SoCs, PSoCs ...
Design your own chip

March 2011

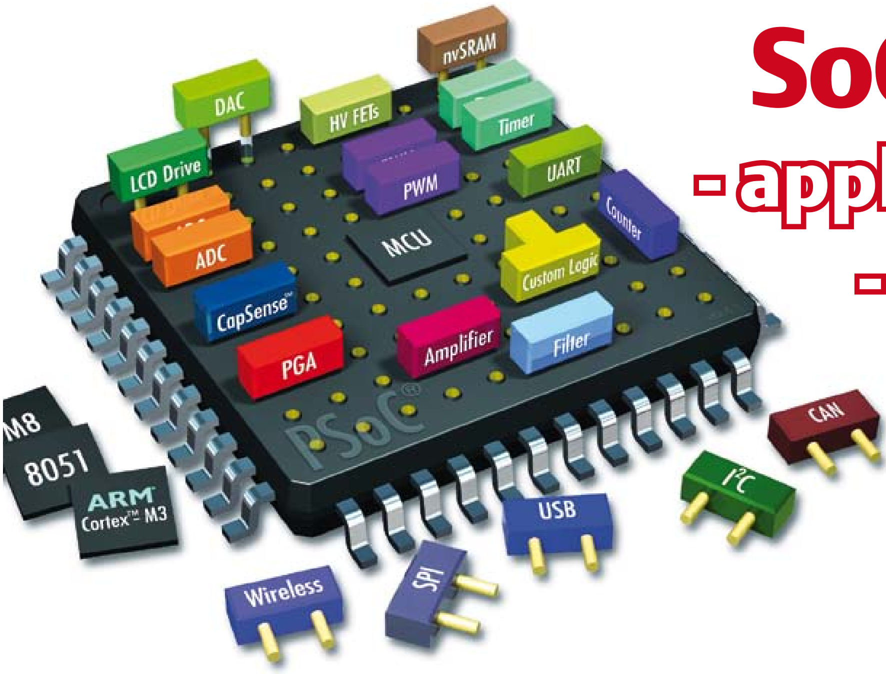
AUS$ 14.50 - NZ$ 17.50 - SAR 102.95 £ 4.80

+ Develop your own MP3 player

elektor

www.elektor.com

SatFinder
TV dish alignment using GPS



SoC, PSoC & Co.
- application examples
- DIY chip building
- big names
- evaluation kits
- getting started

+ Mini Webserver using Bascom-AVR & Minimod18
Handles shopping lists by Internet – and more.

+ A string of 160 RGB LEDs
A colourful display with the ATM18

R12
03
9 770268 451166

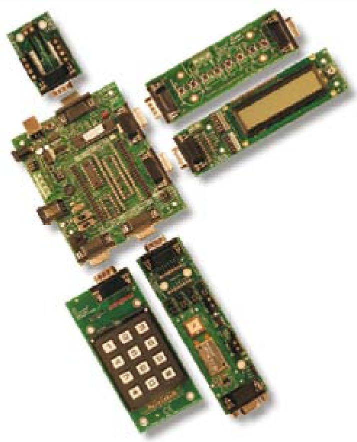
Developing and learni

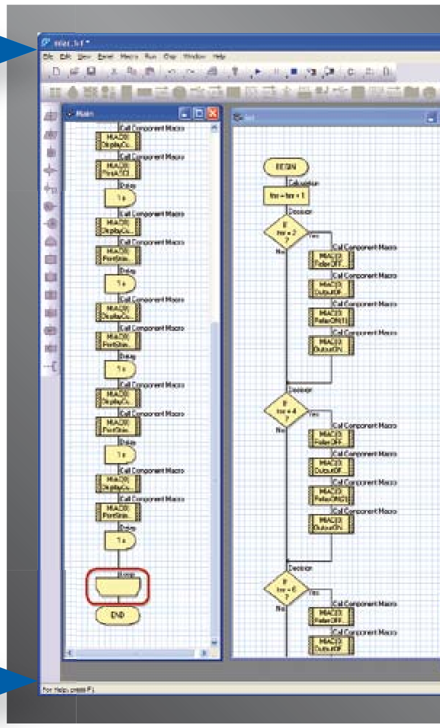
FLOWCODE 4
NO CODING, NO LIMITS...

Flowcode 4 is one of the world's most advanced graphical programming languages for microcontrollers (PIC, AVR, ARM and, brandnew, dsPIC/PIC24). The great advantage of Flowcode is that it allows those with little to no programming experience to create complex electronic systems in minutes.

www.elektor.com/flowcode

... for electronics





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

... for industrial control

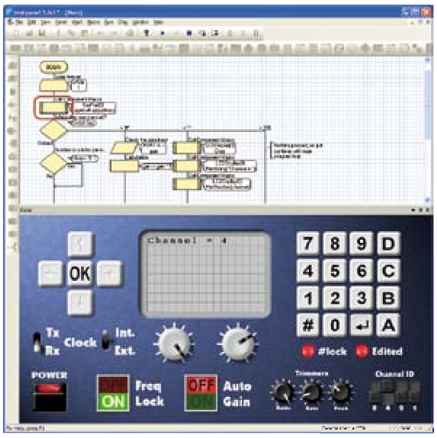
MIAC (**M**atrix **I**ndustrial **A**utomotive **C**ontroller) is an industrial grade control unit which can be used to control a wide range of different electronic systems including sensing, monitoring and automotive. Internally the MIAC is powered by a powerful 18 series PICmicro device which connects directly to the USB port and can be programmed with Flowcode, C or assembly. Flowcode 4 is supplied with the unit. MIAC is supplied with an industrial standard CAN bus interface which allows MIACs to be networked together.

Flowkit

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

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

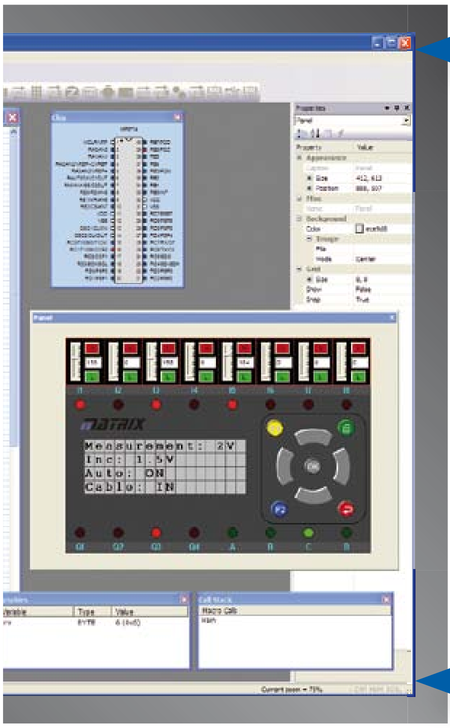
ng with Flowcode 4



New features in Flowcode 4

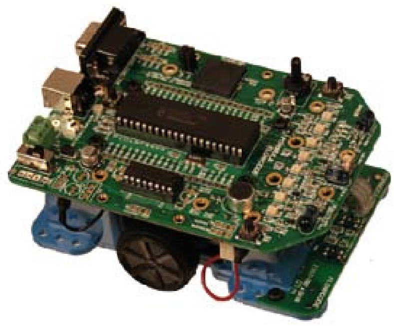
Flowcode 4 is packed with new features that make development easier including:

- **Panel Creator**
- **In Circuit Debug**
- **Virtual networks**
- **C Code customization**
- **Switch Icon**
- **Floating point**
- **Additional string functions**
- **Watchdog timer support**
- **New user interface**
- **New components**
- **Fast USB development**



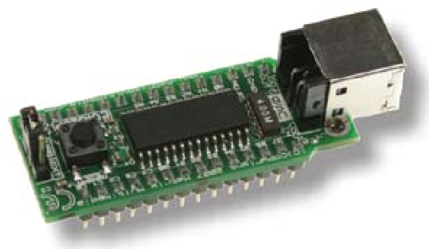
… for robotics

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



… for USB projects

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



More information and products at:

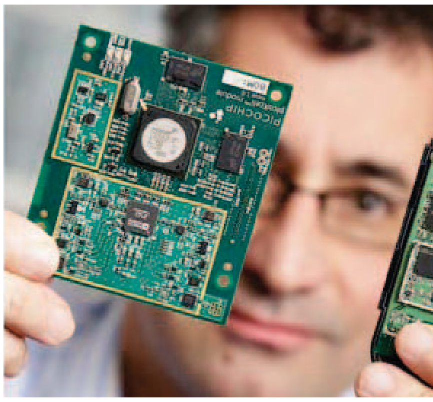
www.elektor.com/eblocks

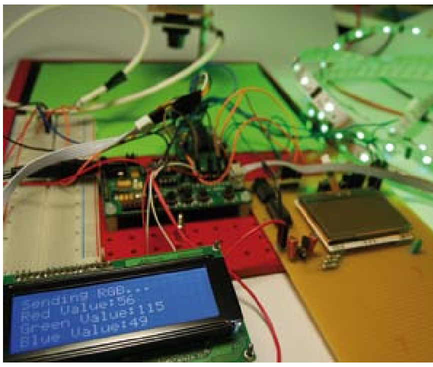
Not a whole new concept, just wider

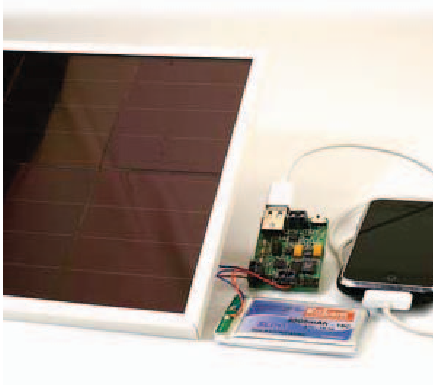
Here at Elektor the rare appearance of SoC (system-on-a-chip) devices in projects proposed by readers does not match the number of jokes and puns about the acronym, like “put a SoC in it” in reply to the PR dept of XYZ Corp. bombarding us with emails and “knock your SoCs off” quoting those taking a dim view of the speeds achievable with SoCs. That’s curious, because “life SoCs”, I mean SoC (and touchscreen) technology is rocketing in consumer equipment like PDA and mobile phones and you’d expect Elektor readers to join the fun. Some encouragement may be in order. The success of PSoC in particular seems to underline that microcontrollers are great devices — flexible, powerful, totally programmable and all that — but not a wonder solution to all design challenges like adding intelligent analogue I/O in a way that suits the 21st century electronics design engineer. Drag & Drop, libraries, graphics, icons and compile-till-you-drop are now the standard where once you had to fear mundane messages like [Error 193b: irretrievable] on the command line and telling the boss the project was going to be delayed over the weekend, linked to shop-till-you-drop to locate the right peripheral device for the application.

It’s not all SoCs this month. Besides spurring you on to exploit the technology and getting to grips with them possibly through an inexpensive evaluation kit, this edition of Elektor also presents two projects for the travel-minded among you. Using positive/forward thinking, March sort of equates to Spring, although still far off in time as I write this. With SatFinder (page 24) close to your mobile satellite TV dish you have an instant view of the two angles you need to set to pick up your favourite DBTV programme from the bird 36,000 kms up in the sky, the Solar Charger described on page 68 helping to keep your equipment powered in the happy absence of an AC power outlet. Don’t attempt to build the TV set on page 76 though, I’ve reliable information it won’t work but I’d love to hear otherwise. It’s definitely better to spend time on Hexadoku ‘Digest’ this month as it may win you a nice PSoC 5 kit.

Enjoy reading this edition,
Jan Buiting, Editor

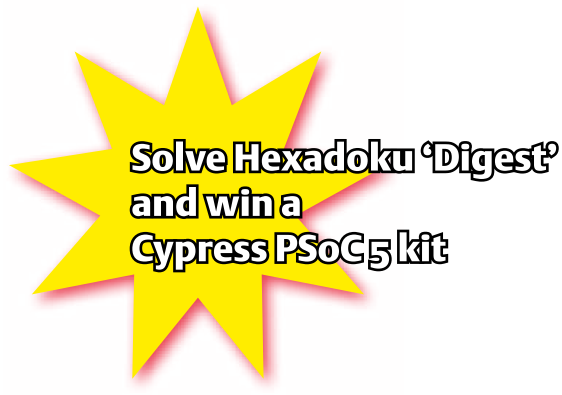






CONTENTS

Volume 37
March 2011
no. 411



page $4B_{hex}$

elektor | international media bv

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 37, Number 411, March 2011 ISSN 1757-0875

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

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

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

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

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

Editor: Jan Buiting (editor@elektor.com)

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

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

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:
Elektor International Media b.v.
P.O. Box 11 NL-6114-ZG Susteren The Netherlands
Telephone: (+31) 46 4389444, Fax: (+31) 46 4370161
Email: t.vanhoesel@elektor.com
Internet: www.elektor.com
Advertising rates and terms available on request.

Saelig: universal USB controller with 50 I/O pins

Saelig Company, Inc. has introduced IO-Warrior56 — a universal USB Controller that allows easy access to input or output functions via a USB bus. Featuring 50 generic I/O lines, IO-Warrior56 is also an I2C/SPI master, allowing interface with a wide range of available ICs.
IO-Warrior56 offers a simple access to the USB since it has been designed as a generic HID device — the protocol is all in the IO-Warrior56 chip. Only a few simple lines of code are needed to access the I/O pins. If you need to connect simple devices to a computer, like relays, switches, keypad, or a small display, IO-Warrior56 is a simple solution. Working with USB used to mean that you had to develop specific code for a USB-enabled microcontroller, developing a unique driver with lots of documentation, using expensive development systems. Now driving LCDs, LEDs, or keypads from USB is easy! IO-Warrior56 also supports a range of industrial standard interfaces such as I2C and SPI to simplify interfacing to chips, modules, or displays.

Features:

- Full Speed USB2.0-compliant interface (12MBit/sec)
- 50 general purpose I/O Pins, typ. 1000Hz rate (input or output)
- I^2C master function, 50, 100, or 400kHz SPI master interface, up to 12 MBit/sec, throughput up to 62 Kbytes/sec
- Controls various display modules, including most graphic modules · Drives up to 8x64 LED matrix Drives 8x8 switch or button matrix
- Software support for Mac(10.2 and up), Linux (Kernel 2.6), and Windows (2K/XP/Vista/7)
- No USB knowledge needed
- Single +5 V power supply (50 mA operating, 25 µA suspend)
- 0.1"-spaced 56-pin module
- Extended temperature range: –40°C to +85°C.

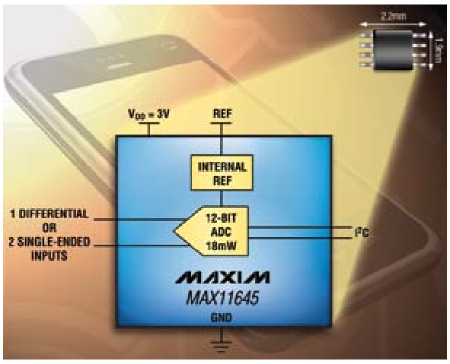
Made in Germany by Code Mercenaries, the USB I/O specialists, IO-Warrior56 is available now from Saelig, with prices starting at $30 USD (qty 100). A Development Starter Kit is also available at $99.00.

www.saelig.com (100926-III)

Industry's smallest 2-channel ADC saves space and power

Maxim Integrated Products introduces the MAX11645, the industry's smallest 2-channel ADC. This 12-bit I2C ADC includes an internal reference in a tiny 1.9 mm x 2.2 mm wafer-level package (WLP). A 4 x 3 bump array with 0.5 mm pitch facilitates layout on four-layer PCBs, while a slim 0.64 mm height makes it perfect for designs laying components out on both sides of the PCB. This ADC minimizes both external component count and total solution size, only requiring a 0.1 microfarad ceramic bypass capacitor on the supply voltage input. Combining a compact footprint with best-in-class power dissipation (18 microwatts at 1 ms conversion times), the MAX11645 is well suited for applications ranging from energy-harvesting sensors, to portable consumer electronics, to point-of-load monitoring (voltage, current, and temperature) in networking and computer systems.
The MAX11645 is capable of sample rates up to 94 ksps — 28x faster than the nearest competitor in a comparable solution size. By taking advantage of the ADC's high sample rate, multiple channels can be converted in a short period of time. This capability allows the device to spend more time in shutdown mode, further reducing total system power consumption.

The MAX11645 operates from a single 2.7 V to 3.6 V supply and includes a 2.048 V internal reference. This 12-bit ADC offers up to 70 dB SINAD, ±1 LSB (max.) INL and DNL error, plus 0.3 ppm/degrees Celsius gain-error temperature drift. Input signals in the 0 to Vref (unipolar) or ±Vref/2 (bipolar) range can be resolved with true 12-bit accuracy.
The MAX11645 is the latest release in Maxim's extensive I2C ADC family. Offering pin-



and software-compatible 2-/4-/8-/12-channel, 8-/10-/12-bit ADCs enables designers to easily trade off space, performance, and cost on platform designs. Uniquely, these ADCs also include internal references, FIFOs, and selectable scan modes. Refer to the selection table for the complete listing of MAX116xx I2C ADCs.
For applications that require less than 12-bit resolution, the MAX11647 is a pin-compatible 10-bit version of the MAX11645. The MAX11645/MAX11647 are offered in 12-bump WLP and 8-pin microMAX(R) packages, and are guaranteed over the –40 degrees Celsius to +85 degrees Celsius extended temperature range. Production quantities are available now. Evaluation kits are available for the 4-, 8-, and 12-channel versions (MAX11613/MAX11615/MAX11617) of these ADCs.

www.maxim-ic.com/MAX11645 (100926-V)

LED solution enables drop-in replacements for halogen MR16 lamps

Maxim Integrated Products introduces the MAX16840, an LED driver that employs a proprietary architecture to ensure flicker-free, dimmable operation with electronic transformers and cut-angle dimmers. Maxim's patent-pending approach enables the design of retrofit LED lamps that can replace halogen MR16s without any changes to the existing electrical infrastructure. This removes an important obstacle to commercial viability, allowing end users to enjoy all the benefits of LED lighting with substantially lower deployment costs.
To facilitate adoption, LED retrofit lamps must be compatible with electronic transformers and cut-angle dimmers. These transformers and dimmers are designed

Kinetic motion wind-up charger boost battery life for Android phones

element14, a collaborative social community and electronics store for design engineers and electronics enthusiasts, and modding guru Benjamin J. Heckendorn, a.k.a. Ben Heck, present an action-packed episode of "The Ben Heck Show" for the modding — and DIY — inclined, featuring a kinetic motion-inspired wind-up charger for Android phones and a Rapid Pinball Prototyping System (RPPS) for Ben's latest pinball creation.

"Have you ever noticed that there never seems to be a charger around when you need one – I sure have," said Ben Heck. "In this latest episode, I look at kinetic motion to create a portable cell phone battery charger in the event you're ever lost on a deserted island and need to call for help to get rescued."

Ben uses two simple, affordable items — a wind-up LED flashlight and micro USB adapter – and morphs them into a portable, wind-up battery charger for Android phones. Later it's back to pinball wars as Ben constructs a custom RPPS gaming cabinet with fellow modding friends that allows users to rapidly fire new shots. Ben rounds out the episode with a complete overhaul of an old, non-working LCD screen and turns it into a perfectly functioning monitor.

"Ben's latest project is a great example of how a little mod-like thinking, some DIY energy, and affordable electronics components can create useful, everyday gadgets," said Alisha Mowbray, senior vice president of marketing, element14. "The wind-up charger gives Android users an easy and affordable way to ensure they always have a ready-to-use cell phone, whether they're on the road without a car charger, on a long camping weekend, or stuck at the airport."

Show fans can enter for a chance to win the kinetic wind-up charger featured on the episode as well as submit project suggestions for future episodes.

www.element14.com (110050-VII)

for the perfectly resistive loads of traditional halogen lamps. LED drivers, however, are very nonlinear and not purely resistive loads. As a consequence, LED lamps flicker, do not dim, and in some cases do not turn on at all when used with the existing electrical infrastructure.

The MAX16840 solves this problem by using a unique, patent-pending approach to control the input current of the lamp. By actively shaping the input current it ensures flicker-free operation with most electronic transformers and dimmers. This enables LED lamp designers to create drop-in replacements for halogen MR16s, thus eliminating the costly infrastructural upgrades required by competitive solutions.

In addition, the MAX16840 can be designed in without electrolytic capacitors. This extends the lifetime of the LED lamp, since electrolytic capacitors are usually the first component that fails in the driver circuit. Operation without electrolytic capacitors reduces the cost and size of the driver, allowing it to fit in the small MR16 form factor. An integrated switching MOSFET further extends these space savings while reducing component count and cost.

