital values, converts them and passes

Table 1

Register	Address	Range of values	Meaning
R (Intensity, red)	17	1-16	1: unlit 16: maximum brightness
G (Intensity, green)	18	1-16	
B (Intensity, blue)	19	1-16	
T (Overall intensity)	20	1-16	1: maximum brightness 16: darkest state
M (Operating mode)	21	1-2	1: Value is converted immediately 2: Value is converted after writing to the T registers

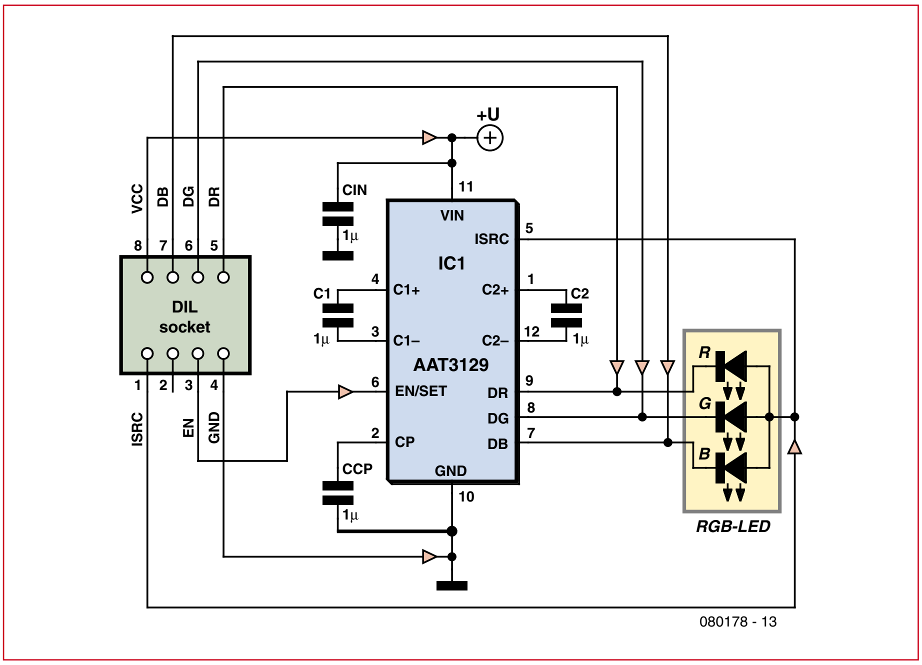


Figure 3. The circuit of the breakout board consists of just the AAT3129, four capacitors and an RGB LED if required.

the result to the driver IC. The small 8-pin ATtiny controllers from Atmel make this task a breeze. Four pins are configured as analogue inputs for R, G, B and T, whilst a changeover switch defines the operating mode. Apart from the two pins for +UB and ground, just one pin remains, the serial output that controls the AAT3129 chip. The software for the chosen microcontroller type, ATtiny 25, is covered here in a separate **inset**.

Control circuitry

The control circuitry can be seen in **Figure 2** and described in very few words. Apart from a 5-V voltage regulator this comprises a microcontroller, four 100-kΩ pots, a switch (operating with a pull-up resistor integrated in IC2), an IC socket for connecting a breakout board and finally another RGB LED if required. The breakout board is a small plug-in board equipped with the AAT3129 chip and the four capacitors mentioned previously.

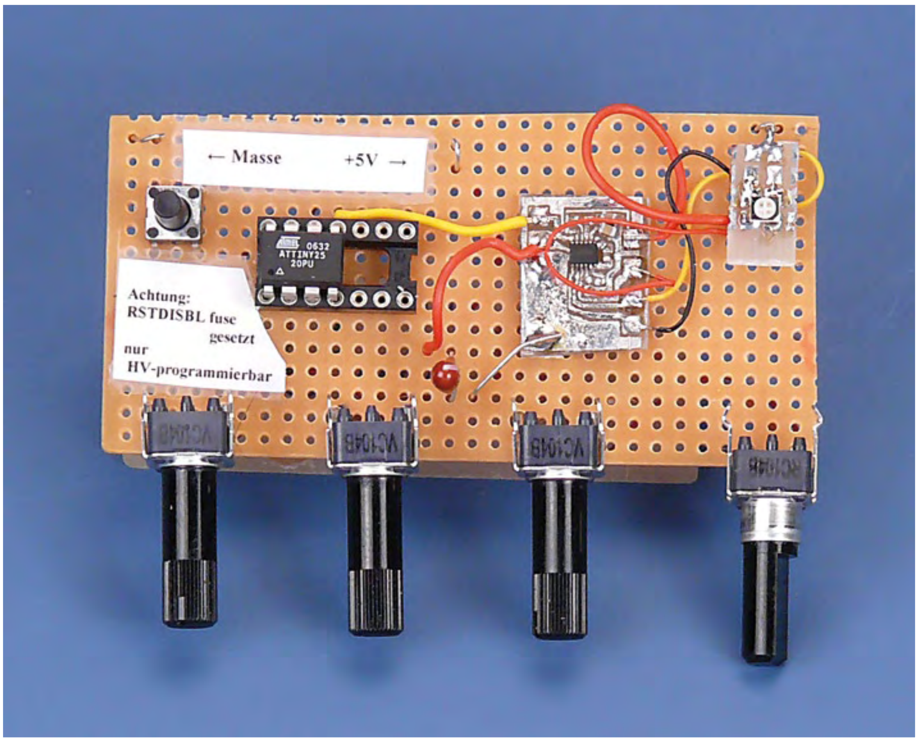


Figure 4. The author's trial set-up looks like this, with the breakout board and microcontroller built on a piece of perfboard.