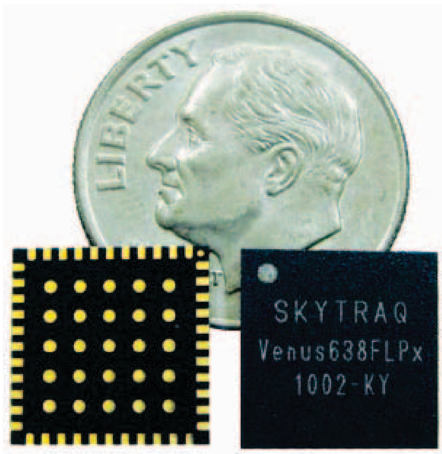
The MAX16840 can deliver up to 20 W of power. It is fully specified over the –40 degrees Celsius to +125 degrees Celsius temperature range and is available in thermally enhanced, 3mm x 3mm, 10-pin TDFN package.

www.maxim-ic.com/Flicker-Free-LEDs

www.maxim-ic.com/MAX16840 (110050-I)

High Performance 20 Hz –165dBm GPS receiver

SkyTraq's new Venus638FLPx features industry leading 20 Hz update rate, –165dBm signal tracking and –148dBm cold starting sensitivity, 29 second cold start TTFF, 67 mW full-power navigation, and in 10x10x1.3 mm LGA69 packaging.



The device contains all the necessary components of a GPS receiver, including GPS RF and baseband, SAW filter, LNA, 0.5 ppm TCXO, RTC crystal, LDO regulator, and passive components. A complete GPS receiver requires only antenna and power supply to work. Its exceptionally low cascaded RF section noise figure of 1.2 dB allows the Venus638FLPx to work directly with a passive antenna.

Best-in-class TTFF, low power, high sensitivity performance and miniature size make Venus638FLPx an ideal choice for embedding location awareness into portable applications. Dedicated signal parameter search engine capable of performing 8 million time-frequency hypothesis testing per second offers fastest signal acquisition performance in the industry. Advanced track engine along with multipath and jamming detection mitigation algorithms allows continuous accurate navigation in harsh environments such as the "urban canyon" and under heavy foliage.

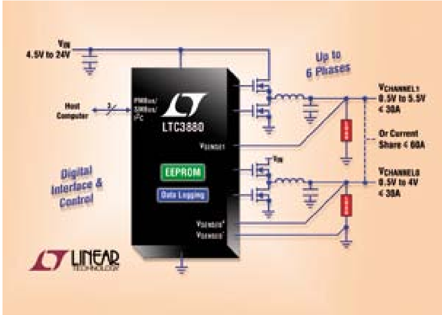
The Venus638FLPx has 8 Mbit internal Flash, 2 UARTs, 2 SPIs, I²C, and 19 GPIO pins. The flash-based design makes it capable of supporting customized firmware; 50% usage of flash memory by the GPS kernel software enables possibility of simple user application development on the remaining 512 kByte program space. The Venus638FLPx has an operating temperature range of –40 to approximately +85 deg C.

www.skytraq.com.tw (110050-IV)

Dual Output Synchronous Step-Down DC/DC Controller

The LTC3880/-1 from Linear Technology Corporation allows for digital programming and read back for real-time control and monitoring of critical point-of-load converter functions. Programmable control parameters include output voltage, margining and current limits, input and output supervisory limits, power-up sequencing and tracking, switching frequency and identification and traceability data. On-chip precision data converters and EEPROM allow for the capture and nonvolatile storage of regulator configuration settings and telemetry variables, including input and output voltages and currents, duty cycle, temperature and fault logging.

The LTC3880/-1 can regulate two independent outputs or be configured for a two phase single output. Up to 6 phases can be interleaved and paralleled for accurate sharing among multiple ICs, minimizing input



and output filtering requirements for high current and/or multiple output applications. An integrated amplifier provides true differential remote output voltage sensing, enabling high accuracy regulation, independent of board IR voltage drops. Applications include high current ASIC, FPGA and processor supplies in telecom, datacom, computing and storage markets.

Configurations for the LTC3880/-1 are easily saved to internal EEPROM over the device's I²C serial interface using Linear Technology's LTpowerPlay GUI-based development software. With configurations stored on-chip,

'microblue' nRF8001 Bluetooth low energy solution

Ultra low power (ULP) RF specialist Nordic Semiconductor ASA (OSE: NOD) today launches the µBlue nRF8001. By delivering sub 12.5 mA peak currents and connected mode average currents as low as sub 12 µA (for 1 s connection intervals), µBlue nRF8001 represents the industry's lowest power Bluetooth low energy solution. The device is Nordic's first chip in its brand new µBlue Bluetooth low energy product line. Complementing Nordic's existing ULP nRF24Lxx and ANT™ 2.4 GHz RF product lines with a complete and fully qualified Bluetooth Version 4.0 ("Bluetooth v4.0") low energy solution, the nRF8001 is a highly integrated wireless connectivity solution supplied in a compact 5x5 mm, 32-pin QFN package that when combined with its best-in-class ultra low power consumption makes the device ideally suited to the typically space-constrained coin cell battery operated applications that Bluetooth low energy technology was designed to target.

These include wireless products and sensors designed to be worn on — or carried close to — the end user's body including mobile phone peripherals such as proximity tags and watches, and sports fitness and health sensors, as well as consumer electronic remote controls, and home and industrial automation devices.

In addition, the nRF8001 integrates a DC/DC regulator that, if enabled, can further cut peak currents and average currents by up to 20 percent when running from a coin cell battery source.

The nRF8001 is also the first fully qualified Bluetooth v4.0 low energy design to combine the Radio, Link Layer, and Host into one End Product Listing (EPL), enabling designers to easily create new Bluetooth end products without any additional listing fees. The nRF8001 chip makes it as straightforward as possible for designers to add Bluetooth low energy wireless connectivity to existing applications by integrating a complete Bluetooth v4.0 low energy Radio, Link Layer, and Host stack supporting Peripheral ("slave") role operation, and featuring a simple serial interface supporting external microcontrollers of a designer's own choosing given the individual requirements of their application. The nRF8001 chip also integrates a unique low tolerance 32 kHz RC oscillator that eliminates the need for external 32 kHz crystals, a 16 MHz crystal oscillator supporting low cost 16 MHz crystals, plus an on-chip linear voltage regulator that provides a supply range of 1.9 to 3.6 V as an alternative to its integrated DC/DC regulator.

Production samples and a development kit for the µBlue nRF8001 are available now directly from Nordic Semiconductor. General availability through sales distribution partners will start mid-February this year, with volume shipments beginning in March 2011.

www.nordicsemi.com (110050-VIII)

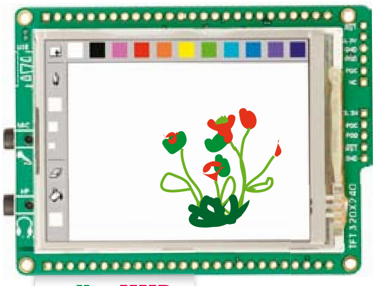
JOIN THE FUN!

MULTIMEDIA BOARDS

New line of multimedia boards can really get your creativity sparks going. You can build all sorts of cool applications that come to your mind. They are ready to meet your demands. They are suported in Visual TFT software, so you can easily build great GUIs for your applications or games in minutes, and have lots of fun along the way.

mikroMMB for PIC18FJ

price: $99



mikroMMB for PIC32

mikroMMB for PIC24

Visual TFT software

Bringing together worlds of design and programming, this software will start a small revolution in ways we build TFT GUIs. Just focus on design, and let the software write the code for you.

price: $99

mikroMMB for dsPIC33

COMPILERS

Visual TFT software and Multimedia boards are best supported with mikroElektronika compilers: mikroC, mikroBasic, and mikroPascal. Having intuitive and fast IDE, powerful compilers and lots of tools, you'll really feel great spending your time programming. Lots of libraries and examples, comprehensive help file and free product lifetime tech support ensure that you get the job done quickly.

mikroC	mikroBasic	mikroPascal
$199	$149	$149

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS
...making it simple

GET IT NOW
www.mikroe.com
www.visualtft.com

Micro-Lite LED illumination ring light

Kimco Distributing Corp., a leading supplier of electronic assembly products and services to the manufacturing, test and repair environments, announces a promotion for the "First Generation" LV 2000 Micro-Lite LED Illumination Ring Light.
As an authorized distributor of O.C. White lighting and magnification products, Kimco is offering a special savings on the LV 2000 Micro-Lite. The ring light, regularly priced at $349, is available for $249.99 while supplies last.
The "First Generation" LV 2000 uses advanced LED technology to offer the versatility and slim design of a fluorescent ring light. Twelve LEDs provide light output approaching fiber optic units and a life span that is claimed to be unrivaled by any other illuminator.

The LV 2000 Micro-Lite is ESD-safe and RoHS compliant. Made in the USA, the ring light has a five-year, 50,000 hour warranty and meets the Energy Star Criteria.

www.gokimco.com (110050-IX)

the controller can power up autonomously without burdening the host processor. Default settings can be optionally configured by external resistor dividers for output voltage, switching frequency, phase and device address. Multiple designs can be easily calibrated and configured in firmware to optimize a single hardware design for a range of applications. The converter loop gain does not change as the power supply parameters are modified, so compensation remains optimized for multiple configurations.
The LTC3880 features high current integrated gate drivers to drive all N-channel power MOSFETs from input voltages ranging from 4.5 V to 24 V, and it can produce ± 0.50% accurate output voltages from 0.5 V to 5.5 V with output currents up to 30 A per phase over the full operating temperature range. Highest efficiency is achieved by sensing the voltage drop across the output inductor (DCR) to sense current, or an external sense resistor can optionally be used. Programmable DCR temperature compensation cancels the TC of the copper inductor to maintain an accurate and constant current limit over a broad temperature range. The device's minimum on-time of just 90 ns makes the LTC3880/-1 ideal for compact high frequency/high step-down ratio applications. Accurate timing across multiple chips and event-based sequencing allow the optimization of power-up and power-down of complex, multiple rail systems. Additional features include constant frequency current mode control with cycle-by-cycle current limit, adjustable soft start, a synchronizable switching frequency, and programmable GPIO pins to indicate part status and to provide autonomous recovery from faults.

www.linear.com/3880 (110050-II)

New cost-effective memory options to 32-bit PIC® microcontrollers with Ethernet, CAN and USB

Microchip announces a new, six-member family of 32-bit PIC32MX5/6/7 microcontrollers that provides the same integrated Ethernet, CAN, USB and serial connectivity peripherals with new, more cost-effective memory options. Additionally, design enhancements also provide lower power consumption of 0.5 mA/MHz active current, higher Flash memory endurance of 20k read/write cycles, and better EEPROM emulation capability. By maintaining common pin-outs, the PIC32 portfolio also provides designers with the optimum balance of memory and cost for their high-performance applications as well as a seamless migration path for scalability and flexibility.
The latest 80 MHz PIC32 microcontroller family helps embedded designers to lower their costs without sacrificing performance or functionality. The raw performance of the MIPS32® M4K® core has been maximised to achieve best-in-class performance of 1.56 DMIPS/MHz, with integrated Ethernet, CAN, USB and multiple serial communication channels, in addition to more cost-effective memory options. The family provides 32 Kbytes of RAM and up to 140 Kbytes of Flash. Each of the six new microcontrollers is available in five different pin-compatible packages: 100-pin TQFP 12x12mm, TQFP 14x14mm and BGA packages, as well as 64-pin TQFP and QFN packages.

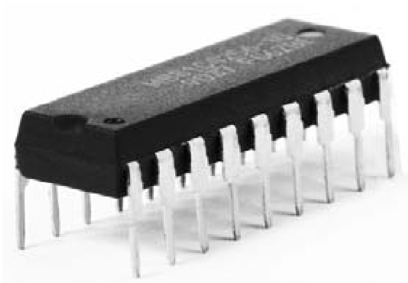
Example applications for the new PIC32MX5/6/7 family include Communications: point-of-sale terminals, Web servers and multi-protocol bridges. Industrial: automation controllers. Medical devices and Security: monitoring equipment. Consumer: audio, MP3 decoders, displays, fitness equipment and small appliances. Automotive: aftermarket products, car alarms and GPS.

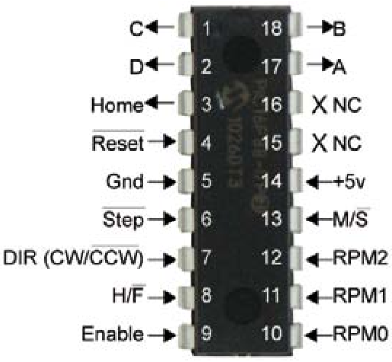
Two Starter Kits support easy development for Ethernet-based designs (PIC32 Ethernet Starter Kit DM320004 at $72) and USB-based designs (PIC32 USB Starter Kit II DM320003-2 at $55).
There is also a $25 plug-in module (MA320003) for developing the PIC32MX5/6/7 family using the Explorer 16 Development Board (DM240001).

www.microchip.com/get/BWUC (110050-III)

Stepper motor controller IC

Uniquely, the new USMC-01 Stepper motor controller IC from Images Scientific can operate as a slave or master. As a slave it runs under another microcontroller (or PC) or it can operate in auto-run (master) mode which allows the user to add a few switches and run the stepper motor manually. RPM's of stepper motors driven by the IC are switch selectable.





Tne USMC-01 stepper Motor chip generates control signals that can be used with for both unipolar & bipolar stepper motors with appropriate drivers like the L298 & L293. The new device is compatible with 4 phase unipolar / 2 phase bipolar motors and can operate in master/slave or standalone free running mode. It allows eight RPM selections in free running mode and is compatible with the L298, L293 drivers as well as discrete transistors. For half/full wave drive the chip offers step modes, direction control and enable.

www.imagesco.com/stepper/usmc.htm

(110050-V)

Configurable potentiostat

National Semiconductor's configurable sensor AFE ICs and WEBENCH® Sensor AFE Designer enable the design engineer

to select a sensor, design and configure the solution and download configuration data to the sensor AFE. A typical sensing application that today may require several boards and up to 25 components is reduced to just one of National's ICs. Weeks or months to create a sensor system design is reduced to just minutes using National's new products and tools.
The first in a family of products, National's two configurable sensor AFE products are each customized to a specific sensor application and have a variety of features, including programmable current sources, voltage reference options and adjustable sample rates.
The LMP91000 is the industry's first fully configurable, low-power potentiostat that provides a complete, integrated signal path solution between a sensor and analogue-to-digital converter (ADC). The LMP90100 is the industry's first multi-channel, low power, 24-bit sensor AFE with true continuous background calibration and diagnostics for high performance transmitter and transducer applications.

The WEBENCH Sensor AFE Designer includes technical specifications for hundreds of temperature, pressure and chemical sensors. The database features products from Omega Engineering, Inc., Honeywell Sensing & Control, Tempco Electric Heater Corp. and All Sensors Corp.

http://bit.ly/LMP90100 (110050-VI)

New from Parallax: Scribbler 2 robot

The new S2 robot from Parallax Inc. is suitable for a whole variety of programming skills. The Scribbler robot arrives pre-programmed with eight demo modes, including light-seeking, object detection, object avoidance, line-following, and art. Place a Sharpie marker in the pen port and it will scribble as it drives. Next, use the Graphical User Interface (S2 GUI) tile-based programming tools, or modify the Propeller source code in our BASIC-like Spin language. Through the use of third-party tools you can also program the S2 on a Mac or under Linux, in PropBASIC and C (resources for these languages will follow the release). The S2 is fully compatible with the Georgia Tech IPRE Fluke, too.

The S2 GUI is backward-compatible with the original S1 GUI. However, coders will use Spin for the Propeller instead of PBASIC as they did for the BASIC Stamp in the S1. Our examples make the transition easy. The benefits and flexibility of Spin in multi-core systems provides easy compartmentalization of S2 subroutines that run concurrently with shared memory. Controlling motors, managing sensors, and interfacing with the hacker port can be done concurrently even while playing sound; the Propeller makes it all possible. Scribbler 2 retails at $129.99.

www.Parallax.com (search '28136')

(110050-X)

Everything on a Single Chip

The world of SoCs

By Dr Thomas Scherer (Germany)

Several technical terms in electronics are imprecisely defined, and ‘SoC’ is a prime example. The term can be used so narrowly as to cover only highly complex industrial controller ICs or so broadly as to cover almost any device capable of computing something, from the humblest microcontroller to a single-chip PC. We will take a wander through the world of smaller and larger devices and try to find out what makes an IC an SoC.

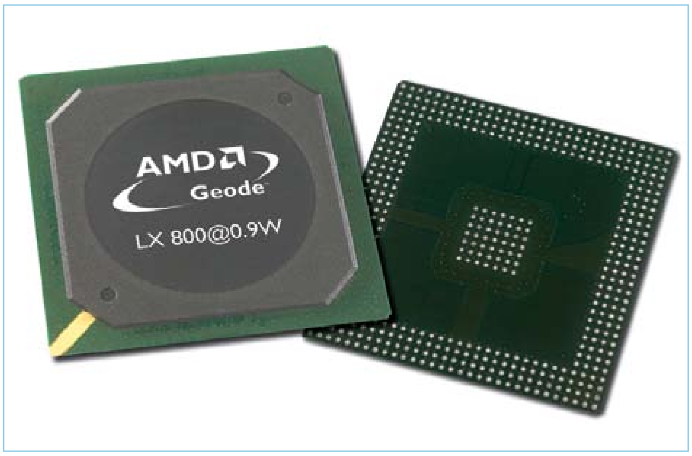


Figure 1. The AMD Geode family is a range of particularly energy-efficient and highly integrated devices which incorporate virtually all the main components of a PC (with the exception of memory and interface chips).

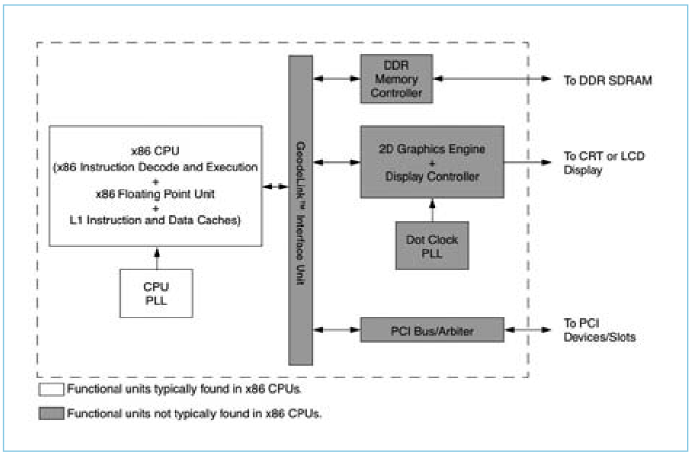


Figure 2. The block diagram of the Geode device shows the x86 CPU core, memory manager, GPU and PCI slot interface. Practically a complete PC on a chip!

This confusion over the definition of the term SoC (system on a chip) can be seen (at the time of writing, at least) in Wikipedia. The article in the English version [1] starts with a picture of an AMD Geode single-chip processor (**Figure 1**), which corresponds to many people’s idea of an SoC. As the diagram in **Figure 2** shows, the device contains a complete x86-family CPU along with a wide range of peripheral function blocks, all integrated into one chip; normally these extra peripherals would be in separate chips. However, the SoC article in the German version of Wikipedia [2] the author found initially starts with a picture of a single-board computer. Is that really a ‘system on a chip’?

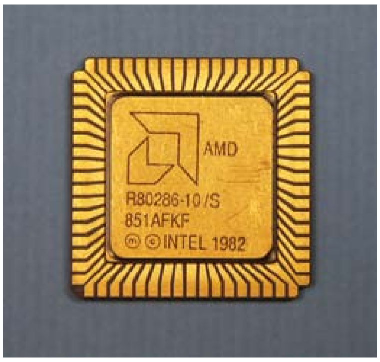
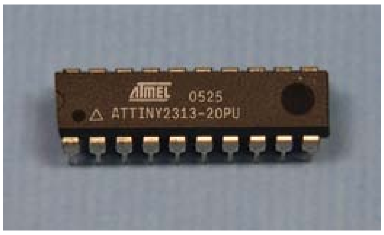
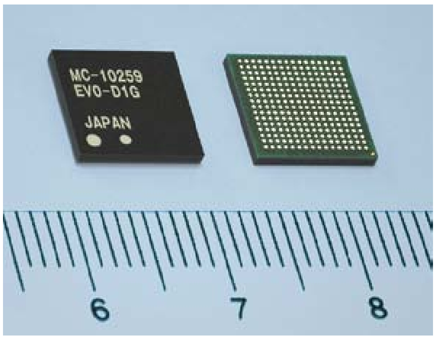
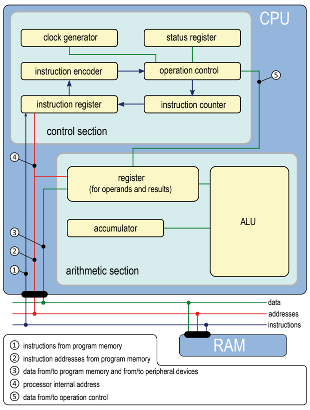
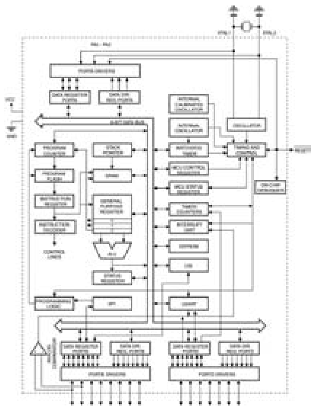
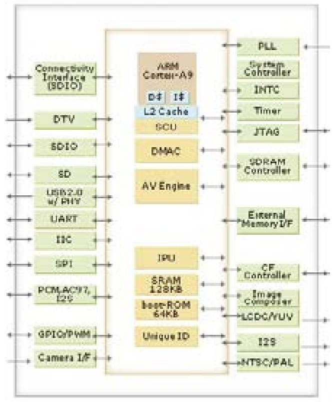
Compare and contrast

It is easier to understand the distinction between an SoC and a ‘normal’ processor by looking at concrete examples rather than by trying to find a watertight definition. In between we have microcontrollers, and it can be hard to draw the line between microcontrollers and SoCs. The usual definition has it that a microcontroller consists of a processor with a number of additional peripheral functions integrated onto the same chip, but this sounds suspiciously like an SoC. Also, microcontrollers can be thought of as specialised single-chip computers, which blurs the boundary further. On the other side, it has long been the case that many devices referred to as CPUs contain more than just a processor, with various peripherals included on the IC to reduce system chip count. So maybe we should give up trying to distinguish processors, microcontrollers and SoCs, and treat the three terms as synonyms?

Although the boundaries between these categories are fluid, the three terms still perhaps retain some value insofar as they reflect the development and function of the devices they refer to. The history of chip development is one of gradual evolution rather than one of sudden revolutions: it is a story of ever-greater levels of integration offering parallel increases in computing power and in device functions.

Table 1 shows examples of each of the three types of IC. A venerable 286-series CPU serves as an example processor, and we have shown an eight-bit Atmel microcontroller, although today very fast 32-bit devices exist with several hundred kilobytes of on-board RAM. Mod-

Table 1. Processors, microcontrollers and SoCs
Side-by-side comparison of typical devices. The degree of specialisation of the devices increases from left to right.

Processor	Microcontroller	SoC
 AMD x86-class processor	 Atmel ATtiny2313 eight-bit microcontroller	 Renesas EVO-D1G integrated multimedia player
 Control logic for scheduling instructions, a processing core, and a system bus form the minimum set of features. (Image: Bernhard Ladenthin, German edition of Wikipedia)	 Processor, internal bus and peripherals (memory, A/D and D/A converters, external buses, I/O, timers and so on).	 Processor, internal bus(es) and sophisticated peripherals (memory, video functions, external bus interfaces, special-purpose units and so on).

ern PC processors undeniably offer the greatest amount of computing power, but current SoCs for multimedia applications are no slouches either: think for example of the gigahertz clock speeds of the SoCs in the iPhone and iPad. The block diagram above shows the diversity of technologies present in modern ARM-based SoCs. In general we can say that the processor is the least specialised device, followed by the microcontroller and then the SoC; in terms of processing power (and of cost) the processor comes top, followed by the SoC and then the microcontroller.

Applications

From the remarks above we can see that an SoC can be thought of as a kind of 'super microcontroller', specially tailored for a particular application area. The main driving force behind SoC development is the integration of as many functions as possible onto a single chip, resulting in higher reliability and reduced costs and, thanks to modern foundry processes, ever more computing power for less electrical power. This last factor is critical for mobile devices such as smartphones and tablet PCs. However, even bread-and-butter SoCs in more mundane applications like WLAN modules confer advantages in product size and power consumption. In control applications from washing machines to heavy industry, where microcontrollers are in an ideal position, SoCs are becoming more and more popular because of their advantages in speed, size and cost.

Besides these advantages, low development costs are another factor in SoCs' appeal. Because the devices sport sophisticated integrated peripheral functions tailored to a particular application area, the manufacturers as a rule normally offer corresponding specialised libraries along with their development environments. This allows the programmer to deal with the on-chip peripherals at a high level of abstraction. Compiling and maintaining device firmware is also simplified when the developer does not have to implement drivers for a diverse collection of peripheral ICs.

Finally, the SoC developer must adapt to a broader spectrum of devices and can expect a much deeper level of support from the manufacturer. Of course development boards exist for SoCs, made both by the SoC manufacturers and by third parties, which make prototyping easier. Another article in this edition looks at this in more detail.

Figure 3. The Analog Devices ADuCRF101 is an SoC with a 32-bit ARM processor core and integrated UHF transceiver. The device is particularly suitable for wireless sensor network applications.

We have also dedicated an article to PSoCs made by Cypress elsewhere in this edition. In contrast to the SoCs discussed here, these chips include analogue and digital functional blocks that can be configured in software. This allows designers in effect to make their own SoCs.

Example 1: radio on a chip

On their website Analog Devices tout the ADuCRF101 [3] as a 'Precision Analog Microcontroller ARM Cortex M3 with ISM band Transceiver', and the integration of analogue RF electronics means that we can justify calling it an SoC. The chip also offers a range of other peripherals such as an A/D converter, ports and timers. In this respect, and in terms of the processing power and amount of memory offered, it is much like any modern microcontroller; as mentioned above, the boundaries are not clear-cut. A complete feature list and further technical information can be found in the accompanying text box.

The diagram from Analog Devices in **Figure 3** clearly shows the type of application the chip is aimed at: battery-powered wireless sensor network devices such as energy meters, medical telemetry, home automation systems, item tracking systems and security devices. Compared to a conventional microcontroller-based approach with external transceiver hardware we can reduce both board area (to just 81 mm²) and current consumption, as the analogue part of the device offers a range of sleep modes. Furthermore, the IC supports the DASH7 standard [4], which is based on ISO 18000-7, for flexible wireless sensor networks operating on ISM-allocated frequency 433.92 MHz, as well as IEEE 802.15.4-based networks. There are evaluation boards and accessories available for the ADuCRF101, as well as a device-specific compiler with a software library for interfac-

Analog Devices ADuCRF101: technical details

UHF transceiver
Frequency: 862 MHz to 928 MHz and 431 MHz to 464 MHz
Data rate: 1 kb/s to 300 kb/s
Sensitivity: –107.5 dBm at 38.4 kb/s
Modulation scheme: binary FSK (in hardware)

Microcontroller
CPU type: ARM Cortex M3 (32 bit)
Clock frequency: 16 MHz

Internal memory
Flash: 128 kB
SRAM: 16 kB

Integrated peripherals
Serial: UART, I2C and SPI
Parallel: 29 GPIO pins
Timer: wake-up, watchdog, eight-channel PWM

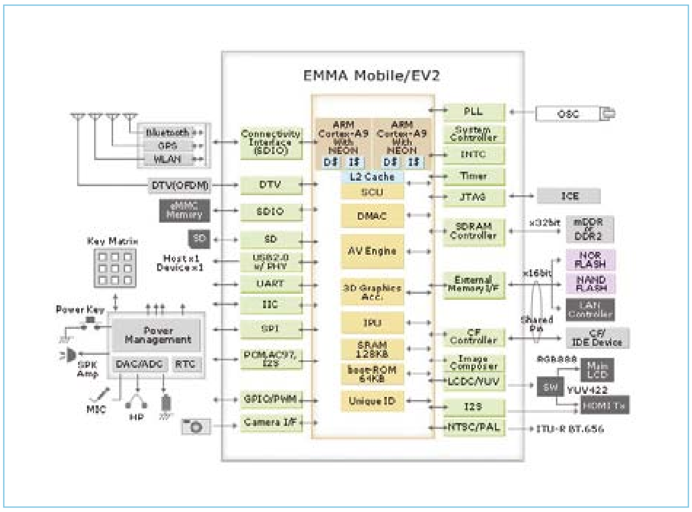


Figure 4. Block diagram of the Renesas EMMA Mobile EV2. This dual-core device has enough processing power that a complete smartphone can be realised with just a couple of extra ICs.

Renesas EMMA Mobile EV2: technical details

Graphics, video and audio
2D/3D graphics: 14.7 Mpolygon/s, 500 Mpixel/s
Video (1080p): H.264, MPEG2, MPEG4, VC-1
Audio: MP3, AAC, HE-AAC, WMA, AC-3 Dolby digital 5.1

Microcontroller
CPU type: ARM Cortex A9/NEON dual core
Clock frequency: 533 MHz

Memory controller
Flash: NOR, NAND, eMMC
RAM: mobile DDR-400 / DDR2-533

Power supply
Voltage: 1.8 V to 3.6 V
Current (sleep mode): 1.6 µA
Current (receive): 12.8 mA
Current (transmit): 9 mA to 32 mA
(CPU in each case in power-down mode)

Analogue I/O
A/D converter: multi-channel
Resolution: 12 bit
Sample rate: 1 MS/s
Integrated voltage reference and temperature sensor

Miscellaneous
Package: 64 pin LFCSP (9 mm x 9 mm)
Temperature range: –40 °C to +85 °C

ing to the built-in peripheral functions and tools including a current consumption and battery life calculator.

Example 2: multimedia on a chip

A rather different class of device is represented by the single-chip multimedia solutions from Renesas. The EMMA Mobile family [5] includes a dual-core variant (the 'EV2') which contains two ARM Cortex A9 cores. Running at 533 MHz it includes a wide range of integrated peripheral functions and has enough computing power to decode 1080p HD video smoothly. With built-in 2D/3D graphics, an LCD driver, video and audio decoders and a range of interfaces, the device just needs the addition of an LCD panel and a couple of extra chips for the keyboard interface, power management, radio (WLAN, Bluetooth and GPS) and memory to make a complete smartphone. As you can see, this single-chip solution comes very close to embodying the idea of an SoC in its purest form.

Looking at the innards of the device in **Figure 4** and at the text box giving its technical specifications gives an idea of its enormously high level of integration. Renesas talks grandly of the 'next generation of SoCs'. The various versions available include the 'EV0-D', a chip measuring just 9 mm square, the same size as the much simpler Analog Devices chip we looked at above. An SoC like this provides almost everything needed to make a multimedia player or e-book reader or to add network TV functions to an existing set design.

Power supply
Voltage: 1.1 V to 1.3 V (without memory)

Integrated peripherals
Serial: 4 x UART, 2 x I2C, SPI or audio
Parallel: configurable GPIOs
USB: 1 x USB 2.0

Other functions
Image resizer, rotator and composer (for LCD)
LCD driver: RGB565/666/888
Camera interface: eight-bit parallel

The future of SoCs

With today's ever-shortening product design cycles and ever-increasing demands for processing power, and the continuing expansion of the market for mobile devices, it is almost inevitable that specialised SoCs will become more and more important. The trend towards high-end SoCs has now progressed so far that the millions of sales of smartphones (as well as the waves of entirely new classes of product such as the iPad and tablet PCs) have meant that the SoC market has grown at the expense of that for netbook and notebook CPUs. Semiconductor giants Intel and AMD, who do not have a strong presence in the SoC market, have profited less from this than the numerous smaller manufacturers specialising in the embedded market, as well as ARM, whose processor cores sit at the heart of many SoC designs. Fortunately the SoC market at the moment is far from monolithic: Wikipedia [6] lists 74 SoC manufacturers. This level of competition ensures that there is a wide range of SoC solutions to choose from, and, with the promise of huge sales, accelerates the development of new designs.

(100855)

Internet Links

[1] http://en.wikipedia.org/wiki/System-on-a-chip

[2] http://de.wikipedia.org/wiki/System_on_a_Chip (in German)

[3] www.analog.com/en/analog-microcontrollers/analog-microcontrollers/aducrf101/products/product.html

[4] www.dash7.org

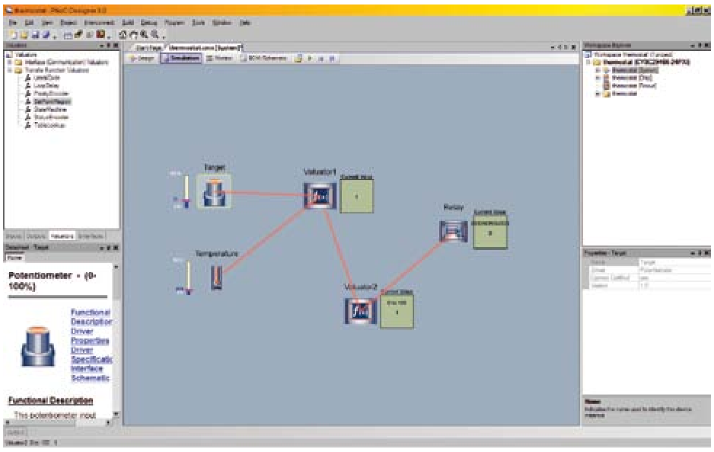
[5] www2.renesas.com/mobile/en/emma_mobile/em_ev.html

[6] http://en.wikipedia.org/wiki/List_of_system-on-a-chip_suppliers

PSoC Designer

Electronics projects, a walk in the park…

Our daily life is filled with electronics — more and more with each passing day — and it opens up countless possibilities to inspired creators: communicating objects, mechanisms, gadgets, peripherals, to mention just the consumer field. Imaginative electronics technicians are revelling in this, and know easily how to make their ideas take shape. But what about creative people who don't have electronics expertise — do they have to give up their dreams?



By Philippe Larcher (France)

All of you who are into robotics or mechanics, designers, model enthusiasts, hobbyist, DIY-ers — what would you say to a tool that lets you design your electronics project in the form of an interconnection of graphic objects, and which from that provides you with a list of the components to be used, the circuit diagrams, and the microcontroller already programmed, ready to use? Hard to believe, we know — so let's try and convince you with an example.

Let's suppose you want to make a thermostat that activates a fan when the ambient temperature exceeds a level set using an adjusting knob, but you don't have any special skills in electronics, whether for converting analogue signals into digital or writing a program for a microcontroller.

The tool we're going to be using is called **PSoC Designer5**, it is available from **Cypress** [1], and can be downloaded and installed free over the Internet. It has two operating modes: the "Chip-level Design" mode is intended for experienced electronics technicians, while the "System-level Design" mode is for non-specialists — this is the one we're going to be using for our demonstration.

The operation takes place in four steps: selecting the input and output functions, defining the transfer functions, simulation, and to conclude, generating the application.

1. Selecting the input and output functions

PSoC Designer has a very comprehensive catalogue of input and output functions, grouped into categories, as shown in **Figure 1**. Among the input functions, the category "Temperature", which has been listed in full by way of an example, covers over 20 types of sensors that can be used directly.

Our application required two input devices: a temperature sensor and a control to set the temperature required. Summarized data is given for each component in the list to help guide our choice. For our project we're going to pick a cheap LM35DZ sensor, which provides an output voltage proportional to temperature over the range 0–100 °C. For the control, we're going to use a potentiometer (category "Tactile"), whose output voltage is converted into a standardized value, from 0–100. For controlling the fan, we're going to pick, from the category "Digital Output" in the output functions cata-

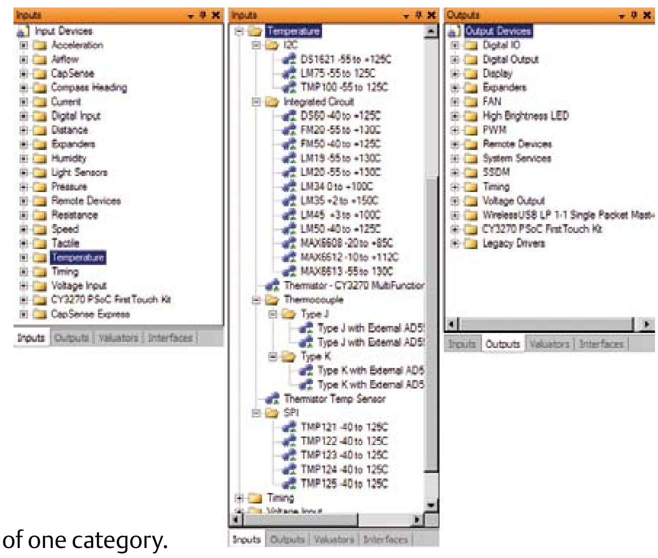


Figure 1. List of the input and output functions, with details of one category.

logue, a relay with a coil driven at 5 V.

Each function is added to the diagram by dragging it with the mouse from its original list, then renamed if necessary. **Figure 2** represents the graphical project after selecting the input/output functions: temperature sensor, potentiometer for setting the temperature (target) and relay.

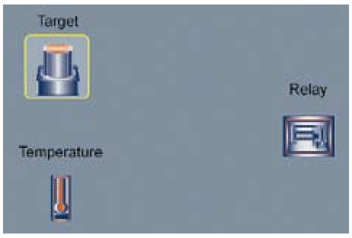


Figure 2. Installing the input and output functions.

2. Defining the transfer functions

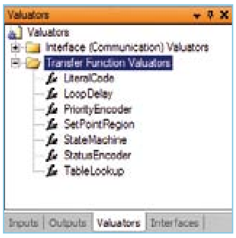


Figure 3. Transfer functions – "Valuators".

In PSoC Designer terminology, transfer functions are called "Valuators". **Figure 3** lists the "Valuators" available.

Our project's transfer function can be described very simply: if the measured temperature is higher than the set temperature, the relay is energized; if not, it is de-energized. So all we need is for Valuator1 to constantly assess the difference "Temperature – Target". To do this, we're going to choose a Valuator of the "Priority encoder" type, which we'll configure as shown in **Figure 4**. Since the condition "if 1" is always true by definition, the instruction means that the Valuator output is at all times equal to the difference "Target – (Temperature/10)" (the data sheet for the temperature sensor tells us that the value it provides is scaled by a factor of 10, i.e. varies from 0–1000 when the actual temperature varies from 0–100 °C, which is why it is divided by 10 in the equation). Fahrenheit fans have some work to do here.

But attention: if the output of Valuator1 drove the Relay output directly, it would lead to unstable operation when the temperature is very close to the set point — which is by definition the most common situation. So it's a good idea to add a stability range, in other words a hysteresis function, around the value 0 of Valuator1, which prevents the relay switching too frequently. To achieve this, we're going to cascade a second Valuator, of the 'Set Point Region' type, which lets us segment a continuous variable into partitioned intervals, where each partition can be protected by a hysteresis.

In our case, the continuous variable is the output of Valuator1 and just two intervals are created, the first comprising the negative values of Valuator1 and the second the positive values, with a hysteresis range that we're going to set at 1 °C (1 K), as shown in **Figure 5**.

All that remains is to connect Valuator2 to the relay, which we do via the transfer function included in the Relay output, as shown in **Figure 6**.

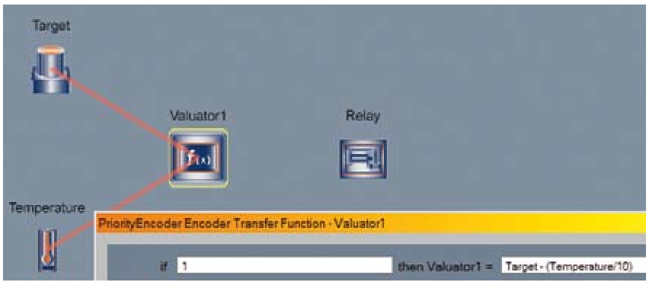


Figure 4. Configuring the main transfer function.

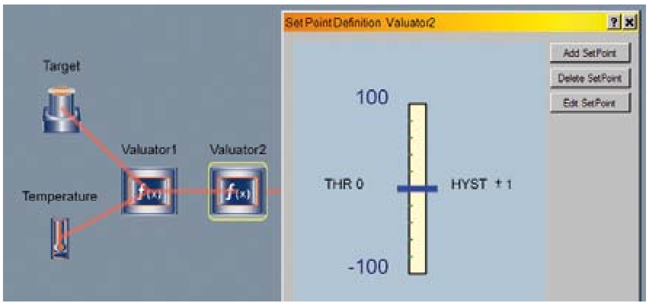


Figure 5. The Valuator2 transfer function.

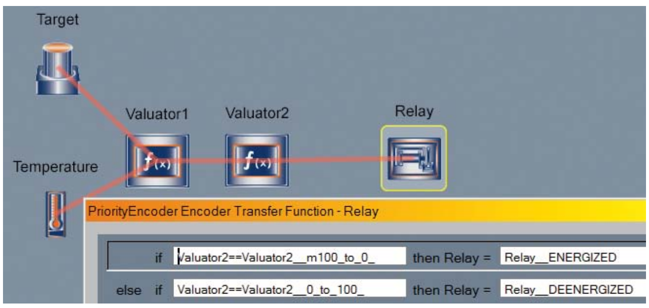


Figure 6. The relay transfer function.

3. Simulation

Once the inputs, outputs, and internal functioning have been defined, it's time to check that the whole thing works, by going from the "Design" tab to the "Simulation" tab. Some extra icons appear on the previous graphic representation, which let us control and graphically change the input parameters, and display the output values along with the internal values, as shown in **Figure 7**.

Simulation is generally done manually, but can also be executed automatically by playing back a previously-recorded scenario.

In our example, the Temperature and Target inputs can be adjusted graphically using the cursor icons, the internal states and outputs react as they are changed. In this way, we can verify correct energizing of the relay depending on the temperature and the displayed set point, along with the effectiveness of the hysteresis range.

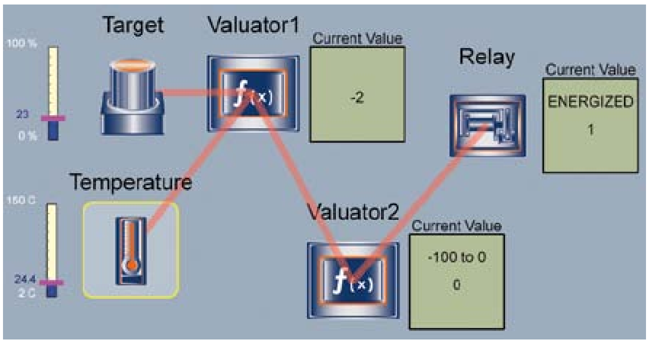


Figure 7. Graphical simulation.