The circuit is so simple that you can build it on a scrap of perfboard without difficulty. As not everybody feels comfortable with soldering the 'fine-pitch' arrangement of the pins of the AAT3129, the author hit on the idea of laying out the breakout board mentioned for the AAT3129 complete with the capacitors (and optionally an RGB LED in PLCC4 form factor) to enable it to be plugged simply into a DIL IC

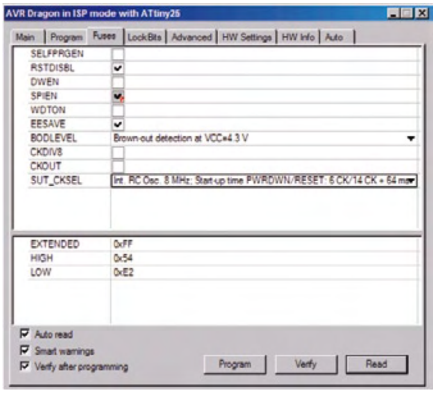


Figure 5. With the ATtiny it is vital to get the fuse settings correct, as this screenshot illustrates.

socket or a breadboard device or else soldered onto some 2.54 mm (.1 inch) pitch Veroboard or perfboard. The circuit of this breakout board is shown in **Figure 3**. You can download the layout files for this tiny board in KiCAD and Gerber format at the web page for this article on the Elektor website. This mini PCB does not have to be used exclusively with the microcontroller recommended here and can also be integrated into other circuits without difficulty. **Figure 4** shows a hookup corresponding to the circuit in Figure 2 in which this little PCB is placed onto perfboard along with an ATtiny25 in a DIL package.

Last but not least

The use of a 78L05 (IC1) means the whole circuit can be powered using any direct voltage between 7.5 and 10 V. On account of the broad supply

About the author

Fred Splittgerber has been involved with hardware-specific programming continually since the first 8-bit CPUs appeared. He works as a technical author and translator.

voltage range of the AAT3129 and ATtiny25 chips, you could also use a 3.3-V voltage regulator — or even omit the voltage regulator altogether and power the rest of the electronics direct from a stabilised 3.3 V supply. In this case the fuse for the brown-out detector needs to be matched correctly.

On the breakout board we have provided a socket for connecting an RGB LED as well as room for soldering a PLCC4 RGB LED direct. However, you should never connect two LEDs in parallel, as otherwise the necessary current splitting will not be achieved.

If the switch connected to port B1 is closed, then you will activate the colour transformation mode preset in the firmware, in which the overall brightness is set by variable resistor P4. When the switch is open circuit the RGB LED illuminates with constant brightness with the colours set with pots P1 to P3.

Source code of some sample firmware for the ATtiny25 is available for downloading free of charge from the Elektor web page [5] (see also the inset 'Software').

(080178-I)

Software

There are two important things to note in this software written in C. Firstly, the reset pin of the ATtiny25 is used as an input pin, meaning that the fuse value RSTDISBL (see Figure 5) must be defined. Once this has been done, no further SPI programming is possible. Special care is vital, as your nice new controller will turn out useless if it contains any software errors. Secondly, during colour changes the software optimises the display of uncommon greyscale hues during the transformation of one colour to another. Colour saturation is calculated according to the HSV colour model [3] and the transition between colours of low saturation is accelerated.

The serial control signal for the AAT3129 is generated using the following function:

```
void tx_pulses(uint8_t n){
  for (i=n; i>0; i--) {
    PIN_AAT=1<<AAT_BIT;
    PIN_AAT=1<<AAT_BIT;
  }
}
```

This generates 'n' pulses on Bit 'AAT_BIT' of output 'PORT_AAT'. This connection needs to present a 'high' level whenever no data is being transmitted. Here is the initialisation of the Port of the ATtiny25:

```
#define PIN_AAT   PINB
#define AAT_BIT PB0
#define PORT_AAT  PORTB
#define DDR_AAT DDRB

PORT_AAT|=1<<AAT_BIT;// Output AAT_BIT = 1
DDR_AAT|=1<<AAT_BIT; //  AAT_BIT is Output
```

Writing to PINx toggles the polarity of the corresponding Bit of PORTx. Instead of writing `'PIN_AAT=1<<AAT_BIT'` twice we can also write:

```
PORT_AAT&=~(1<<AAT_BIT); // AAT_BIT= 0
PORT_AAT|=1<<AAT_BIT;// AAT_BIT= 1
```

With a controller clock rate of between 14 kHz and 20 MHz both methods produce the required negative-going pulses with a duration of from 50 ns to 75 µs.

For the pause that indicates the separation signal between pulse sequences you can set one of the timers or implement a delay routine. In the latter case writing the intensity '10' to the red LED looks like this:

```
#include <util\delay.h>
//…
#define CHANNEL_RED 17
//…

tx_pulses(CHANNEL_RED);// CHANNEL_RED pulse select RED register
_delay_ms(0.5);
tx_pulses(10);// Data RED register (brightness 10 from 1-16)
_delay_ms(0.5);
```

If the GCC compiler [4] is used the optimisation option '–O2' must be used.