It is of course possible to go back and forth as many times as you like between the Design and Simulation modes to correct errors, refine the operation, or modify parameters.

4. Creating the application

The final stage in the process is generating the application, when the project construction file is also created. To do this, the software is only lacking one piece of information, namely which PSoC microcontroller is chosen as the heart of the project. PSoC Designer displays a list of models capable of accepting the project, each described succinctly, as shown in **Figure 8**. We're going to choose the CY8C29466, which is one of the most commonly used versions, as it has numerous internal functions and is available in a DIL package, among other reasons.

The project can now be compiled, at the end of which operation four documents will be provided to the user: circuit diagrams, component list, documentation (data sheet) and programming file.

Figure 9 shows the circuit diagrams produced by the software: the microcontroller itself with its pin-out, the wiring to be done in the event of in situ programming, and the circuit diagrams for the input and output devices.

The components list (**Figure 10**) summarizes the components needed and their specifications. If necessary, the user can ask for the component part numbers to be displayed for a specific distributor, for example Digikey.

PSoC Designer also produces the documentation for the project, which summarizes the configuration of each of the external or internal functions, and to conclude, generates the file for programming the microcontroller, to be used when doing the actual construction.

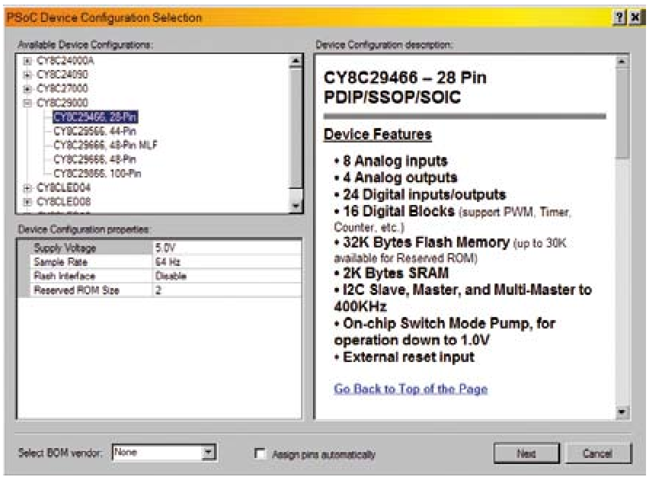


Figure 8. Choice of microcontroller.

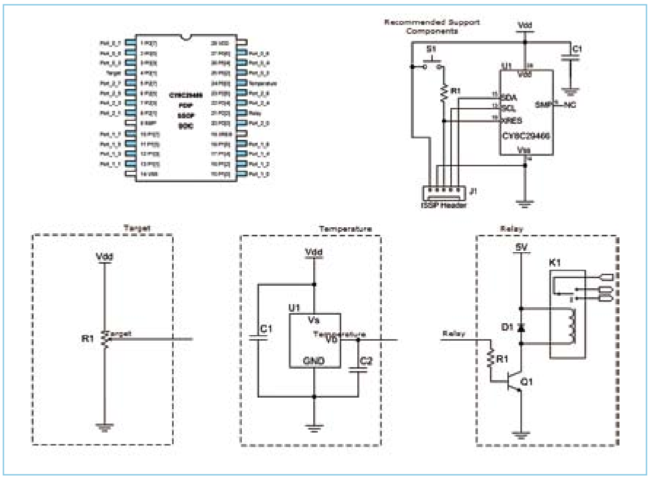


Figure 9. The circuit diagrams produced by PSoC Designer.

5. Construction

Label	Device	Value	Notes	Power	Volts
thermostat_R1	Resistor	1k	-		
thermostat_J1	100mil	-	ISSP Connector		
thermostat_S1	Button	-	Reset Switch - Normally Open		
thermostat_C1	Capacitor	0.1uF	-		
thermostat_U1	CY8C29466	-	-		
Consigne_R1	Potentiometer	10k	-		
Relais_R1	Resistor	1kOhms	-	0.125W	
Relais_Q1	Transistor	2N2222	NPN		
Relais_D1	Diode	1N4003	-		
Relais_K1	Relay		-		
Temperature_U1	IC	LM35DZ	+2'C to +100'C		
Temperature_C1	Capacitor	0.1µF	-		6.3V
Temperature_C2	Capacitor	0.1µF	-		6.3V

Figure 10. The components list.
Note the part numbers by block.

Up to now, PSoC Designer has been doing most of the work; now it's our turn. We can consider two construction methods: using an existing evaluation board and adding to it the few elements that are lacking; or producing the whole of the electronic circuit ourselves. These two methods can be used one after the other, the first for prototyping, the second for the final construction. In all cases, programming the microcontroller requires a Miniprog programmer, which is includ-

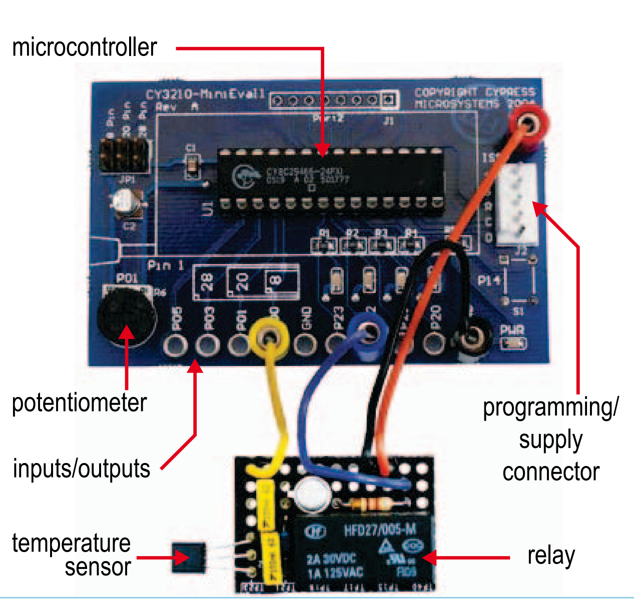


Figure 11. Construction using a CY3210-MiniEval1 board.

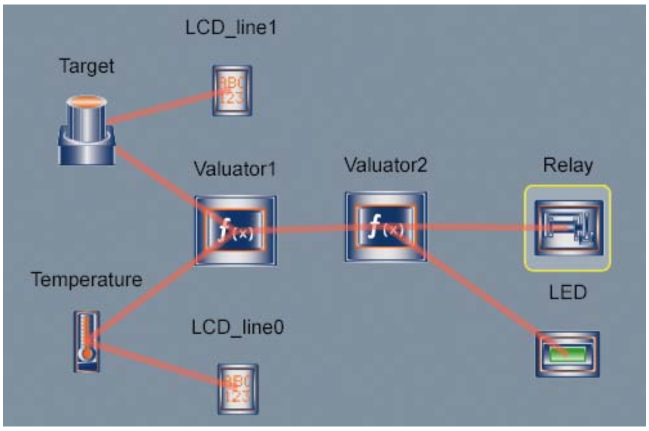


Figure 12. Project with LCD display and LED.

ed in the Cypress development kits.

Cypress offer several development/evaluation kits at affordable prices, including: the CY3210-Miniprog1 entry-level kit (around €/£ 45) comprises a Miniprog1 USB programmer, a CY8C29466 microcontroller in a 28-pin DIL package, a compact test board (CY3210-MiniEval1) with a socket for the microcontroller, a few input/outputs, including a potentiometer (which can be used for our project), and a 5-pin connector for programming the microcontroller. It's worth noting that the Miniprog1 can also be used as a 5 V PSU.

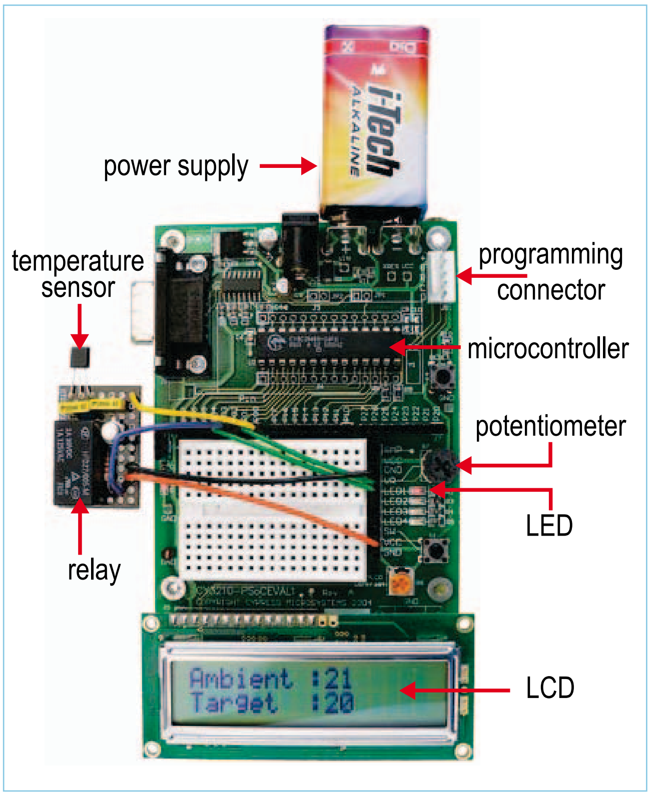
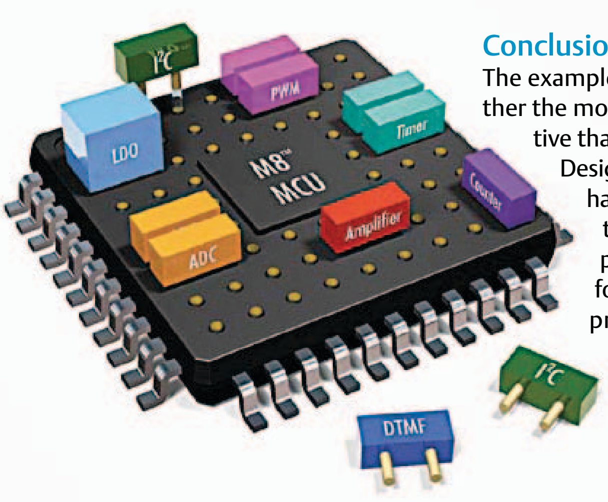


Figure 13. Construction using a CY3210-Eval1 board.

The most universal kit is the CY3210-Eval1 (around €/£ 100) comprises a Miniprog1, a CY8C29466 microcontroller in a DIL package, and a development board with prototyping area, several input/output interfaces, a semi-graphic LCD display, a connector for in situ programming, and several powering options.

Figure 11 shows the thermostat built using the CY3210-MiniEval1 board. The thermostat functions are divided between the development board (microcontroller and its programming/power interface, potentiometer for setting the temperature) and a specially-made add-on board ("Temperature" and "Relay" functions from **Figure 9**). The 5 V supply for the whole project is provided by the Miniprog programmer, not shown in this photo.

The project has also been built on a CY3210-Eval1 board. For this second version, given the extra possibilities offered by this board, we've added to the original project an LCD display for displaying the set temperature and the ambient temperature, and an LED giving a visual indication of the relay state. These additions, produced by following the same procedure described in this article, took all of ten minutes, including simulation. **Figure 12** shows the completed PSoC Designer project, and **Figure 13** how it was built on the CY3210-Eval1 board, powered on this occasion by a battery.



Conclusion

The example we've been looking at is neither the most complex nor the most creative that could be handled using PSoC Designer. Certain PSoC components have internal functions that open the door to really innovative projects: capacitive detection for producing tactile interfaces, proximity detection, RF transmission for remote controls or wireless sensors, I²C, SPI, USB, etc. communication. All these functions are incorporated in PSoC Designer's library and can be used according to the simple principle described in this article, without any special knowledge of electronics or programming languages.

(090076)

Internet links

[1] www.cypress.com

Quasar Electronics Limited
PO Box 6935, Bishops Stortford
CM23 4WP, United Kingdom
Tel: 01279 467799
Fax: 01279 267799
E-mail: sales@quasarelectronics.com
Web: www.quasarelectronics.com

Postage & Packing Options (Up to 0.5Kg gross weight): **UK Standard 3-7 Day Delivery** - £4.95; **UK Mainland Next Day Delivery** - £11.95; **Europe (EU)** - £11.95; **Rest of World** - £12.95 (up to 0.5Kg)
!Order online for reduced price UK Postage!
We accept all major credit/debit cards. Make cheques/PO's payable to Quasar Electronics. Prices include 17.5% VAT.
Please visit our online shop now for details of over 500 kits, projects, modules and publications. Discounts for bulk quantities.

01279 Credit Card Sales 467799

QUASAR electronics
The Electronic Kit Specialists Since 1993

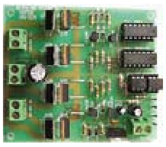
Motor Drivers/Controllers

Here are just a few of our controller and driver modules for AC, DC, Unipolar/Bipolar stepper motors and servo motors. See website for full range and details.

Computer Controlled / Standalone Unipolar Stepper Motor Driver

Drives any 5-35Vdc 5, 6 or 8-lead unipolar stepper motor rated up to 6 Amps. Provides speed and direction control. Operates in stand-alone or PC-controlled mode for CNC use. Connect up to six 3179 driver boards to a single parallel port. Board supply: 9Vdc. PCB: 80x50mm.
Kit Order Code: 3179KT - **£15.95**
Assembled Order Code: AS3179 - **£22.95**

Computer Controlled Bi-Polar Stepper Motor Driver

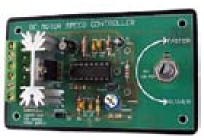


Drive any 5-50Vdc, 5 Amp bi-polar stepper motor using externally supplied 5V levels for STEP and DIRECTION control. Opto-isolated inputs make it ideal for CNC applications using a PC running suitable software. Board supply: 8-30Vdc. PCB: 75x85mm.
Kit Order Code: 3158KT - **£23.95**
Assembled Order Code: AS3158 - **£33.95**

Bi-Directional DC Motor Controller (v2)

Controls the speed of most common DC motors (rated up to 32Vdc, 10A) in both the forward and reverse direction. The range of control is from fully OFF to fully ON in both directions. The direction and speed are controlled using a single potentiometer. Screw terminal block for connections.
Kit Order Code: 3166v2KT - **£22.95**
Assembled Order Code: AS3166v2 - **£32.95**

DC Motor Speed Controller (100V/7.5A)



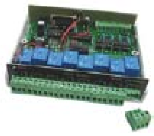
Control the speed of almost any common DC motor rated up to 100V/7.5A. Pulse width modulation output for maximum motor torque at all speeds. Supply: 5-15Vdc. Box supplied. Dimensions (mm): 60Wx100Lx60H.
Kit Order Code: 3067KT - **£18.95**
Assembled Order Code: AS3067 - **£26.95**

Most items are available in kit form (KT suffix) or assembled and ready for use (AS prefix).

Controllers & Loggers

Here are just a few of the controller and data acquisition and control units we have. See website for full details. Suitable PSU for all units: Order Code PSU445 £7.95

8-Ch Serial Isolated I/O Relay Module



Computer controlled 8-channel relay board. 5A mains rated relay outputs. 4 isolated digital inputs. Useful in a variety of control and sensing applications. Controlled via serial port for programming (using our new Windows interface, terminal emulator or batch files). Includes plastic case 130x100x30mm. Power Supply: 12Vdc/500mA.
Kit Order Code: 3108KT - **£69.95**
Assembled Order Code: AS3108 - **£84.95**

Computer Temperature Data Logger

4-channel temperature logger for serial port. °C or °F. Continuously logs up to 4 separate sensors located 200m+ from board. Wide range of free software applications for storing/using data. PCB just 45x45mm. Powered by PC. Includes one DS1820 sensor.
Kit Order Code: 3145KT - **£19.95**
Assembled Order Code: AS3145 - **£26.95**
Additional DS1820 Sensors - **£3.95 each**

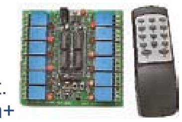
Rolling Code 4-Channel UHF Remote

State-of-the-Art. High security. 4 channels. Momentary or latching relay output. Range up to 40m. Up to 15 Tx's can be learnt by one Rx (kit includes one Tx but more available separately). 4 indicator LED 's. Rx: PCB 77x85mm, 12Vdc/6mA (standby). *Two and Ten channel versions also available.*
Kit Order Code: 3180KT - **£49.95**
Assembled Order Code: AS3180 - **£59.95**

DTMF Telephone Relay Switcher

Call your phone number using a DTMF phone from anywhere in the world and remotely turn on/off any of the 4 relays as desired. User settable Security Password, Anti-Tamper, Rings to Answer, Auto Hang-up and Lockout. Includes plastic case. Not BT approved. 130x110x30mm. Power: 12Vdc.
Kit Order Code: 3140KT - **£74.95**
Assembled Order Code: AS3140 - **£89.95**

Infrared RC Relay Board



Individually control 12 on-board relays with included infrared remote control unit. Toggle or momentary. 15m+ range. 112x122mm. Supply: 12Vdc/0.5A
Kit Order Code: 3142KT - **£59.95**
Assembled Order Code: AS3142 - **£69.95**

New! 4-Channel Serial Port Temperature Monitor & Controller Relay Board

4 channel computer serial port temperature monitor and relay controller with four inputs for Dallas DS18S20 or DS18B20 digital thermometer sensors (£3.95 each). Four 5A rated relay channels provide output control. Relays are independent of sensor channels, allowing flexibility to setup the linkage in any way you choose. Commands for reading temperature and relay control sent via the RS232 interface using simple text strings. Control using a simple terminal / comms program (Windows HyperTerminal) or our free Windows application software.
Kit Order Code: 3190KT - **£69.95**

PIC & ATMEL Programmers

We have a wide range of low cost PIC and ATMEL Programmers. Complete range and documentation available from our web site.

Programmer Accessories:
40-pin Wide ZIF socket (ZIF40W) £14.95
18Vdc Power supply (PSU120) £19.95
Leads: Serial (LDC441) £3.95 / USB (LDC644) £2.95

USB & Serial Port PIC Programmer

USB/Serial connection. Header cable for ICSP. Free Windows XP software. Wide range of supported PICs - see website for complete listing. ZIF Socket/USB lead not included. Supply: 16-18Vdc.
Kit Order Code: 3149EKT - **£49.95**
Assembled Order Code: AS3149E - **£59.95**

USB 'All-Flash' PIC Programmer

USB PIC programmer for all 'Flash' devices. No external power supply making it truly portable. Supplied with box and Windows Software. ZIF Socket and USB lead not included.
Assembled Order Code: AS3128 - **£49.95**

See website for full range of PIC & ATMEL Programmers and development tools.

No.1 FOR KITS

www.quasarelectronics.com

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

SatFinder
TV dish alignment using GPS

By Klaus Hirschelmann (Germany)

Those of you who regularly need to realign a satellite TV dish will find this gadget extremely valuable. Caravan owners and campers on long journeys who crave their home TV channels can now keep up with developments in sports, news and the soaps back home with the help of the SatFinder. This GPS based design includes a database containing positional information of a number of popular TV satellites. With the help of GPS data it calculates the precise angles to find the satellite first time!

To align a satellite TV receiver dish with a chosen satellite you need the value of two angles: one angle in the horizontal plane (the azimuth) and one angle in the vertical plane (the elevation) — not forgetting a compass, of course. The SatFinder uses its built-in satellite data base to calculate these angles using an ATmega8 microcontroller and information from a GPS receiver module.

A stand-alone solution

There are already a number of internet sites which give the exact azimuth and elevation positioning of a satellite dish. You just need to enter your location in the form of GPS coordinates, nearest town or post code and then choose a satellite from a pulldown list. The problem: Internet access is required.
The idea behind the project described in this article is to make a compact stand-alone device to achieve the same goal. The design uses a standard AVR microcontroller together with a 2x16 character LC display.

The intention was to write the software using BASCOM-AVR but then it was realised that a restriction of this particular implementation of BASIC is that it only allows one single mathematical operation per line. Calculating the values of azimuth and elevation requires a reasonable amount of data manipulation so there were fears that this may not be the best compiler to use. In the end it proved to be not too much of a hindrance so that all of the SatFinder firmware has been written in BASCOM-AVR.

The circuit

From the hardware point of view the project consists of an off the shelf GPS receiver module and an ATMEGA8 AVR microcontroller together with an LC-Display. A low cost GPS receiver module has been chosen for the design; it has an integrated antenna and outputs GPS positional data using a standard serial NMEA interface. The author used the Navilock NL-507TTL [1] unit for his prototype, this company produces a good range of receivers suitable for this application. Some of them however are fitted with a USB interface and these cannot be used with this design.

The circuit (**Figure 1**) and accompanying software are designed to cope with both TTL level and RS232 signals with a data rate switchable between 4800 bps and 9600 bps.
Irrespective of whether the signal from the GPS receiver arrives using RS232 or TTL signal levels the controller expects the data to conform to the NMEA protocol type 'RMC', which is the standard employed by the vast majority of GPS receivers.
A jumper fitted to S2 switches the displayed

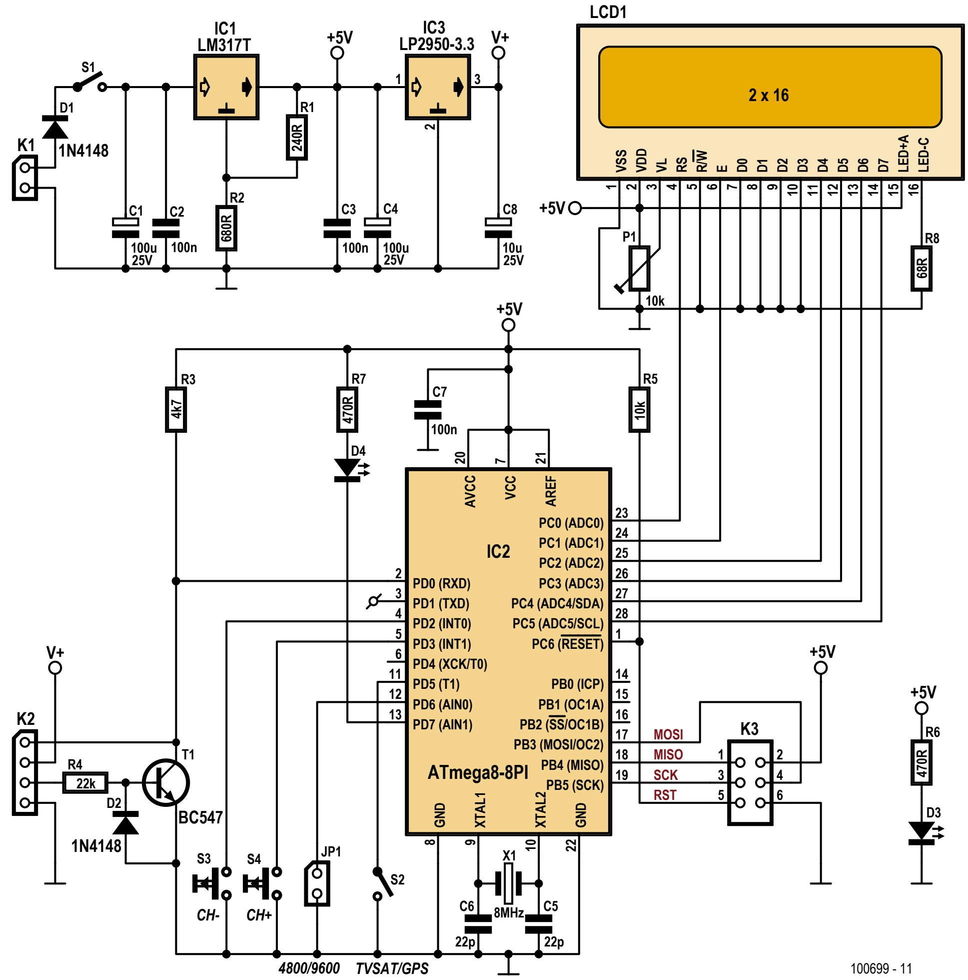


Figure 1. The SatFinder circuit diagram consists of a power supply, microcontroller and an LC display.
A GPS receiver module is connected to K2.

information, instead of showing the satellite dish set-up angles it displays the GPS coordinates of your current position. Jumper JP1 defines the communications data rate: with no jumper fitted it will be 4800 bps, with the jumper in place (forcing PD6 to ground)

Features

- TV dish azimuth and elevation angle display calculated from GPS data for 33 preinstalled TV satellites
- Display of GPS positioning data (geographical latitude and longitude)
- NMEA (RMC) input
- Input signal data rate switchable between 4800 and 9600 bps
- Input signal level switchable between RS232 and TTL
- Programming language: BASCOM-AVR
- All stored satellite data can be reprogrammed
- Source and hexcode files both free to download
- ISP interface for microcontroller programming
- Test pin for serial data output (TTL level, data rate same as serial input data)
- Operates from 12 V supply (external power from 8 to 15 V)
- Supplies 3.3 V or 5 V to the GPS module
- Supply current (no GPS module connected and without LCD backlight) approx 30 mA at 12 V

Elektor Products & Services

- Printed circuit board: # 100699-1
- Programmed ATMEGA8A-PU: # 100699-41
- Kit of parts, including programmed controller and PCB: # 100699-71
- Firmware and source code (free download): # 100699-11.zip
- PCB artwork: #100699-11.pdf
- Hyperlinks in article
- Items accessible through www.elektor.com/100699

COMPONENT LIST

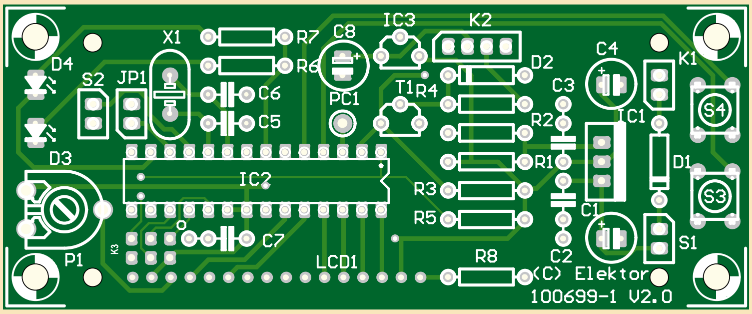


Figure 2. No SMDs are used on the double-sided PCB.

Resistors
R1 = 220Ω
R2 = 680Ω
R3 = 4.7kΩ
R4 = 22kΩ
R5 = 10kΩ
R6,R7 = 470Ω
R8 = 68Ω
P1 = 10kΩ trimpot (horizontal)

Capacitors
C1,C4 = 100µF 25V radial
C2,C3,C7 = 100nF
C5,C6 = 22pF
C8 = 10µF 25V radial

Semiconductors
D1,D2 = 1N4148
D3,D4 = LED, 3mm, low current
T1 = BC547B
IC1 = LM317T
IC2 = ATMEGA8A-PU, programmed, Elektor # 100699-41 [2]
IC3 = LM2950 3V3

Miscellaneous
JP1 = 2-pin pinheader, lead pitch 0.1 in. (2.54mm), with jumper
K1 = 2-pin pinheader, lead pitch 0.1 in. (2.54mm)
K2 = 3-pin pinheader, lead pitch 0.1 in. (2.54mm)
K3 = 6-pin (2x3) pinheader, lead pitch 0.1 in. (2.54mm)
S1,S2 = single pole switch
S3,S4 = pushbutton, 1 make contact, e.g. Multicomp # MC32830
X1 = 8MHz quartz crystal, e.g. Vishay Dale type XT49S-208M
LCD1 = 2x16 characters, e.g. Lumex S01602 D/A
PCB # 100699-1, see [2]

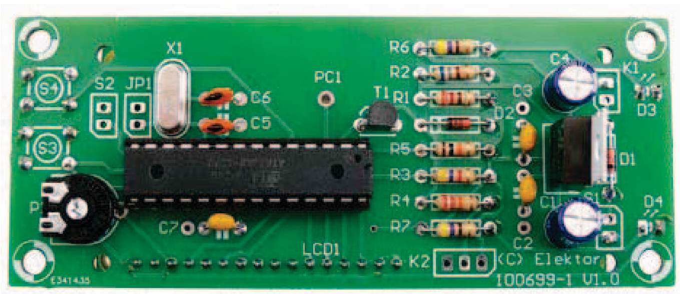


Figure 3. The populated Elektor prototype.

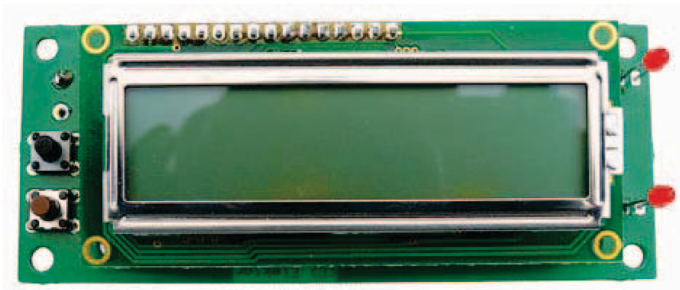


Figure 4. Two LEDs, two pushbuttons and the display are mounted on the underside of the PCB.

the rate will be set to 9600 bps.

Pushbuttons S3 and S4 (controller pins PD2 and PD3 respectively) allow the user to scroll through the list of satellites stored in memory. LED D4 (pin PD7) blinks when the controller detects a valid message from the GPS receiver and LED D3 indicates that a 5 V supply is available on the board.

Connector K3 is a 6-pin ISP connector for attachment of an AVR programmer.

For the purpose of debugging in the prototype unit the important navigation and angle information is also sent out from pin PD1 (TxD) of the microcontroller, this data output is not used in normal operation.

The only circuitry required to drive the LC display is a preset (P1) to adjust the display contrast and a fixed resistor (R8) to limit current to the backlight LEDs.

The SatFinder is designed to operate from a 12 V vehicle supply but it can cope with a supply anywhere in the range of 8 V to 15 V. The ATMEGA8 runs at 8 MHz which allows it to have a supply voltage between 2.7 V and 5.5 V. The LC display must operate at the same voltage as the microcontroller. The most common LCDs run at 5 V so the on-board voltage regulator IC1 (LM317T) has been set (via R1 and R2) to provide 5 V for the board. If a 3 V LCD is used the values of R1 and R2 can be changed to produce 3 V. Power to the GPS receiver module can be supplied over pin 2 of K2. The LP2950 voltage regulator (IC3) is used to provide 3.3 V for this purpose. If the GPS module however requires 5 V the LP2950 can simply be replaced by a wire link between its pins 1 and 3 (leave pin 2 open circuit) so that the positive lead of C4 connects to the positive lead of C8.

Figure 5. The lab prototype uses a GPS receiver from Trimble.

Assembling

A double-sided PCB was developed for this design (**Figure 2**). There are no special components or SMD packages used in the design so soldering components to the

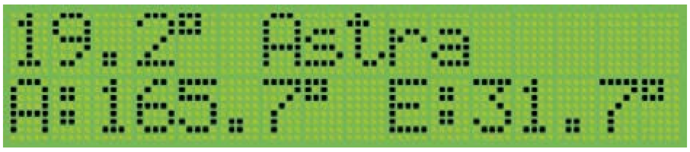


Figure 6. As soon as the GPS receiver sends positional data the azimuth and elevation of the chosen TV satellite is shown on the display.

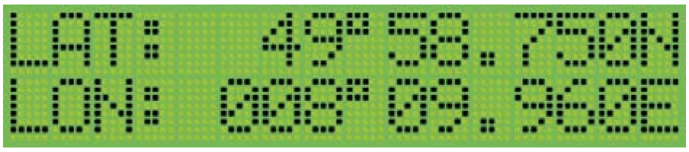


Figure 7. The display shows the GPS coordinates when switch S2 is closed.

board should be fairly trouble-free even for those inexperienced with the use of a soldering iron. The two push buttons S3 and S4, LEDs D3 and D4 together with the LC display are the only components mounted on the underside of the PCB. The display connections can be made using pin and socket strips. The display can then be secured to the PCB using M3 bolts together with suitable length non-conducting spacers.
Figure 3 shows the component side of the Elektor prototype PCB and **Figure 4** shows the underside with the LCD in position. The GPS module used for this example can be seen in **Figure 5**.

Putting it to use

Before the unit is powered up it is necessary to set the input data rate at the correct speed for the GPS data. For communication at 4800 bps leave port pin PD6 open circuit, for 9600 bps fit a jumper to JP1 so that PD6 is grounded. Now connect the GPS receiver cable to K2, the supply voltage is on pin 2 and ground on pin 4. If the GPS data is at RS232 levels then it must be connected to pin 3, if the data is at TTL levels then use pin 1. In this case it is not necessary to fit components T1, D2, R3 and R4. Make sure that RS232 signals are not directly connected to the microcontroller port pin otherwise it may cause damage.

Now with a 12 V supply connected to K1 the SatFinder is ready for use. The GPS module will only start sending data if it has an unobstructed view of sufficient satellites.

When the SatFinder is first switched on using S1 the display shows a copyright message and the software version in use. Try tweaking the display contrast control (P1) if there is nothing to see. The time taken for the GPS receiver to lock on to and synchronise with a sufficient number of satellites is dependant on the type of receiver module used. During this period the display shows 'WAITING FOR VALID GPS DATA'. Once the position has been determined the display changes to 'GPS-FIX'. The upper line on the display shows the position and name of the TV satellite that you have selected while the second line shows its actual azimuth and elevation from your current position (**Figure 6**). The program includes a table (**Table 1**) containing the positional data and abbreviated names of the most important TV satellites broadcasting to Europe with positions from around 50° east to 50° West of due South. A particular satellite can be selected using the plus/minus buttons (S3/S4). When the satellite selection is changed it will only be accepted on the read pulse following the change. This helps to reduce switch contact bounce. The positional information of the last satellite selected is retained even after a power down.
Switch S2 (shorting PORT input PD5 to ground) is used to switch the display so that it shows the current values of latitude and longitude supplied by the GPS receiver module (see **Figure 7**).

Preprogrammed TV satellites (European version)						
Express AM22	53.0° East	(Europe beam)		Sirius 4	4.8° East	
Intelsat 12	45.0° East	(Europe beam)		Eutelsat W1	4.0° East	
Turksat	42.0° East	(West/S1 beam)		Thor 3/5	0.8° West	
Hellassat 2	39.0° East	(F1/F2 beam)		Intelsat 1002	1.0° West	
Sesat 1	36.0° East	(Europe beam)		Amos 1/2	4.0° West	(Europe beam)
Eurobird 33	33.0° East			Atlantic Bird 3	5.0° West	
Astra 2C	31.5° East			Nilsat 901	7.0° West	
Turksat 1C	31.3° East			Atlantic Bird 2	8.0° West	
Eurobird 1	28.5° East			Atlantic Bird 1	12.5° West	(Europe beam)
Astra 2A/B/D	28.2° East			Telstar 1/2	15.0° West	(Europe beam)
Badr 4	26.0° East			Eutelsat W2	16.0° East	(Europe beam)
Astra 3	23.5° East			Intelsat 901	18.0° West	(Spot-1)
Astra	19.2° East			NSS 7	22.0° West	(Europe beam)
Hotbird	13.0° East			Hispasat 1C/1D	33.0° West	(Europe beam)
Eutelsat W1	10.0° East			Intelsat 3R	43.1° West	(Europe beam)
Eurobird 9A	9.0° East			Intelsat 1R	45.0° West	(Europe/North-Africa
Eutelsat W3	7.0° East					

The LED (D4) connected to port pin PD7 flashes approximately once a second indicating that a message conforming to NMEA protocol has been successfully received and processed. Note that this occurs even if 'GPS-FIX' has not yet been achieved.

Software

The SartFinder software receives positional data sent from GPS receiver at approximately one second intervals. This information conforms to the NMEA type RMC protocol standard. Once the data has been extracted and checked for errors it is used in conjunction with the stored orbit position of the selected TV satellite to produce and display the azimuth and elevation angle of the TV dish. The display is refreshed approximately every second.

Satellite selection from the list is made with the help of interrupt controlled counters and use of the 'Plus' and 'Minus' buttons. As already mentioned a new satellite selection will only be read on the following read pulse to help mask switch bounce.

A pre-programmed ATmega8 microcontroller is available from the Elektor Shop. Those of you have an AVR programming device and prefer to do it yourself can find the necessary software files on the Elektor website [2], the hex and source files are both available. Elektor USA readers require file # 100699-12.zip which is based on about 40 satellites for the region.

Modify to suit your needs

The firmware source code is available from the Elektor web site so you are free to modify it at your leisure. The list of satellites stored in the code covers larger Europe but these can be easily changed to make the design suitable for other regions (e.g. North America). For any of the changes to take effect it will of course be necessary to recompile the source code using the BASCOM-AVR compiler.

Information giving the positions of all the TV satellites currently orbiting the globe can be found at [3].

(100699)

The Author

After many years working in the communications and electronics industry Klaus (ham callsign: DJ700) now has time to pursue his interests in the fields of amateur radio and microcontrollers where he has already made many useful and significant contributions.

Internet Links and Sources

[1] www.amazon.co.uk/s/ref=nb_sb_noss?url=search-alias%3Delectronics&field-keywords=navilock&x=0&y=0

[2] www.elektor.com/100699

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

[4] www.spaceacademy.net.au/watch/track/locgsat.htm

[5] www.angelfire.com/trek/ismail/theory.html

[6] Dennis Mitchell, Receiving signals from Space, *Ham Radio Magazine*, November 1984, pp. 67-69.

[7] Paul Shuch, Calculating Antenna Bearings for Geostationary Satellites, *Ham Radio Magazine*, May 1978, pp. 67-69.

MP3 One-Two-Three

A software project for the TMS320C5515 starter kit

By Lars Lotzenburger (Germany)

Would you like to develop your own MP3 player? The example application presented here describes the essential components and parameters and shows that nowadays anyone with a bit of programming skill can develop a player tailored to their personal wishes. Free project software makes getting started nearly effortless.

Real-time decoding of audio data streams encoded in MP3 format has long since become standard practice, so current MP3 players now have to rely on other features to set themselves apart from the competition. For instance, they may emphasise ease of use, style, or the number of compression formats they support. Another decisive aspect is power consumption, since these small players are primarily designed for mobile use. Low power consumption allows the device to operate from a smaller battery or provide longer playback time. Nevertheless, the player must provide the processing power necessary to decode the compressed data stream in real time. Low-power digital signal processors, such as the members of the ultra low-power TMS320C55x family from Texas Instruments, are very suitable for this sort of application.

Low-power DSP

The latest member of this family is the TMS320C5515 signal processor [1] used in this project. The DSP core features an especially low power consumption of approximately 11 mW at 75 MHz or 27 mW at 120 MHz. As can be seen from a glance at the integrated peripheral units (See **Figure 1**), the designers of this IC had audio decoding in mind as one of its possible applications. For example, it directly supports access to MMC and SD cards. A High Speed USB port for connection to a computer and a real-time clock are also on board, as well as support for a liquid crystal or OLED display and several I^2S audio ports.

Starter kit

To simplify getting started with application development using the TMS320C5515, Spectrum Digital Inc. [2] has put together a starter kit called “TMS320C5515 eZDSP USB Stick” [3]. The name is perhaps a bit misleading, since the actual device is a PCB measuring approximately 70 × 70 mm (**Figures 2** and **3**). This board holds all the essential components necessary for a modern MP3 player: a USB port, an MMC/SD card connector, an OLED display (monochrome), pushbuttons, an audio DAC with headphone output, and a Line input. The board is powered from the USB port. The program code can be stored in NOR flash memory to enable the board to operate stand-alone. During the development phase, the integrated XDS100 emulator serves as the communication interface between the development environment on the PC and the circuitry on the board.
The starter kit [4] includes a mini-CD holding the development environment software, which is Code Composer Studio V.4 with an IDE based on Eclipse. The DSP/BIOS real-time operating system used in this MP3 player project is also included in the installation.

Software

As we all know, a processor is only as good as the software available for it. Nowadays the bulk of the effort goes into developing software for embedded systems. Here the

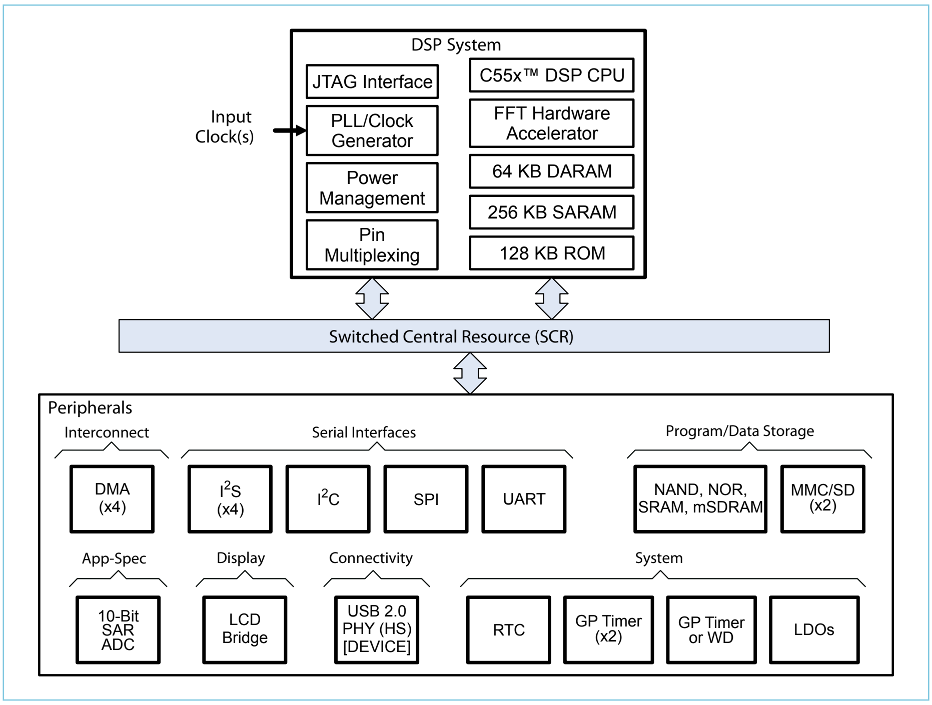


Figure 1. Block diagram of the TMS320C5515 digital signal processor IC.

most effective approach is to reuse as many existing software components as possible in order to minimise application development time.