Internet Links and Literature

[1] AAT3129 Data Sheet:
www.analogictech.com/products/digitalfiles/AAT3129.pdf

[2] AS2Cwire application notes:
www.analogictech.com/resources/applications/appnotes/AN110_S2Cwire_TLAT.pdf

[3] HSV Colour Space and Colour Space Conversion:
http://en.wikipedia.org/wiki/HSV_color_space
http://www.cambridgeincolour.com/tutorials/color-space-conversion.htm

[4] GCC Compiler for AVR:
http://winavr.sourceforge.net

[5] www.elektor.com/080178

Hexadoku

Puzzle with an electronics touch

Sure, we've seen Sudoku puzzles that can be solved online and yes there are many nifty programs around to crack these brain teasers but Elektor's Hexadoku should remain a pencil-paper-brain exercise. Do participate! All correct solutions we receive enter a prize draw for an E-blocks Starter Kit Professional and three Elektor Shop vouchers.

The instructions for this puzzle are straightforward.

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 4x4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

All correct entries received for each month's puzzle go into a draw for a main prize and three lesser prizes. All you need to do is send us the numbers in the grey boxes. The puzzle is also available as a **free download** from the Elektor website

SOLVE HEXADOKU AND WIN!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for an

E-blocks Starter Kit Professional

worth **£300**

and three **Elektor SHOP Vouchers** worth £40.00 each.

We believe these prizes should encourage

all our readers to participate!

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

PARTICIPATE!

Please send your solution (the numbers in the grey boxes) by email to:

hexadoku@elektor.com - Subject: hexadoku 05-2009 (please copy exactly).

Note: new email address as of this month!

Include with your solution: **full name and street address.**

Alternatively, by fax or post to: **Elektor Hexadoku**

Regus Brentford - 1000 Great West Road - Brentford TW8 9HH

United Kingdom - Fax (+44) 208 2614447

The closing date is **1 June 2009.**

PRIZE WINNERS

The solution of the March 2009 Hexadoku is: **813D2**.

The **E-blocks Starter Kit Professional** goes to:

J.C. Launay (France).

An Elektor SHOP voucher worth £40.00 goes to:

Jürgen Ackelbein (Germany); Eloi Dranka Jr (Brazil); Doug Blansit (USA).

Congratulations everybody!

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

(c) PZZL.com

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

Elektor Mini Crescendo (1984)

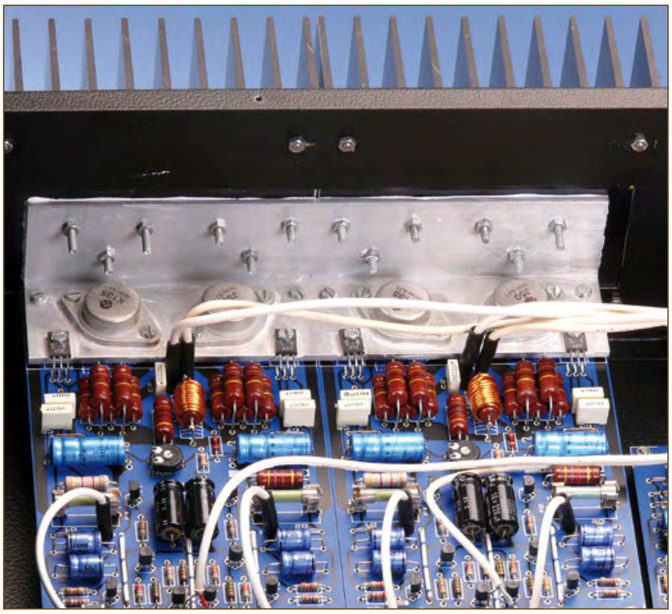
Eric Bogers
(The Netherlands)

'With a good amplifier, all you hear is the music.'

In December 1982, Elektor surprised the world with a top-class MOSFET final amplifier boasting a hefty output power: the Crescendo. With a rated power of 180 watts into 4 ohms and a harmonic distortion level that remained well below 0.01% over the frequency range of 20 Hz to 20 kHz, it was a design that could please even the most pampered ear.
However, such a combination of quality and power did not come cheap. With four MOSFETs per channel (the famous Hitachi 2SK135 and 2SJ50) and with each channel powered by a separate DC supply with a heavy-duty (and correspondingly heavy) toroidal transformer and several 'fat' electrolytic filter capacitors, it added up to a tidy sum. A complete Crescendo would easily cost upwards of 250 pounds or the equivalent in dollars at the time.

As a result, there was a flood of requests for a design with similar features but a more modest price. They found a ready ear, and in May 1984 Elektor proudly presented the 'baby brother' of the Crescendo: the Mini-Crescendo – although here 'mini' is only relative, since two channels rated at 70 watts each into 4 ohms is still more than enough to let you neighbours share in your musical pleasure.
With 'only' two power MOSFETS per channel, a single power transformer, and somewhat smaller and less expensive electrolytic capacitors, this version was within my budget. I purchased the components in early 1987, and it was all put together a few weeks later. Thanks to the carefully designed printed circuit board, construction did not present any insurmountable problems, but there were two aspects of the project that I will never forget.