Among other things, developing a good MP3 algorithm is far from trivial. Although numerous software projects are available on the Web and can be ported to any desired processor with the aid of a C compiler, it's still necessary to invest a lot of time in optimising the code for the target processor in order to take advantage of its strengths, even if the MP3 algorithm is already optimised. Otherwise it's not possible to minimise the algorithm's resource usage. Ultimately, optimisation results in a lower clock frequency and therefore lower power consumption.

MP3 project

Texas Instruments offers a lot of support for developers. The MP3 algorithm can be downloaded free of charge in the form of a software library [5] and simply be bound into the software project. The application code can use a standardised API to call the functions in the library. An MP3 data stream encoded at 128 kbps needs no more than 20 million clock cycles per second at maximum amplitude.

Of course, the MP3 algorithm also needs to communicate with a data source and a data sink. Here the data source is an SD card, which is driven by the integrated MMC/SD peripheral unit. Existing software can be utilised here as well. The Chip Support Library (CSL) [6] from Texas Instruments provides software drivers for all integrated peripheral units in the DSP. It comes complete with full source code, examples, and a precompiled library. This library is bound into the software project in the same way as the decoder library.

At this point, it is essentially possible to read data from the SD card using the CSL and provide it to the MP3 algorithm for decoding. However, we're still missing some key information: the location of the data on the SD card. The data is stored on the card in

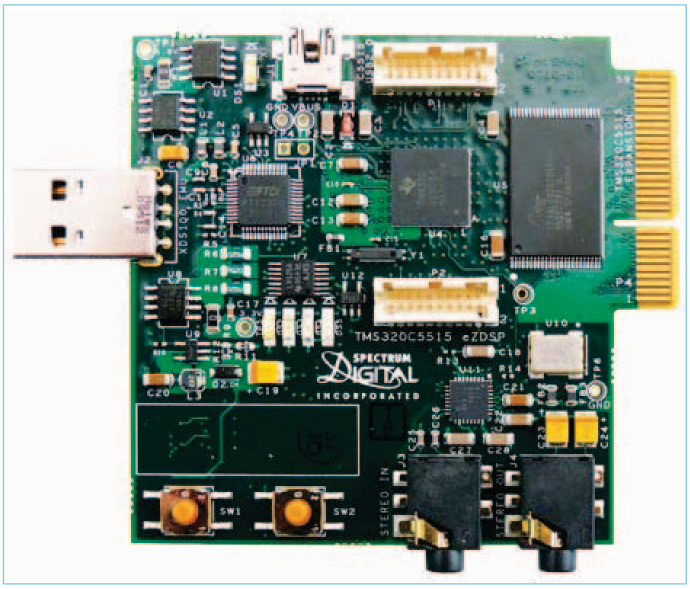


Figure 2. The starter kit board is fitted with all the hardware needed for an MP3 player.

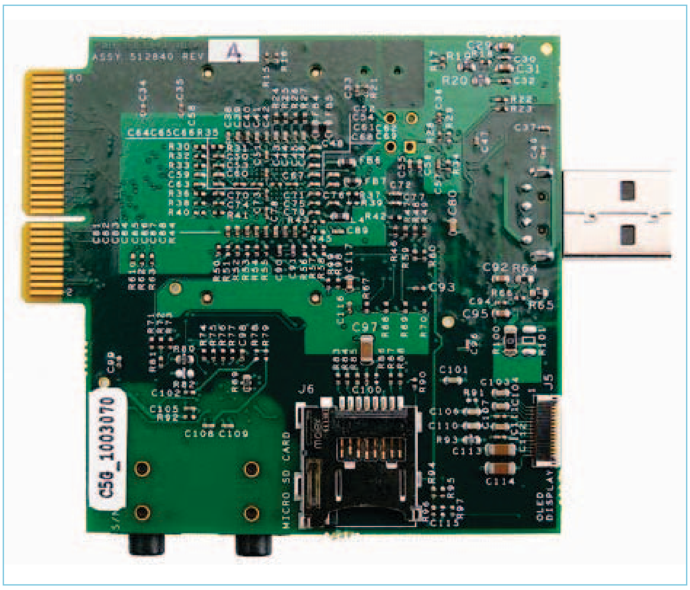


Figure 3. The MMC/SD card connector is located on the rear of the board.

a file system composed of directories and files. The commonly used file systems are File Allocation Table (FAT16 or FAT32) and New Technology File System (NTFS). A file system handler controlled by the application must be integrated between the SD card driver and the decoder to enable access to the desired data. Here again it is not necessary to reinvent the wheel for this function, since an abundance of software projects are available on the Web. We used a FAT FS handler package called *FATFs* [7] for our MP3 player project. This software is written in C, and in theory it can run on any processor and work with any mass storage medium (hard disk, CD-ROM or SD card). The code comes with stubs (empty functions) to provide the necessary flexibility. They must be completed with program code to drive the corresponding hardware. In our case, this consists of the CSL functions for driving the SD card. On the application side, *FATFs* provides a set of high-level functions such as *Read File 'Test.mp3'*. With the aid of file system parameters, *FATFs* translates this function call into addresses and uses the CSL functions to initiate a read operation on the SD card. The application then passes the received MP3 data to the MP3 algorithm. This ensures that the SD card sends the data desired by the application to the MP3 algorithm.

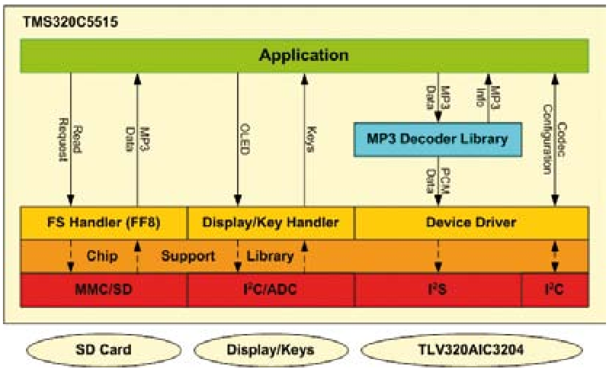


Figure 4. Block diagram of the software components of the project.

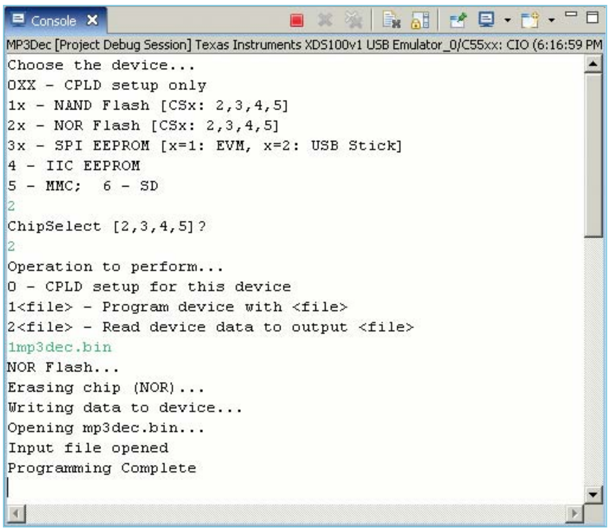


Figure 5. Writing the NOR flash memory for stand-alone operation.

After the MP3 algorithm has decoded an MP3 frame, there are 2×576 16-bit audio values (with a stereo signal) available in an audio buffer for further processing. This data can be sent via the I²S interface directly to the data sink, which in this case is a TLV320AIC3204 audio DAC. Here the application uses a device driver for all communication with the audio DAC. This allows a low-level driver (also known as a mini-driver) to be used to simplify swapping the data buffers on the I²S interface. When a data buffer has been consumed, which means that its data has been sent over the I²S interface to the audio DAC, the application reclaims the buffer and the mini-driver can issue a new audio buffer filled with data. The objective of this is to decouple the application from the driver software of the TLV320AIC3204 by using buffers for all data communication. **Figure 4** shows a block diagram of the software components.

Post-decoding modification of the audio signal is often desirable. A library is also available for this purpose: the TMS320C55x DSP Library [8]. It contains functions for many commonly used signal processing algorithms, such as (adaptive) digital filtering, FFT, and standard mathematical functions. If you only want to change the volume or make a simple adjustment to the frequency spectrum, such as bass boost, the TLV320AIC3204 can do the job. It has integrated units for IIR and FIR filters, which offload these tasks from the DSP core. The audio DAC also has an integrated PLL to generate the sampling rate clock for the audio signal. Commonly used values are 44.1 kHz and 48 kHz. The audio DAC uses this clock to generate the bit clock and frame clock signals for the I²S interface. This allows the DSP to operate at a clock rate that is independent of the sampling rate.
The sampling rate is known after the first frame of the MP3 signal has been decoded, and it is adjusted accordingly via the I²C control interface of the audio DAC. The same I²C interface is used to control the OLED display. The two pushbuttons are read via the analogue input of the DSP IC's internal A/D converter instead of GPIO ports. The two buttons are connected in parallel so that only one input pin is needed to detect the actuation of either one of them.

Downloads and instructions

The software project can be downloaded from the Elektor website page for this article [9]. Detailed instructions for installing the development environment software, binding and running the MP3 software, and configuring and operating the MP3 player are also available on the same page.
After compilation, the software should be written directly to the NOR flash memory (see **Figure 5**). The program searches the root directory of the SD card for MP3 files and plays them one after the other. The name of the file currently being played is shown on the OLED display. The volume can be adjusted with the two buttons.
In addition to the MP3 library, software libraries for decoding files in Windows

Media Audio 9 (WMA9) and Advanced Audio Coding (AAC) formats are available. The standardised API common to all audio decoders makes it easy to swap algorithms. This could conceivably be used as the basis for a multi-format player.

Audio recording, with the audio data being encoded in real time and written to the SD card, is also potentially possible. A library for encoding audio signals in AAC format is available. As the TLV320AIC3204 is a codec (coder/decoder) device, digitising other types of analogue input signals would also be fairly straightforward. For example, you could implement a dictation recorder without any hardware modifications. Developers can also connect their own hardware to the expansion ports of the board.

We hope you have a lot of fun developing your own personal MP3 player!

(100822-I)

[1] **TMS320C5515 website**
http://focus.ti.com/docs/prod/folders/print/tms320c5515.html

[2] **Spectrum Digital, Inc. website**
www.spectrumdigital.com

[3] **TMS320C5515 eZDSP USB Stick website**
http://support.spectrumdigital.com/boards/usbstk5515/reva/

[4] **TMS320C5515 eZDSP USB Stick source:**
http://focus.ti.com/docs/toolsw/folders/print/tmdx-5515ezdsp.html

[5] **Codecs website**
http://focus.ti.com/docs/toolsw/folders/print/c55xcodecs.html

[6] **TMS320C55x Chip Support Library (CSL) website**
http://focus.ti.com/docs/toolsw/folders/print/sprc133.html

[7] **FATFs website**
http://elm-chan.org/fsw/ff/00index_e.html

[8] **TMS320C55x DSP Library website**
http://focus.ti.com/docs/toolsw/folders/print/sprc100.html

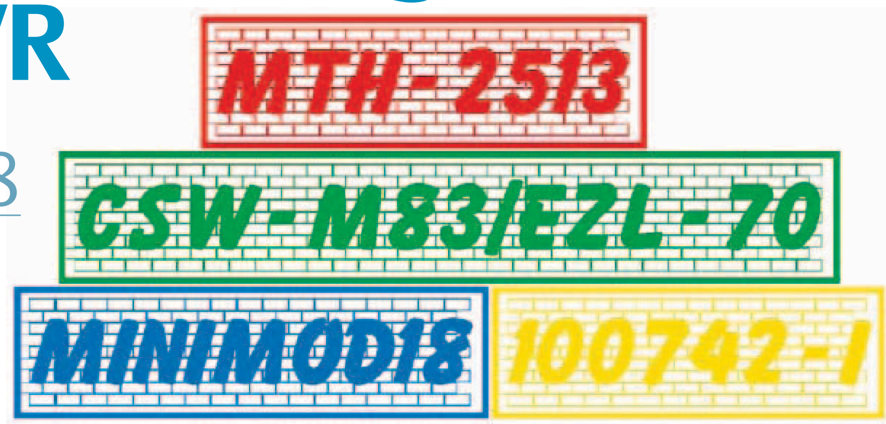
[9] **Elektor article website**
www.elektor.com/100822

About the author

Lars Lotzenburger is a system engineer at Texas Instruments Germany, located in the town of Freising near Munich.

Mini Webserver using BASCOM-AVR

Elektor Minimod18 at your service...



By Grégory Ester (France)

In this article two old faithfuls together form an exciting project. For hardware we have Elektor's very own Minimod18 and as the client there's the Firefox browser, which can be relied on to handle your very own HTML data faultlessly. New to the show is the EZL-70 module for all Ethernet data processing. Are you ready? Then it's time for "curtain up!" on this project...

Remote printing, driving a 7-segment display, reading the logic states of several inputs, controlling binary outputs, displaying the value of an analogue voltage, preparing a shopping list before merrily going off to the supermarket, writing some tender words to one's girl- or boyfriend — what's that got to do with anything, you say?
Yet there is indeed a common point in all that: an Ethernet/Serial interface!

So today we're going to take a look at using this converter to implement an embedded-technology webserver. The Elektor Minimod18 gets kitted out with network connectivity that lets it join the immense Internet cloud!

The Internet interface and its configuration

The EZL-70 (or EZL-70A with reduced power consumption) (**Figure 1**) and CSW-M83 (for the Wi-Fi version) modules are Ethernet/Serial converters marketed by the Korean company Sollae Systems [1] and distributed in the UK by Equinox [2]. In this application we're going to be using the T2S (TCP to Serial) mode offered by these modules, which lets us exchange data between a serial port and a TCP/IP network.

To route the data over the network, the TCP/IP protocol associates each network access point with an IP address. These addresses must be chosen in accordance with the addressing plan of your LAN (local area network). Let's assume that the sub-network mask can only take the values 0 or 255 and that we choose to assign each piece of equipment with a private, static Class C IPv4 address. For the PC to be able to communicate with the Ethernet module, it is vital that the result of the logical 'AND' between the PC's address and the mask should be identical to the 'AND' result between the Ethernet module address and the mask. In effect, this result corresponds to the logical sub-network.

Minimod18

This article uses the Minimod18 (Elektor # 090773-91) published in Elektor, April 2010 edition. It's a very compact universal microcontroller module offering the most commonly used peripherals like keys, display, USB, I²C, and ISP/SPI interfaces.

www.elektor.com/090773

So we're going to choose 192.168.1.55 for the module and 192.168.1.56 for the PC with a mask 255.255.255.0. In this way, both pieces of equipment are on the same sub-network 192.168.1.0.
If the DHCP service (dynamic IP address assignment) is enabled on your (broadband) router, choose an address that's outside the range of reserved addresses.
It's now possible to transport packets of data. In order to do this in a way that is transparent for the user, TCP/IP will first check that the destination host can be contacted and is ready to receive the data. This link is established and checked throughout the communication. Thus the data are exchanged in a reliable fashion by way of a mechanism for acknowledging received packets and having them re-sent if errors are detected.

To fully identify the connection, we need to assign the module a port number; here we've chosen the number '49500'. The socket address (i.e. the virtual socket through which the data will be passing) is thus 192.168.1.55:49500.
Figure 2 shows which fields to fill in so as to configure the EZL-70 module with the help of the ezConfig [1] utility. Remember to set Timeout — the value '0' inhibits it. Here, it's set to 5 s. If the client fails to close the connection of its own accord, after 5 s the module will do it, thereby making it available once again.
A small, basic ping (straight from command prompt) from the PC to the module will let you quickly check that the participants are able to communicate with each other:

```
ping 192.168.1.55
```

Communicating in HTML by HTTP

We now have all the ingredients to be in a position to access the Ethernet/Serial module via a browser — Firefox for example — by typing the full address of the page to be displayed like this: http://192.168.1.55:49500/texto.html (or another page, see **Table 1**) and then confirming. The prefix 'http' means that the communication will be taking place using the HTTP protocol, which allows data to be transferred in the HTML format — the universal language of the Internet.

On the network side, you send an HTTP request to our HTTP server, the EZL-70 module, and it will send back a response (a file or a frame) in HTML. On the serial side of the same module, we are going to be able to view the contents of the requests by using HyperTerminal, for example. The first line we'll see is:

```
GET /texto.html HTTP/1.1
```

Which means that the request is of the type 'GET' and the resource requested by the client is the file 'texto.html', and it's version 1.1 of the HTTP protocol which the client is using. Other information can be seen, like the language (Accept-Language) and the character set (Accept-Charset) expected by the browser, but as it happens,

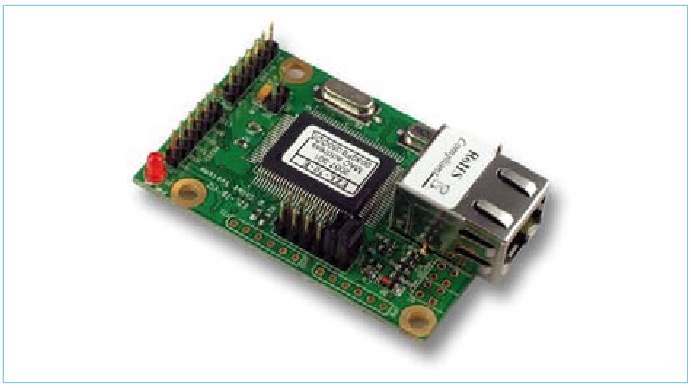


Figure 1. EZL-70 module Ethernet/Serial converter.

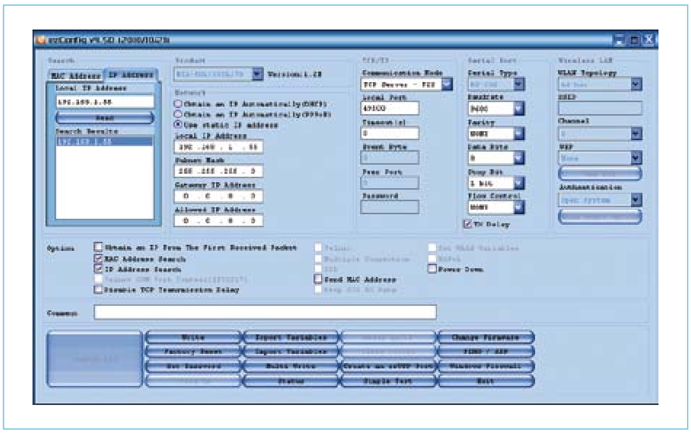


Figure 2. ezConfig for 'easy' configuration of the Sollae modules.

only the first line will be interpreted by the program written in BASCOM-AVR.

```
Char = Inkey(#2)
'--- if first chars = 'GET' then get File name
If Char = "G" Then
  Do
    Index1 = Index1 + 1
    Get #2 , Request(index1)
  Loop Until Request(index1) = Chr(13)
  Length = Index1 - 10
  If Request(1) = "E" And Request(2) = "T" Then
    File_name = Request(5) + Request(6) +
Request(7) + Request(8) + Request(9)
  End If
End If
...
If File_name = "texto" Then
  <send header + body of page>
End if
...
```

If the request is of the type 'GET' and 'texto' is the name of the file requested, we'll be able to hoodwink our browser by sending it a header containing in particular the number of characters that will be sent in the body (Content-Length) along with the field 'Connection:

Figure 3. A test page served up by our mini-server.

Close' so that the browser terminates the connection once it has received what it wanted, followed then by some HTML instructions (group of characters) which will be interpreted by the browser. The browser won't turn a hair, but will comply by displaying the corresponding page (**Figure 3**).

All this is fairly simple if the server only offers static HTML pages, since in this case all it has to do is send a page stored in advance in the server's memory. If, on the other hand, we want a server that responds to commands, things get a bit trickier, and it will be necessary to create HTML pages 'on the fly'. Let's take a look, for example, at the request when we enter some text to be printed and then click on the 'ILLICO TEXTO' button:

```
GET /texto.html?mth_mess=Hello+world HTTP/1.1
```

If the command is of the type 'GET' and 'texto' is the filename, then all we have to do is recover the contents of the text to be sent to our printer, "Hello+world". It's worth noting here that special characters like &{]()'~"#@|%,;:<> etc. are processed and display correctly on the thermal printer.
The server responds with an updated page, constructed on the fly. Each time it is sent, the text box background changes colour from blue to pink and back again, thus indicating that the message has been received OK:

```
Toggle Flag4
If Flag4 = 1 Then Background = "bg1"
If Flag4 = 0 Then Background = "bg2"
...
Print #1, "<style text="; Chr(&H22); "text/
css"; Chr(&H22); ">"
Print #1, ".bg1{BACKGROUND-COLOR:#a9eeec;}"
Print #1, ".bg2{BACKGROUND-COLOR:#ffcbec;}"
Print #1, "</style>"
```

This chunk of code sends the following HTML frame (if Flag4 = 1):

```
<style text="text/css">.
bg1{BACKGROUND-COLOR:#a9eeec;}</style>
```

To create the page's text entry box, we use the following HTML command for a blue background (the parameter 'class' is thus alternately 'bg1' and 'bg2'):

```
<input type="text" name="mth_
mess" size="40" maxlength="40" class="bg1">
```

Which in BASCOM-AVR gives:

```
Print #1, "<input type="; Chr(&H22); "text";
Chr(&H22); " name="; Chr(&H22); "mth_
mess"; Chr(&H22); " size="; Chr(&H22); "40";
Chr(&H22); " maxlength="; Chr(&H22); "40";
Chr(&H22); " class="; Chr(&H22); Background;
Chr(&H22); ">"
```

The instructions `Chr(&H22)` that make this fragment of code difficult to read are used to send the " character — not to be confused with the " characters (yes, the same!) which are used simply to frame the text for the 'Print' instruction and aren't actually sent. Are you still with me?
The page title, enclosed by the HTML tags <title> and </title>, is also filled in with here the name "MINISERVER: TEXTO":

```
Print #1, "<title>MINISERVER: TEXTO</title>"
```

Now let's add a serial thermal printer

Thermal printers use special paper that reacts to heat, with a sort of 'comb' that is heated to produce dots on the paper. These can be found on the Internet in the form of a small module with an easy-to-drive serial interface. We chose the MTH2513 from Megatron [3], but there's nothing to stop you choosing another printer (adapting the commands, of course). The case/head/electronic interface of our wonderful printer is about as big as a large box of matches, and is very sturdy. There are very many functions, which are well described in a 40-page booklet.