The first was fitting the output transistors, which was rather difficult. The circuit board of the final amplifier was attached to a generously sized heat sink by an aluminium angle and a few screws, and the transistors were bolted to the aluminium angle with their leads passing through carefully drilled openings in the aluminium to the holes in the PCB. As the transistors had to be electrically isolated from the aluminium but at the same time fitted to it with the lowest possible thermal resistance, it took a few hours to get this job right.
The second aspect was the power supply. When the time came for final assembly, it was naturally the first part of the amplifier to be fitted in the enclosure and tested. After carefully checking the wiring, I switched on the power, which fortunately did not result in any explosions or clouds of smoke, and measured the output voltage. The no-load voltage was approximately ±65 V — exactly as specified. However, I forgot to connect a bleeder resistor across the capacitors to discharge them after the power was switched off, probably because my thoughts were already on a well-deserved beer. When I resumed work on the amplifier the next evening, I received a rather strong shock (something that has probably happened to every electronics hobbyist at some time). Obviously the capacitors did a damned good job of holding their charge.



I built my Mini-Crescendo, complete with the combined switch-on delay and DC protection circuit described in the January 1983 issue of Elektor, into a sturdy 19-inch rack. During the course of 1987, I augmented my sound system with the Preamp described in the December 1986 and January 1987 issues (probably the best high-end preamplifier ever to leave the Elektor labs) and the unsurpassed class-A headphone amplifier described in the February 1983 issue.
Now, twenty-five years after the publication of the original design and twenty-two years after I assembled the various components, this system (along with two Magnat Viva loudspeakers, an excellent CD player, and an outstanding turntable) is still in service. After all these years, I still take considerable pleasure in reading Elektor, and I still listen to music from my Mini-Crescendo every day.

Incidentally, the SK135 and SJ50 output transistors were discontinued many years ago, but the UK mail-order company LittleDiode (www.littlediode.com) apparently still has a good stock on hand. That's a comforting thought.

(081096-I)

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

ELEKTOR SHOWCASE

products and services directory

www.elektor.com

OBD2CABLES.COM

http://www.obd2cables.com

- Thousands of OBD cables and connectors in stock
- Custom cable design and manufacturing
- OBD breakout boxes and simulators
- Guaranteed lowest prices
- Single quantity orders OK
- Convenient online ordering
- Fast shipping

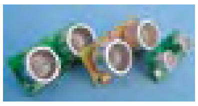
Visit our website, or email us at:
sales@obd2cables.com

ROBOT ELECTRONICS

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

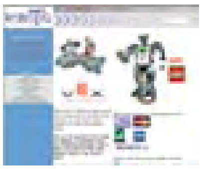
Advanced Sensors and Electronics for Robotics

- 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!

- MeccanoTM Compatible
- Computer Control
- Radio Control
- Tank Treads
- Hydraulics

Internet Technical Bookshop,
1-3 Fairlands House, North Street, Carshalton,
Surrey SM5 2HW
email: sales@robotiq.co.uk Tel: 020 8669 0769

SCANTOOL.NET

http://www.scantool.net

ScanTool.net offers a complete line of PC-based scan tools for under £50.

- 1 year unconditional warranty
- 90 day money back guarantee
- For use with EOBD compliant vehicles
- Fast shipping
- Compatible with a wide range of diagnostic software

Visit our website, or email us at:
sales@scantool.net

www.elektor.com

USB INSTRUMENTS

http://www.usb-instruments.com

USB Instruments specialises in PC based instrumentation products and software such as Oscilloscopes, Data Loggers, Logic Analaysers which interface to your PC via USB.

VIRTINS TECHNOLOGY

www.virtins.com

PC and Pocket PC based virtual instrument such as sound card real time oscilloscope, spectrum analyzer, signal generator, multimeter, sound meter, distortion analyzer, LCR meter. Free to download and try.

CANDO - CAN BUS ANALYSER

http://www.cananalyser.co.uk

- USB to CAN bus interface
- USB powered
- FREE CAN bus analyser S/W
- Receive, transmit & log. CAN messages
- ISO11898 & CAN 2.0a/2.0b compliant
- Rugged IP67 version available

SHOWCASE YOUR COMPANY HERE

Elektor Electronics has a feature to help customers promote their business, Showcase - a permanent feature of the magazine where you will be able to showcase your products and services.

- For just £242 + VAT (£22 per issue for eleven issues) Elektor will publish your company name, website address and a 30-word description
- For £363 + VAT for the year (£33 per issue for eleven issues) we will publish the above plus run a 3cm deep full colour image - e.g. a product shot, a screen shot from your site, a company logo - your choice

Places are limited and spaces will go on a strictly first come, first served basis. So-please fax back your order today!

I wish to promote my company, please book my space:
• Text insertion only for £242 + VAT • Text and photo for £363 + VAT

NAME: .... ORGANISATION: ....
JOB TITLE: ....
ADDRESS: ....
....
.... TEL: ....
PLEASE COMPLETE COUPON BELOW AND FAX BACK TO 00-44-(0)1932 564998
COMPANY NAME ....
WEB ADDRESS ....
30-WORD DESCRIPTION ....
....
....
....
....

Going Strong

A world of electronics from a single shop!

CD/DVD-ROMs

Completely updated

Elektor's Components Database 5

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. Each databank contains the following on (almost) any component: enclosure drawing, pin connections, technical data (as far as known). Also included is a search engine acting on user supplied parameters. The ECD gives you easy access to design data for over 5,400 ICs, more than 35,800 transistors, FETs, thyristors and triacs, just under 25,000 diodes and 1,800 optocouplers. All databank applications are fully interactive, allowing the user to add, edit and complete component data. This CD-ROM is a must-have for all electronics enthusiasts!

ISBN 978-90-5381-159-7 • £24.90 • US $39.50

Modern technology for everyone

FPGA Course

FPGAs have established a firm position in the modern electronics designer's toolkit. Until recently, these 'super components' were practically reserved for specialists in high-tech companies. The nine lessons on this courseware CD-ROM are a step by step guide to the world of Field Programmable Gate Array technology. Subjects covered include not just digital logic and bus systems but also building an FPGA webserver, a 4-channel multimeter and a USB controller. The CD also contains PCB layout files in pdf format, a Quartus manual, project software and various supplementary instructions.

ISBN 978-90-5381-225-9 • £14.50 • US $29.00

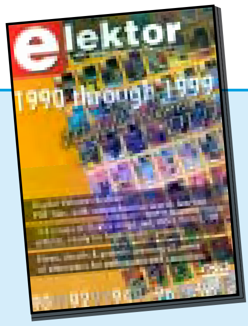
Embedded USB Know How

USB Toolbox

This CD-ROM contains technical data about the USB interface. It also includes a large collection of data sheets for specific USB components from a wide range of manufacturers. There are two ways to incorporate a USB interface in a microcontroller circuit: add a USB controller to an existing circuit, or use a microcontroller with an integrated USB interface. Both options are available on this CD-ROM. Included on this CD-ROM areUSB Basic Facts, several useful design tools for hardware and software, and all Elektor articles on the subject of USB.

ISBN 978-90-5381-212-9 • £19.90 • US $39.00

Prices and item descriptions subject to change. E. & O.E



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. The DVD also contains (free of charge) the entire 'The Elektor Datasheet Collection 1…5' CD-ROM series, with the original full datasheets of semiconductors, memory ICs, microcontrollers, and much more.

ISBN 978-0-905705-76-7 • £69.00 • US $109.00

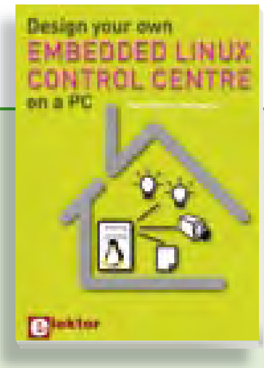
Bestseller!

All articles published in 2008

DVD Elektor 2008

This DVD-ROM contains all editorial articles published in Volume 2008 of the English, Spanish, Dutch, French and German editions of Elektor magazine. Using Adobe Reader, articles are presented in the same layout as originally found in the magazine. The DVD is packed with features including a powerful search engine and the possibility to edit PCB layouts with a graphics program, or printing hard copy at printer resolution.

ISBN 978-90-5381-235-8 • £17.50 • US $35.00



A DIY system made from recycled components

Design your own Embedded Linux control centre on a PC

This book covers a do-it-your-self system made from recycled components. The main system described in this book reuses an old PC, a wireless mains outlet with three switches and one controller, and a USB webcam. All this is linked together by Linux. This book will serve up the basics of setting up a Linux environment – including a software development environment – so it can be used as a control centre. The book will also guide you through the necessary setup and configuration of a webserver, which will be the interface to your very own home control centre. All software needed will be available for downloading from the Elektor website.

234 pages • ISBN 978-0-905705-72-9
£24.00 • US $42.00

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: sales@elektor.com

elektor SHOP

New!

Bring your microcontroller to life

Artificial Intelligence

This book contains 23 special and exciting artificial intelligence machine-learning projects, for microcontroller and PC. Learn how to set up a neural network in a microcontroller, and how to make the network self-learning. Or discover how you can breed robots, and how changing a fitness function results in a totally different behavior. Several artificial intelligence techniques are discussed: expert system, neural network, subsumption, emerging behavior, genetic algorithm, cellular automata. roulette brains etc.

256 pages • ISBN 978-0-905705-77-4
£32.00 • US $46.00

45 projects for PIC, AVR and ARM

Microcontroller Systems Engineering

This book covers 45 exciting and fun Flowcode projects for PIC, AVR and ARM microcontrollers. Each project has a clear description of both hardware and software with pictures and diagrams, which explain not just how things are done but also why. As you go along the projects increase in difficulty and the new concepts are explained. You can use it as a projects book, and build the projects for your own use. Or you can use it as a study guide.

329 pages • ISBN 978-0-905705-75-0
£29.00 • US $52.00

Books

Bestseller!

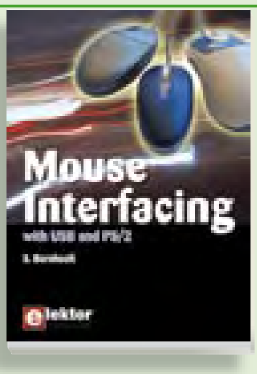
Learn by doing

C Programming

for Embedded Microcontrollers

If you would like to learn the C Programming language to program microcontrollers, then this book is for you. No programming experience is necessary! You'll start learning to program from the very first chapter with simple programs and slowly build from there. Initially, you program on the PC only, so no need for dedicated hardware. This book uses only free or open source software and sample programs and exercises can be downloaded from the Internet.