Figure 4. Who said you shouldn't judge by first impressions?

After connecting pin K1(7) on the Minimod18 to RXD on the printer and powering everything up, all you have to do is open the lid, fit the roll of paper, close the lid again, and load the program ELEKTOR_LOGO_CODE39_MTH2513.hex into our ATmega328's flash memory. Once you've run it, a little curl of paper (**Figure 4**) will tell you that everything has gone according to plan.

Once again, we have used The Dot Factory [4][5] software to break our logo down into bytes. Once these bytes are arranged in a table, all you have to do is send them to the printer in the right order, line by line — four bytes per line in our example — using the command:

```
<Esc> K n octet[1] … octet[n]
```

In BASCOM-AVR:

```
For X = 0 To 255 Step 4
  Print #3, Chr(27); "K"; Chr(4);
Chr(logo(x+1)); Chr(logo(x+2)); Chr(logo(x+3));
Chr(logo(x+4));
Next X
```

The barcode in Code 39 format is generated in the following way:

```
Print #3, Chr(27); Chr(&H22); Chr(&H32);
Chr(&H31)  'barcode enlargement factor = 1
(largest)
Print #3, Chr(27); Chr(&H22); Chr(&H31);
Chr(&H04)  'code 39
Print #3, "CODE 39"
Print #3, Chr(27); Chr(&H22); Chr(&H30); "E";
"L"; "E"; "K"; "T"; "O"; "R"; Chr(255)
```

At this stage, our printer is ready to be incorporated into the system.

The microsoftware

Once compiled and loaded into the flash memory of the Minimod18 connected to your local network, the example programs written in BASCOM-AVR (available from [6]) will allow simulation of the same responses as an HTTP server to a request made by a web browser. The browser will then display the corresponding HTML page — as long as you have of course loaded the correct software and connected the whole thing up as shown in **Table 1**.

- 8574_INPUT_AND_ADC:
  reads the state of the binary inputs on the tutorial board (JP8–JP15) and the value of the voltage present on ADC6 (page refreshes automatically every 4 s). Above 2 V, the voltage is displayed on a red background.
- 8574_OUTPUT:
  Drives the seven segments and decimal point of the display on the tutorial board with the logic state of each output displayed on the HTML page. The 8-bit binary number corresponding to the state of the segments is also displayed on the second line of

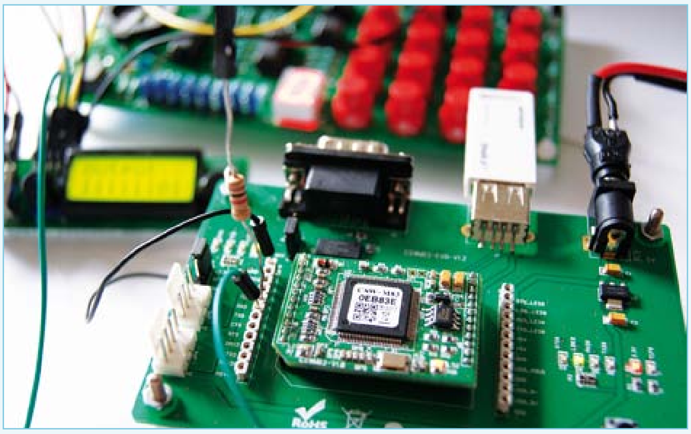
Table 1. The correspondence between the HTML pages, the software, and the interconnections between the various modules. All the programs have been successfully tested with the Firefox 3.6.12 and Internet Explorer 8.0.6001.18702 browsers.

Address	Software	Minimod18	EZL-70 CSW-M83	Tutorial board	MTH-2513	Sounder
input.html	8574_INPUT_AND_ADC.bas	K1(5)	TXD	-	-	-
output.html	8574_OUTPUT.bas	K1(6)	RXD	-	-	-
		K1(7)	-	SDA	-	-
		K1(8)	-	SCL	-	-
texto.html	ILLICO_TEXTO.bas	K1(5)	TXD	-	-	--
		K1(6)	RXD	-	-	-
		K1(7)	-	-	RXD	-
		K1(8)	-	-	-	(+)
shopping.html	SHOPPING_xx.bas	K1(5)	TXD	-	-	-
		K1(6)	RXD	-	-	-
		K1(7)	-	-	RXD	-
–	ELEKTOR_LOGO_CODE39_MTH2513.bas	K1(7)	-	-	RXD	-

The CSW-M83 module

A Wi-Fi USB key, a CSW-M83 test/mounting board, the CSW-M83 module itself, plus a 1 kΩ resistor = total independence! [1] Configuration is similar to that for the EZL-70 module, made possible by the RS-232 serial port on the mounting board, using ezTCP Manager and with jumper JP3 set to ON. You'll also need to associate the board to your router manually or automatically, in the same way you do for a computer with a Wi-Fi connection.

The ISP# link will remain constantly tied to 3.3 V via jumper JP2. It's important to point out that the serial communication from the CSW-M83 module is at 3.3 V. As a result, it is necessary to insert a 1 kΩ resistor between K1(6) on the Minimod18 and Rx on the CSW-M83.



the Minimod18's LCD.

- ILLICO_TEXTO:
  from your browser, you have the possibility of entering a text of 40 characters maximum; clicking the 'ILLICO TEXTO' button will send this message via the network to the thermal printer.
- SHOPPING_xx (where xx is FR or UK):
  from anywhere in the world, connect to your Minimod18 Mini-server back at home, prepare your shopping list by checking the products to go into your shopping trolley. With a bit of luck, your partner will find the list before you get home!

Port Forwarding — what's that?

The Sollae modules are configured with a private local IP address.

Nom	Activé	Protocole	Du port	Au port	Adresse IP locale	Action
serveur_thonon	Oui	TCP	80	80	192.168.1.54	
serveur_thonon	Oui	UDP	80	80	192.168.1.54	
minimod18	Oui	UDP	49500	49500	192.168.1.55	
minimod18	Oui	TCP	49500	49500	192.168.1.55	
Nouvelle Entrée						

Figure 5. Welcome home!
(example using a French Sagem Livebox).

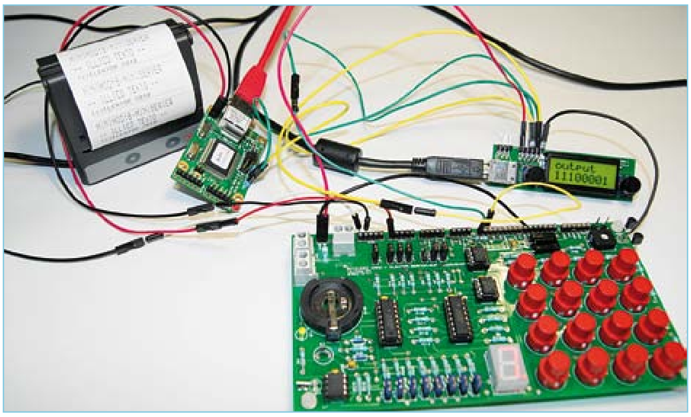


Figure 6. A few modules and some wires... hard to believe that this is a real multi-purpose webserver.

Your router has a LAN address configured in our case statically in the same addressing plane, on the same physical and logical network as your PC and the Ethernet/Serial converter. On the WAN side (towards your Internet access provider), the public address visible 'from outside' is rarely static and changes automatically and regularly. The router normally displays this address on the status page, but it can also be obtained by connecting, for example, to the www.monip.org website.

Let's assume that your public IP address is 80.197.119.229; so in order to be able to access the Minimod18 Miniserver from the Internet — to print out your shopping list, for example — you'll need to enter the following address into a browser:

http://83.197.119.229:49500/shopping.html.

The port forwarding will be carried out transparently as long as your modem/router has been configured as shown in **Figure 5**.

If you don't want to have to find out your IP address every time to connect to your projects, you can make free use of the services of a dynamic domain name server (DNS) like DynDNS. You will then obtain a domain name of the style myhome.dyndns.org that you can type into your favourite browser in place of your public IP address. Some routers have a built-in DynDNS client, as is the case with the Livebox. As a result, once configured, the connection between the domain name and the IP address will be established automatically.

(100815)

Internet links

[1] www.eztcp.com/en/home/

[2] www.equinox-tech.com/products/manufacturer.asp?ID=43

[3] www.megatron.fr/imprimantes/mpanel/mth2500_f.php

[4] www.pavius.net

[5] www.elektor.com/100256

[6] www.elektor.com/100815

PSoC Enables Custom Glass LCDs

with none of the fuss

By Robert Jania, Product Marketing Manager, Cypress Semiconductor Corp. (USA)

Cypress PSoC 3 and PSoC 5 devices surround industry standard microcontroller cores with high-performance programmable analog and digital. The new PSoC Creator integrated development environment enables schematic capture design. When these two products are used together, designers experience a whole new level of freedom.

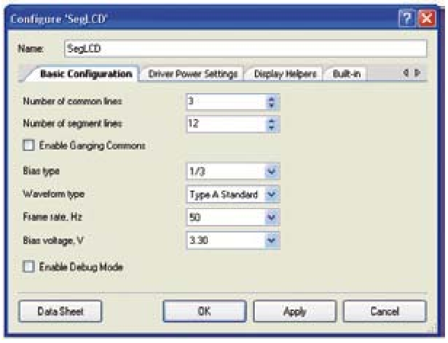


Figure 1. PSoC Creator Configuration Window.

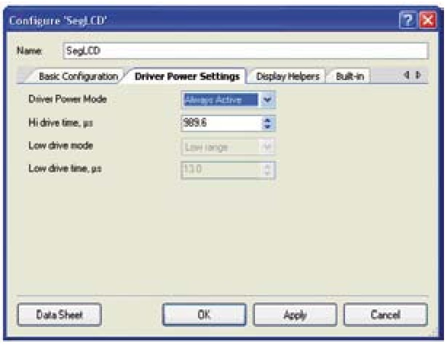


Figure 2. Driver Power Settings.

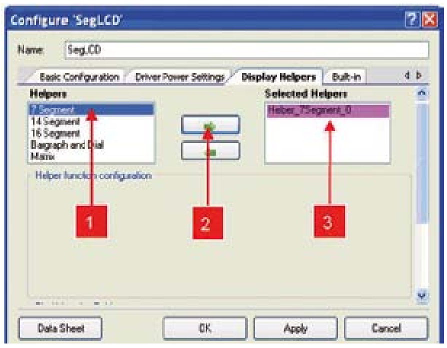


Figure 3. Display Helpers.

Instead of selecting how many timers, PWMs, and UARTs you need, PSoC (Programmable System on a Chip) allows you to select devices with different amounts of programmable fabric.

Your silicon is no longer limited to perform one task — instead it can be dynamically configured, allowing you to avoid wasted die space. No longer do you need to purchase a part that has more peripherals than your design requires.

Your very² own LCD

What makes PSoC Creator really unique is that it doesn't just give you standard peripherals, but it also offers you drag and drop functions. Let's take segment LCD direct drive for an example. Typically a microcontroller will use its I/Os to create the voltage waveforms for the LCD's common and segment lines. To control these waveforms one has to write countless lines of code, lookup tables and custom functions so that your design works with your LCD display. All of this takes away time which you could have been using to really differentiate your product from the competition. PSoC Creator solves the problem for you. Let's look at how it's done.

DIY at its best

PSoC Creator offers many preconfigured, pre-characterized components that can reduce your development time. LCD segment drive is one such example. The tool's component allows for you to select the number of commons and segments, select type of waveform that's required, set the frame rate or refresh rate, select the bias voltage or set up contrast level, set drive power level and select the display helpers. The first step is to drag and drop the **Segment LCD** component from the component catalog inside the PSoC Creator tool. Double-clicking on the component opens its configuration window. There are three tabs of interest, Basic Configuration, Driver Power Settings, and Display Helpers. The Built-in tab is for advanced functions and the default parameters don't need to be changed for this example. With the settings discussed and a small piece of code, you can build, run and test the project.

The parameters for the Basic Configuration tab are shown in **Figure 1**. They include the following:

Number of common lines:
This setting is depending on the glass used. For this example, set it to '3'.

Number of segment lines:
This parameter is also glass depending. For this example you will set it to '12'. Your number of common lines multiplied by your number of segment line equals the maximum number of pixels your LC display

Solve this month's Elektor Hexadoku and win a Cypress PSoC 5 First Touch Starter kit

For sure, it's possible to make the PSoC-designed LCD discussed in this article display letters A through F besides the usual numbers 0 through 9. Even if you do not foresee any custom LCD designing, we're still talking hexadecimal. So what's keeping you from solving this month's extra challenging Hexadoku Digest puzzle and enter a prize draw for one of 10 Cypress PSoC 5 FirstTouch™ Starter kits (CY8CKIT-014) worth € 45.00/ £40.00 / $40.00 each (approx.) jointly offered by Elektor and Cypress?

The new kit enables designers to get acquainted with the new PSoC 5 architecture. Users will also have the opportunity to use Cypress' software, PSoC Creator IDE, a powerful design environment that combines schematic and textual entry with pre-configured, pre-tested components that can be simply 'dropped-into' designs.

The kit also includes pre-loaded example projects which take advantage of the kit's onboard sensors including an accelerometer, a thermistor, proximity sensing, and CapSense. The user is given a practical tutorial of how the PSoC® 5 FirstTouch™ Starter Kit works by following a range of hands-on projects, and then by road-testing their own prototyping ideas.

The kit enables easy development via 28 general-purpose I/O pins, a 12-pin wireless module header, and Serial Wire Debugging (SWD).

Hexadoku was designed by Elektor and is now in its fifth year of publication. Thousands of Elektor readers from across the globe have sent in their solutions over the years. Turn to page $4B_h$ and get cracking!

could support. For this example, the maximum is 36, however our glass only has 28.
Enable ganging commons:
This is useful when driving a glass with very large segments or when the capacitance offered by commons is greater than 5,000 pF. For this example, leave it un-checked.

Bias type:
This parameter is read-only and is set by the tool depending on the number of common lines

Waveform type:
This affects the current consumption, for this example you will use the default setting.

Frame rate:
This parameter determines how many times each segment is refreshed in a second. For this example we will set it to 50 Hz. This means that each of the 28 LCD pixels are refreshed 50 times per second.

Bias voltage:
This parameter controls the contrast of the display. We will set it to 3.3 V.

The parameters for the Driver Power Settings tab are shown in **Figure 2**. They include the following:

Driver Power Mode:
There are two power levels in which the LCD driver can operate: Hi Drive (more power) and Low Drive (less power). Using these two power levels, the following options are provided:

- Always Active: The LCD driver is in Hi Drive mode for the time specified in the configure window. By default, it is set to the minimum value according to the frame rate, number of commons, and type of waveform selected. The LCD driver is in Low Drive mode for the remaining time of the refresh period.
- Low Power: In this mode, you have the option to select the Hi Drive time and the Low Drive time. In the beginning of the refresh period, the LCD driver is kept in Hi Drive mode; after the specified time interval (Hi drive time), the LCD driver is moved to the Low-Drive mode. After Low drive time, all LCD pins are tristated and the LCD driver is shut down. This offers significant power savings when compared to 'Always Active' mode. It is useful for battery operated applications.

For this example we will select Always Active mode with a Hi drive time of 989.6 µs. The last two settings, which are related to low power, remain disabled.

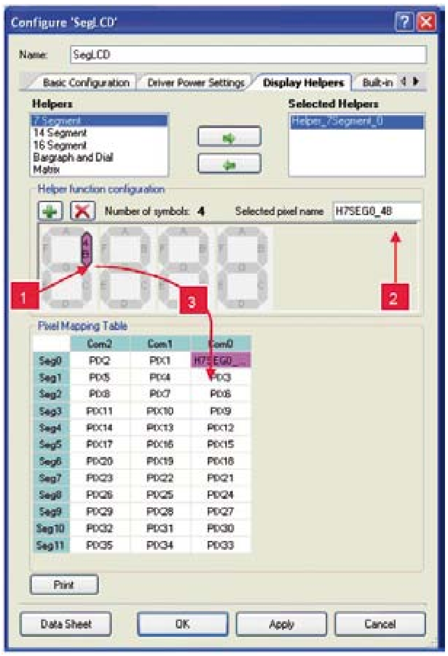


Figure 4. Adding digits to the Helper.

Listing 1. The project's main.C code.

```
for(;;)
{
      CyDelay(1000);
      SegLCD_Write7SegDigit_0(digitValue,digitPosition);
      digitValue++;

      if(digitValue > 0x10)
      {
          digitValue=0;
          digitPosition++;
          SegLCD_ClearDisplay();

          if(digitPosition==4)
              digitposition=0;

      }
}
```

The parameters in the Display Helpers tab are shown in **Figure 3**. For this example you will only need one of the helpers, the 7 Segment. First select 7 Segment (1). Then click on the right arrow (2) and the Helper_7Segment_0 is seen in the Selected Helpers field. Finally, click on Helper_7Segment_0 (3) and the window will expand, which is shown in **Figure 4**.

By pressing the green plus button three times you obtain the four seven-segment displays show in Figure 4. The next steps are to select a pixel, rename it, and assign it to the pixel mapping table. These steps are indicated in the screendump. First click on segment B of the fourth symbol referring to the LCD glass (1). In the selected pixel name field, you can name the pixel whatever you want (2). Drag and drop the selected pixel to the pixel mapping table below (3). Make sure that the pixel mapping matches your LCD's datasheet.

After repeating this procedure for all of the pixels, verify that the relationship between the segments and the commons of the glass is maintained in the pixel mapping table.

The PSoC device's pins can be mapped from the tool using the design wide resources file. A unique feature of PSoC's LCD direct drive is that any GPIO can be configured either as a common or a segment. This makes board routing much easier. You can map the PSoC device's pinout by dragging and dropping.

Now do some coding

The Segment LCD component provides user friendly APIs (Application Programming Interfaces) to develop real life applications for the LCD. The component's datasheet provides a full list of the APIs, but they allow for the user to display text and numbers on the screens with simple function calls. All of this is enabled from the Display Helpers tab. The remarkably simple code for this example project is shown in **Listing 1**.

Now that the code is complete you can build and test the project. The project can be built for either a PSoC 3 or PSoC 5 target device. After selecting the target device, you can build the project; program the chip using the PSoC Creator software and a device programmer. After programming you can reset the device and observe its behavior. If any problems are found, you can use PSoC Creator's built in debugger.

That's it, you're done. You now have a four digit, seven segment LCD display that can write any text or number with simple, predefined APIs.