324 pages • ISBN 978-0-905705-80-4
£32.50 • US $52.00



Connect your mouse into new embedded applications

Mouse Interfacing

This book describes in-depth how to connect the mouse into new embedded applications. It details the two main interface methods, PS/2 and USB, and offers applications guidance with hardware and software examples plus tips on interfacing the mouse to typical microcontrollers. A wide range of topics is explored, including USB descriptors, a four-channel, millivolt-precision voltage reference all with fully documented source-code.

256 pages • ISBN 978-0-905705-74-3
£26.50 • US $53.00

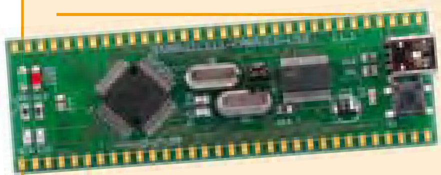
Automotive CAN controller

(April 2009)

Since cars contain an ever increasing amount of electronics, students learning about motor vehicle technology also need to know more about electronics and microcontrollers. In collaboration with the Timloto o.s. Foundation in the Netherlands, Elektor designed a special controller PCB, which will be used in schools in several countries for teaching students about automotive technologies. But it can also be used for other applications, of course. The heart of this board is an Atmel AT90CAN32 with a fast RISC core.

Kit of parts, incl. PCB with SMDs prefitted

Art-Nr. 080671-91 • £52.00 • US $79.00



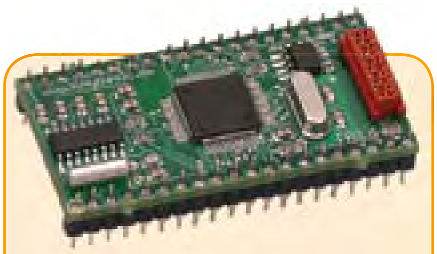
The 32-bit Machine

(April 2009)

With this attractively priced starter kit you get everything you need for your first hands-on experiments with the new R32C/111 32-bit microcontroller. The power supply is drawn from your computer via the USB connection, which simplifies things rather nicely. The starter kit consists of an R32C carrier board (a microcontroller module equipped with the R32C/111 chip) and a software CD-ROM containing the necessary development tools. As with the earlier R8C/13 'Tom Thumb' project in Elektor Electronics (November 2005 through March 2006), the R32C carrier board is an in-house-development of Glyn, an authorised distributor for Renesas in Germany.

R32C/111 Starterkit (32-bit-Controller-board & CD-ROM)

Art-Nr. 080928-91 • £27.00 • US $42.50



M16C TinyBrick

(March 2009)

A TinyBrick is a small self-contained microcontroller module fitted with a powerful Renesas 16-bit M16C microcontroller. A BASIC interpreter is installed in the module to simplify software development. Beginners will find it an ideal starting out point while more experienced users will appreciate its power and convenience. With this evaluation board (together with a TinyBrick) you can build an intruder alarm that sends SMS texts.

Kit of parts incl. TinyBrick-PCB with SMD parts and microntroller premounted plus all other parts

Art-Nr. 080719-91 • £54.00 • US $87.50

Bestseller!

LED Top with Special Effects

(December 2008)

If you fit a line of LEDs on a circular PCB and power them on continuously, they generate rings of light when the board is spun. If you add a microcontroller, you can use the same set of LEDs to obtain a more interesting effect by generating a 'virtual' text display. The article also describes a simple technique for using the Earth's magnetic field to generate a synchronisation pulse. The potential applications extend from rotation counters to an electronic compass.

Kit of parts incl. SMD-stuffed PCB and programmed controller

Art-Nr. 080678-71 • £42.00 • US $59.00

Kits & Modules

Prices and item descriptions subject to change. E. & O.E

Product Shortlist

May 2009 (No. 389)	£	US $
Experimenting with the MSP430		
080558-91 .... PCB, populated and tested	35.00	55.00
080558-92 .... TI eZ430-F2013 Evaluation Kit	see www.elektor.com	
RGB LED Driver		
080178-41 .... Programmed controller	8.90	13.75

April 2009 (No. 388)	£	US $
The 32-bit Machine		
080928-91 .... R32C/111 Starterkit (32-bit-Controllerboard & CD-ROM)	27.00	42.50
Automotive CAN Controller		
080671-91 .... Kit of parts, incl. PCB with SMDs prefitted	52.00	79.00
Automatic Running-in Bench		
080253-71 .... Kit of parts incl. PCB-1 with SMDs prefitted	185.00	270.00
090146-91 .... ARMee plug-in board mk. II	50.00	74.00

March 2009 (No. 387)	£	US $
M16C TinyBrick		
080719-91 .... Kit of parts: TinyBrick-PCB with SMD parts and microntroller premounted; plus all other parts	54.00	87.50

February 2009 (No. 386)	£	US $
Model Coach Lighting Decoder		
080689-1 ...... PCB, long (l = 230 mm)	7.30	10.95
080689-2 ...... PCB, medium (l = 190mm)	7.30	10.95
080689-3 ...... PCB, short (l = 110mm)	5.80	8.95
080689-41 .... PIC12F683, programmed	6.20	9.50
Transistor Curve Tracer		
080068-1 ...... Main PCB	26.50	42.00
080068-91 .... PCB, populated and tested	55.00	82.50

January 2009 (No. 385)	£	US $
Radio for Microcontrollers		
071125-71 .... 868 MHz module	7.20	9.95
ATM18 on the Air		
071125-71 .... 868 MHz module	7.20	9.95
Meeting Cost Timer		
080396-41 .... ATmega168, programmed	8.50	12.50
Capacitive Sensing and the Water Cooler		
080875-91 .... Touch Sensing Buttons Evaluation kit	27.50	39.95
080875-92 .... Touch Sensing Slider Evaluation kit	27.50	39.95
Three-Dimensional Light Source		
080355-1 ...... Printed circuit board	24.90	39.90
Moving up to 32 Bit		
080632-91 .... ECRM40 module	32.00	46.50

December 2008 (No. 384)	£	US $
PLDM		
071129-1 ...... Printed circuit board	5.80	9.50
Hi-fi Wireless Headset		
080647-1 ...... Printed circuit board: Transmitter	7.90	15.80
080647-2 ...... Printed circuit board: Receiver	7.90	15.80
LED Top with Special Effects		
080678-71 .... Kit of parts incl. SMD-stuffed PCB and programmed controller	42.00	59.00

November 2008 (No. 383)	£	US $
Motorised Volume Pot		
071135-41 .... Programmed controller ATMEGA8-16PU	5.90	11.80
Speed Camera Warning Device		
080615-1 ...... Printed circuit board	15.50	31.00
080615-41 .... Programmed controller PIC16F876A-I/SO	11.80	23.60
Remote Control by Mobile Phone		
080324-1 ...... Printed circuit board	17.80	35.60
080324-41 .... Programmed controller ATMEGA8-16PU	5.90	11.80
080324-71 .... Kit of parts	54.00	99.00

Bestsellers

Books

1. **C Programming** for Embedded Microcontrollers
   ISBN 978-0-905705-80-4 .... £32.50 ..... US $52.00
2. **Microcontroller Systems Engineering**
   ISBN 978-0-905705-75-0 .... £29.00 ..... US $52.00
3. **Mouse Interfacing**
   ISBN 978-0-905705-74-3 .... £26.50 ..... US $53.00
4. **Artificial Intelligence**
   ISBN 978-0-905705-77-4 .... £32.00 ..... US $46.00
5. **PIC Microcontrollers**
   ISBN 978-0-905705-70-5 .... £27.95 ..... US $52.00

CD/DVD-ROMs

1. **DVD Elektor 2008**
   ISBN 978-90-5381-235-8 .... £17.50 ..... US $35.00
2. **Elektor 1990 through 1999**
   ISBN 978-0-905705-76-7 .... £69.00 ... US $109.00
3. **FPGA Course**
   ISBN 978-90-5381-225-9 .... £14.50 ..... US $29.00
4. **USB Toolbox**
   ISBN 978-90-5381-212-9 .... £19.90 ..... US $39.00
5. **Ethernet Toolbox**
   ISBN 978-90-5381-214-3 .... £19.50 ..... US $39.00

Kits & Modules

1. **LED Top with Special Effects**
   Art. # 080678-71 .... £42.00 .... US $59.00
2. **The 32-bit Machine**
   Art. # 080928-91 .... £27.00 ..... US $42.50
3. **Transistor Curve Tracer**
   Art. # 080068-91 .... £55.00 ..... US $82.50
4. **Evaluation Kit CapSense Buttons**
   Art. # 080875-91 .... £27.50 ..... US $39.95
5. **M16C TinyBrick**
   Art. # 080719-91 .... £54.00 .... US $87.50

Order quickly and securely through www.elektor.com/shop or use the Order Form near the end of the magazine!

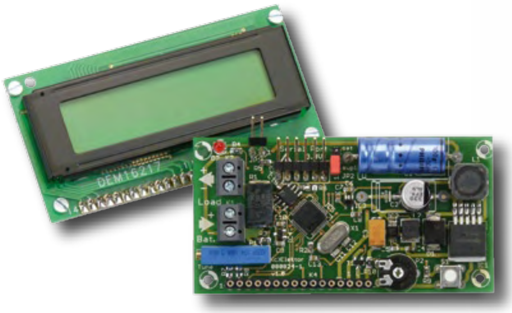
elektor SHOP

Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH • United Kingdom
Tel. +44 20 8261 4509
Fax +44 20 8261 4447
Email: sales@elektor.com

NEXT MONTH IN ELEKTOR

Campground Current Regulator

Mains receptacles in campgrounds can usually supply only a limited current. If you draw too much current, a circuit breaker trips, and it has to be reset by the campground manager. You have to pay for this (a sort of fine). To avoid this problem, the Elektor labs have developed a special current regulator that ensures that the amount of current actually drawn is limited to a present value. The regulator circuit is built around an Atmel U2008B. This 8-pin phase-control regulator needs only a few external components, and it senses the load current internally.



Battery Monitor

This device was originally developed to monitor the charge state of batteries used with a solar panel, but it can also be used with other types of rechargeable batteries. The circuit measures the charge and discharge current, nominal voltage, momentary capacity, and the energy supplied or accumulated by a battery. The battery monitor is built around an LPC2103 microcontroller, which can be programmed via a serial interface. A 22-bit A/D converter provides very accurate current and voltage measurements. The measurement data is shown on a two-line LCD.

Portable Solar Panels

If you like to spend a few days out of doors with a rucksack or a bike, you often discover that the batteries of your electronic travelling companions such as your mobile phone or GPS receiver, or even your pocket light, are empty just when you need them. And of course, there's no mains receptacle in sight. Fortunately, portable solar panels are available nowadays in various sorts and sizes, and you can use them to recharge a couple of batteries or your mobile phone. In next month's issue, we examine several of these modules and test their effectiveness.

Note: we regret that "True RMS Voltmeter with Frequency meter and "Mini PWM Audio Amplifier" could not be accommodated in the April 2009 issue as planned.

Article titles and magazine contents subject to change, please check 'Magazine' on www.elektor.com

The June 2009 issue comes on sale on Thursday 21 May 2009 (UK distribution only). UK mainland subscribers will receive the issue between 16 and 19 May 2009.

w.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.

Elektor on the web

All magazine articles back to volume 2000 are available online in pdf format. The article summary and parts list (if applicable) can be instantly viewed to help you positively identify an article. Article related items are also shown, including software downloads, circuit boards, programmed ICs and corrections and updates if applicable. Complete magazine issues may also be downloaded.

In the Elektor Shop you'll find all other products sold by the publishers, like CD-ROMs, kits 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
- Surveys and polls
- FAQ, Author Guidelines and Contact

Order Form 05-2009

Description	Price each	Qty.	Total	Order Code
DVD i-TRIXX Freeware Collection 2009 NEW	£27.50			
CD-ROM ECD 5 NEW	£24.90			
C Programming for Embedded Microcontrollers NEW	£32.50			
Artificial Intelligence NEW	£32.00			
Mouse Interfacing	£26.50			
Microcontroller Systems Engineering	£29.00			
DVD Elektor 2008	£17.50			
Free Elektor Catalogue 2009				
	Sub-total			
	P&P			
	Total paid			

Prices and item descriptions subject to change.
The publishers reserve the right to change prices without prior notification. Prices and item descriptions shown here supersede those in previous issues. E. & O.E.

METHOD OF PAYMENT
(see reverse before ticking as appropriate)

- ☐ **Bank transfer**
- ☐ **Cheque** (UK-resident customers ONLY)
- ☐ **Giro transfer**
- ☐ VISA ☐ Eurocard/MasterCard

Expiry date: ______________

Verification code: ______________

Please send this order form to*
(see reverse for conditions)

Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom

Tel.: +44 20 8261 4509
Fax: +44 20 8261 4447
www.elektor.com
sales@elektor.com

*USA and Canada residents should use $ prices, and send the order form to:
Elektor US
PO Box 876
Peterborough NH 03458-0876
Phone: 603-924-9464
Fax: 603-924-9467
E-mail: custservus@elektor.com

Name ______________

Address + Post code ______________

Tel. ______________ *Email* ______________

Date – – *Signature* ______________

EL05

Subscription Order Form 05-2009

Yes, I am taking out an annual subscription to Elektor and receive a free 2GB MP3 player*.

I would like:

- ☐ **Standard Subscription (11 issues)**
- ☐ **Subscription-Plus (11 issues plus the Elektor Volume 2009 CD-ROM)**

* *Offer available to Subscribers who have not held a subscription to Elektor during the last 12 months. Offer subject to availability. See reverse for rates and conditions.*

Name ______________

Address + Post code ______________

Tel. ______________ *Email* ______________

Date – – *Signature* ______________

EL05

METHOD OF PAYMENT
(see reverse before ticking as appropriate)

- ☐ **Bank transfer**
- ☐ **Cheque** (UK-resident customers ONLY)
- ☐ **Giro transfer**
- ☐ VISA ☐ Eurocard/MasterCard

Expiry date: ______________

Verification code: ______________

Please send this order form to

Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom

Tel.: +44 20 8261 4509
Fax: +44 20 8261 4447
www.elektor.com
subscriptions@elektor.com

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. 40209520 held by Elektor Electronics with ABN-AMRO Bank, London. IBAN: GB35 ABNA 4050 3040 2095 20. 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.

Giro transfer into account no. 34-152-3801, held by Elektor Electronics. Please do not send giro transfer/deposit forms directly to us, but instead use the National Giro postage paid envelope and send it to your National Giro Centre.

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.

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 2009

SUBSCRIPTION RATES FOR ANNUAL SUBSCRIPTION

	Standard	Plus
United Kingdom	**£44.00**	**£53.00**
Surface Mail Rest of the World	**£58.00**	**£67.00**
Airmail Rest of the World	**£74.00**	**£83.00**
USA	**£59.95**	See www.elektor-usa.com for special offers
Canada	**£70.95**	

HOW TO PAY

Bank transfer into account no. 40209520 held by Elektor Electronics. with ABN-AMRO Bank, London. IBAN: GB35 ABNA 4050 3040 2095 20. 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.

Giro transfer into account no. 34-152-3801, held by Elektor Electronics. Please do not send giro transfer/deposit forms directly to us, but instead use the National Giro postage paid envelope and send it to your National Giro Centre.

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 £35.00, a Student Subscription-Plus costs £44.20 (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. Subsciptions cannot be cancelled after they have run for six months or more.

January 2009

Index of Advertisers

Advertising space for the issue of 25 June 2009
may be reserved not later than 26 May 2009
with Huson International Media– Cambridge House- Gogmore Lane- Chertsey, Surrey KT 16 9AP– England – Telephone 01932 564999 – Fax 01932 564998 – e-mail: p.brady@husonmedia.com to whom all correspondence, copy instructions and artwork should be addressed.


LEAP INTO THE FUTURE...