(100854)

Internet Resources

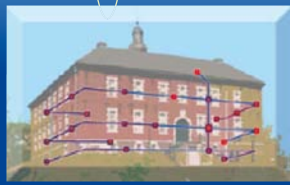
www.cypress.com/go/psoc3

www.cypress.com/go/psoc5

www.cypress.com/go/psoccreator

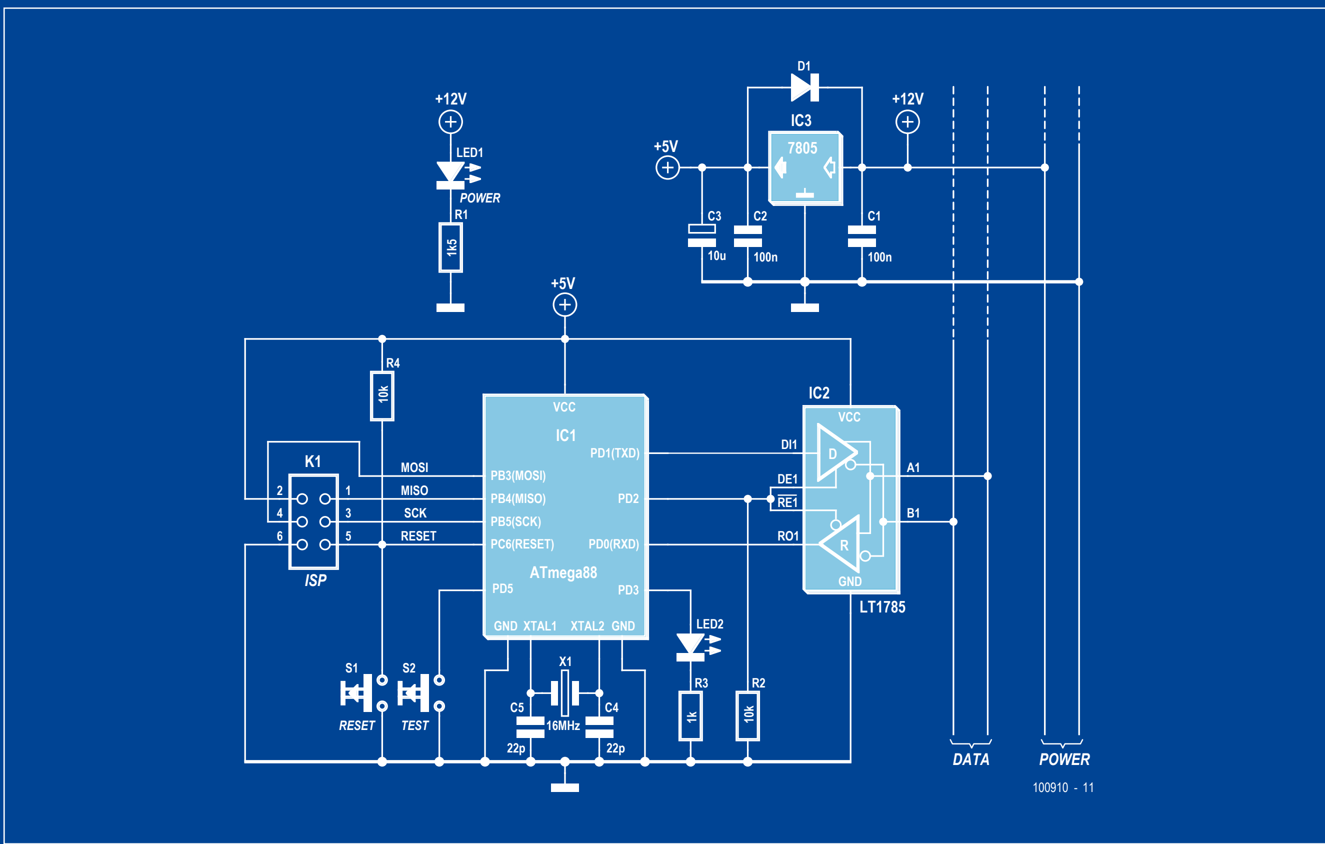
www.cypress.com/go/cy8ckit-029

Here comes the Bus! (3)



By Jens Nickel (Elektor Germany Editorial)

As I sit down to write these words, the January 2011 edition of the magazine, containing the first part of this series, has already been on sale for two weeks. We have already had around a dozen e-mails full of ideas and suggestions: thank you very much! Two readers preferred the CAN bus over the RS-485 bus, since it offers a collision detection protocol which could be very useful in a home automation system. Also, CAN transceiver devices are not too expensive, and so this approach seems like a good idea. However, the time will surely come when we want to squeeze just a little more bandwidth out of the bus, which will probably be possible with RS-485; standard CAN, however, is limited to 'only' 1 Mbit/s. Furthermore, the basic RS-485 bus gives more room for experimenting and new designs. If we had been focussed solely on results, we might well have done better to take greater advantage of others' work in home automation and other buses, as was indeed pointed out in several other e-mails.



Another reader suggested using the one-wire bus, which is used for connecting to devices such as temperature sensors. In the first part of this series we set ourselves the target of being able to make a node (microcontroller, support circuitry, RS-485 transceiver and the rest) for no more than fifteen Euro, as we could easily imagine home automation systems with a hundred nodes or more. This presents an opportunity to save money, as we can connect several sensors to a single node using a one-wire bus (or, equally an I²C bus), with the node providing the full RS-485 bus functionality, and we plan to do something along these lines in a later instalment in this series. We also plan to take a look at the possibilities of wireless sensors (another reader suggestion). Other e-mail correspondents were concerned with the problem of power supply over the bus. Readers Markus Aebi and Fabien Noir independently suggested using a 24 V supply rather than 12 V, in order to reduce power losses in the cables. This is particularly important in larger networks. To analyse this question properly we need to know whether we will be allowing power-hungry nodes (for example, those containing several relays) to be bus-powered. It also makes a difference whether the microcontroller power supply will be derived from a linear regulator or a step-down switching converter. And of course we need to know the number of nodes connected to the bus and a few other things. For our experimental system (see below) we have preliminary answers to some of these questions, but the issues of power supply over the bus are sufficiently complex and interesting that we shall return to them later in this series.

In the previous part of this series we decided that our nodes would bring a little intelligence to the bus. For example, a node

might implement a simple control loop autonomously, or automatically monitor a value to detect when it exceeds some threshold. These requirements entail the use of some non-volatile storage for configuration data in the node, which in this case means an EEPROM. This also lets us implement dynamic addressing: the node is issued with an address from a central point, uniquely identifying it on the bus. Without this feature, each node would have to be 'factory' programmed with an address. Since we would like these addresses to be globally unique, this involves quite a lot of work, perhaps even allocating an address range to each of our readers! A simpler alternative is to allow the address of a node within its particular bus segment to be set manually using a DIP switch or jumpers. In practice we might allow eight bits of address (more might put too much strain on the fingers setting DIP switches!), which in turn puts a limit on the number of nodes on one bus.
So we kept the idea of dynamic addressing, and face a chicken-and-egg problem: if a bus node does not have a unique address, how can we access it over the bus in order to issue one?

One approach that might be used with the master-slave system we described in the last issue would be for a slave to report to the master when the latter is in 'listen mode' waiting for an event: 'hello there, I'm new here and don't have an address yet'. The slave could also report what functions it can carry out. An 'at your service' report like this could be sent out automatically when power is applied to a node, or could be triggered manually by pressing a button on the node: the button could also be used for test purposes. In a home automation network you would first put the master (probably PC-based software) into listen mode and then wander around the building either power cycling or pressing the button on each node.
And how would this work in the system suggested by my colleague Clemens, where a scheduler polls the bus nodes in order? With a little thought I realised that the scheduler could ask the connected nodes 'hello, is there anyone out there who I haven't seen before?', and then each new bus participant could reply. On the next scheduling round the device would be assigned an address. It is not possible for several new bus nodes to start up simultaneously in this way, but nevertheless we can realise a kind of 'hot plugging' facility within the scheduler and enable the nodes one at a time.

Enough with all this theoretical talk: let's get down to actually making a first test system. For initial experiments we will need a PC with a USB-to-RS-485 adaptor to act as the master or scheduler, and perhaps two or three nodes. We can use the USB-to-RS-485 converter published in the December 2010 issue [1], which, in a further article in that issue [2], we modified for half duplex operation.
Since at the Elektor Labs we have some experience in using AVR microcontrollers, and we know that these are popular with our readers, we decided to use one as the controller in each node. Also, free tools are available for programming these devices in C, and BASCOM provides a good BASIC development environment. An ATmega device costs just a couple of pounds, so why not use our old favourite the ATmega88, as featured in the ATM18 project [3]? This device includes an A/D converter and 512 bytes of EEPROM [4], and many readers will already have experience with it.

With the help of my editorial colleague Thijs I sketched a first circuit diagram (see figure) and put together a brief parts list. One of Thijs' hobbies is electronics (the other is playing drums in a rock band) and from the next issue onwards he will turn professional, reporting from his home lab in the E-Labs Inside pages of the magazine (but perhaps not on the subject of drumming).

Four two-way screw terminal blocks are needed so that the bus signal can be looped through the node to implement the bus topology we described in the previous instalment. (Suggestions for improvements to the design in this area or elsewhere gratefully received!) LED1 indicates when power is present and LED2 is used for testing, as is one of the buttons. The other button (reset), the programming header (compatible with the *Elektor* AVRprog [5] and other programmers), the crystal and the passive support components are standard. The same goes for the voltage regulator, which produces the 5 V node supply from the 12 V supplied on the bus. Thijs and I wondered briefly about the possibility of using a step-down switching regulator (diverging from the design of the ATM18 test board) to reduce the current drawn from the bus. However, this would have increased the price of a node somewhat, and in any case we wanted to keep everything as simple as possible. We don't expect the current consumption of the test system to be very high, and we plan to provide any node with an actuator of any kind with its own supply.

Günter Gerold (the chap behind the Wheelie GT) kindly told us about the LT1785 [6], which is easier to use than the LTC1535 and has better overvoltage protection, which is always good to have when experimenting. Linking the microcontroller and the transceiver there are two data signals (receiver-out and driver-in) and a direction control signal which determines whether the transceiver is transmitting or receiving (from PD2 to /RE1 and DE1). These last two pins can be connected together on the transceiver since we will always either be receiving or transmitting and we do not use the shutdown state (where /RE1 is taken high and DE1 low). Since RXD and TXD on the ATmega88 are part of port PD, we decided to connect the direction control signal, the test button and the test LED to this port too. The microcontroller has internal pull-up resistors on these port pins, and so we don't need an external pull-up for the test button.
We also want to test the node thoroughly using different transceiver devices. The bus should be sufficiently open, flexible and universal that we are not dependent on using a particular IC. We do want readers to be able to build bus nodes in thirty years' time!

(100910)

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

[1] http://www.elektor.com/100372
[2] http://www.elektor.com/100369
[3] http://www.elektor.com/071035
[4] http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf
[5] http://www.elektor.com/080083
[6] http://cds.linear.com/docs/Datasheet/178591fc.pdf

Two newcomers in the E-Lab

By Jens Nickel (Elektor Germany Editorial)

Every now and again professional developers feel the urge to update their test and measurement (T&M) gear. That applies equally to our lab of course, where last year we had an urgent need for new scopes. My colleague Antoine, at that time director of the labs, settled for two middle-range digital storage oscilloscopes (DSOs) from trusted brands: Tektronix and LeCroy. So we called up distributor Distrelec [1] to send us a LeCroy WaveAce 224 [2] plus a Tektronix TDS2024B [3] with the same basic specifications: 4 channels, 200 MHz bandwidth and a maximum sample rate of 2 gigasamples/s (per channel). Both devices use a 6-inch colour LCD display in QVGA resolution (320 x 240 pixels) and are equipped with a USB connector for hooking up to a PC plus a USB host interface for memory sticks and external drives. Beyond this you don't get a huge amount more for your money; the LeCroy can be found here for around £1,520 plus VAT from the respective suppliers but you'll have to fork out closer to £1,770 plus tax for the Tektronix. The similarity of the specifications inclined us to arrange both machines side-by-side on the bench for technical comparison.

Our T&M specialist Harry, Luc for our lab and I were first to switch on the scopes. It may sound trivial to many readers to mention this but far from it: a DSO is really no different from a small computer running an operating system. And all computers need to be run up first, which can seemingly take an age. With the Tek we timed this as more than 30 seconds but the LeCroy was up and running, ready to use, in just 15 seconds. In reality this delay is not a serious disadvantage, even when one or two electronicists in their daily work need to use the scope 'just for a quick test'. People who are used to instant availability from their analogue equipment have simply got to modify their expectations.

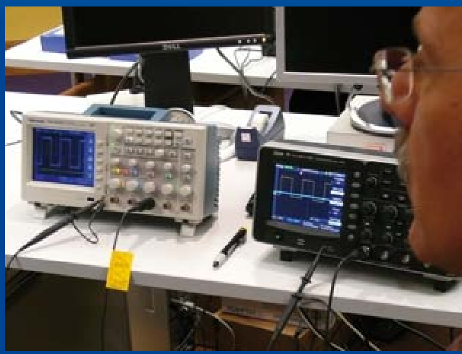
While we were waiting we had time to consider the scopes' usability. For me at least, coming with no particular preconceptions on this subject, it was a bit of a revelation: the knobs for the basic functions (timebase and attenuators, trigger, and so on) were located almost identically on both devices. This continued for the press switches for selecting menu options, even down to details such as the button for calling up mathematical functions.

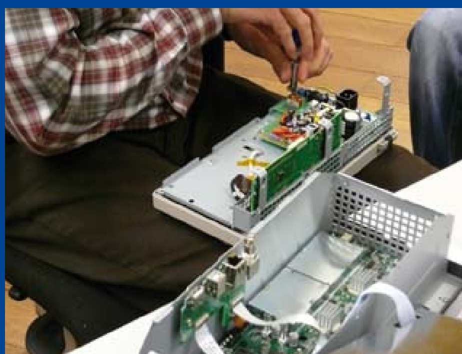
Next we examined the test probes and connected them to the built-in squarewave generator provided for equalising them. With three people on the job we discovered a shortcoming of the display — the restricted angle of view. This is particularly apparent with the Tektronix and anyone not sitting directly in front of the display will literally be straining to see it.

Our T&M expert Harry set about animating some curve traces, to check what happened when the deflection and amplification were changed. During calibration these frequently need to be aligned exactly to specified reference voltages. With analogue scopes this can be done smoothly and continuously of course. Because DSOs work on a different principle, we cannot do this the same way; what we observe is the result of calculations, involving a slight delay.

Comparing the traces on the two displays showed up a small difference: the Tektronix trace seemed more solid because the always-visible noise made curves somewhat blurred in width. On the LeCroy you can see the pixels dancing around, giving us the subjective impression that this device presented a more accurate rendition of what was actually going on. Overall, however, both devices earn equal praise when it comes to signal rendering: the contrasting colours used provide a good overview of what's going on at all times. Two small markers serve additionally to display the trigger level and position.

Next we wanted to check out some 'higher' functions. As the devices are equipped with some computing power (the LeCroy incorporates a Blackfin DSP BF531 from Analog Devices for example), mathematical functions such as Fast Fourier Transforms are merely a matter of software. Adding and multiplying signals and suchlike is no longer problematic either. Both scopes score on these features and on their versatile trigger functions. Admittedly you will be hunting for the handbook frequently, both at the outset and also later on when the function you need is not available direct from a front-panel button. With both devices menu navigation is not exactly intuitive, which granted





is not easy to achieve with only a couple of menu buttons and a fairly restricted display available (makers of USB oscilloscopes have it easier here). Another disappointment was Luc's failed attempt to rouse the network interface of the WaveAce into life. He did succeed rapidly in entering IP addresses and suchlike into the menu — but how do you know for certain that you are inputting the right figures? And with the Tektronix we were never quite sure whether the changes we input had actually been accepted.

Enough of this playing around; it was now time to roll up our sleeves and get down to more serious investigation. Harry (*photo*) and Luc wielded their screwdrivers on the casings. The interior workings of the scopes can be seen in the photo; here we would award a slight advantage to the LeCroy (*the lower device in the photos*). We liked the clearly sectionalised construction and the superior screening: the general electronics and the input amplifiers are both fully screened with sheet metal. In the Tektronix the electronic assembly is divided between several printed circuit boards (one of these with really awkward access). This modular approach of course has its advantages, allowing the input amplifiers to be separated spatially from the computing and digital chips. In this section we noted a MC68SEC000 microcontroller and a CY7C67300 USB controller by the way.

Once we had reassembled the DSOs it was time to substantiate their capabilities in our lab, specifically on a diet of RF signals. First to be deployed was our HP3325 synthesizer-function generator. To compare the way that traces were rendered both devices were connected in parallel to the test reference signal using a BNC T adapter. With a 10-MHz squarewave on Channel 1 the Tektronix and the LeCroy each demonstrated a typically crooked rising leading edge and a steeply falling trailing edge. But oh dear: on the Tektronix we spotted a distinct dip in the signal ceiling, whereas the LeCroy displayed a slight upward bulge. This is an aberration that frankly we could not explain. Finally (and shamefully) we resolved this mystery; we had not terminated the BNC cable by the book, meaning that the scopes were indicating very accurately the signal overlaid with all the cable reflections! After we introduced a further T adapter and a stuffer plug (terminating resistance of 50 ohms) into the chain, a still slightly rounded but totally consistent square wave was presented. Admittedly we could still detect (now minimal) differences in the signal shapes but probably this could be blamed on the reflections from the T adapters. After we had swapped position of the scopes it was clear (to me at least) no variation in the curve form could be detected any more. Finally we brought in the really big guns in the form of an HP8640B signal generator (even Harry conceded a professional respect for its weighty rotary controls). We began with a 100 MHz sinewave and raised the frequency gradually. Up to around 300 MHz both oscilloscopes still produced a clean, stable signal display but then the triggering failed. The bandwidth of 200 MHz was thus confirmed.

Bottom line: the oscilloscopes present a solid impression and perform precisely as how it says in their specifications (this is by no means always the case with T&M gear). The user features are not to be sneezed at either. Nevertheless it's evident that equipment developers in this price class cannot avoid making some compromises. This starts with the display and continues with the DSOs reacting somewhat lethargically now and then. The LeCroy has the slight edge with marginally better graphics and the authentic action of the signal display, although the last-mentioned (along with black front panel) are something of a matter of taste.

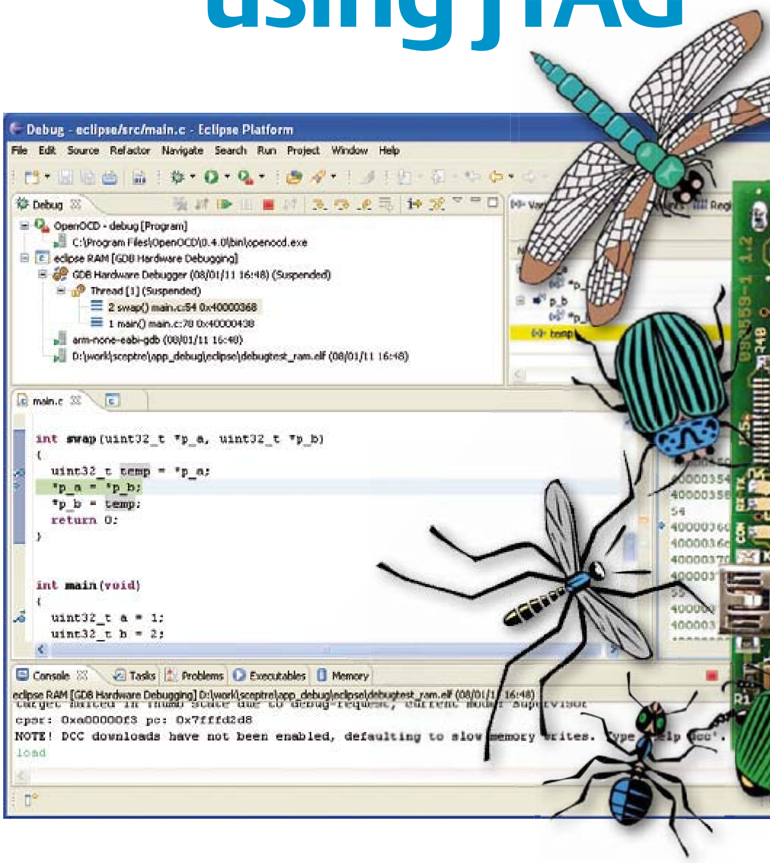
This emphasises once again that you should buy an oscilloscope according to your exact requirement. If you can manage with two channels (instead of four) or if 60MHz bandwidth will suffice in place of 200 MHz, you can already shave a couple of hundred pounds off your outlay. If the specification extremes are required only occasionally (as in our lab), then these devices may represent a good compromise. On the other hand, people whose daily bread involves multichannel measurement should consider whether they might not invest a bit more money to secure, for example, a larger display.

(100451)

[1] https://www.distrelec.de/ishop/StaticHTML/shared/distrelec/
[2] www.lecroy.com/oscilloscope/OscilloscopeModel.aspx?modelid=2121&capid=102&mid=504
[3] www.tek.com/products/oscilloscopes/tds2000

Debugging the Sceptre using JTAG

BY Clemens Valens (Elektor France Editorial)



Developing a short program for a microcontroller board can be done using very simple means, but when the software starts to get a bit bigger, these simple means very soon start becoming a limitation, or even bring you to a grinding halt. Elektor's Sceptre has 512 KB of program memory, and to take full advantage of this, you need appropriate tools, such as a debugger worthy of the name and a tool that loads the executable into the board memory quickly. Buying commercial tools for several thousand pounds is only for the professionals, for whom the time saved is worth more than the cost of the programming tools; so amateurs will have to get by some other way. Fortunately, there are some solutions available.

In the case of the Sceptre [1] and the other members of the large family of boards using ARM processors (and not only ARM), the solution is called JTAG. ARM has specified a JTAG interface using a 20-pin (2×10) connector, which has become fairly standard since a large number of board manufacturers have adopted it. Elektor has done the same and has included it on the InterSceptre (June 2010) and the Automatic Running-in Bench (April 2009). This interface makes it possible to debug not only the hardware, but also the software and flash memory programming.

We're not going to talk here about hardware debugging, but confine ourselves to a single component on the JTAG bus: the microcontroller. Even though in the first instance this article addresses the Sceptre, and hence an NXP LPC2148 microcontroller with an ARM7TDMI-S core, the techniques described are just as valid for other controllers. In most cases, all that is needed is to adapt some of the configuration files.

The tools we're going to be using are **GDB** [2] and **OpenOCD** [3]. The former is the GNU Project Debugger open-source debugger, the latter is also an open-source debugger, OCD stands for On-Chip Debugger, but works at a lower level. However, for debugging software for the Sceptre we need to use both of them. Once we've taken care of our first steps (that's what debuggers call them), we'll add a layer to improve convenience with a graphical interface. But before we get to that point, we're going to have to start at the bottom of the ladder: the Command prompt.

OpenOCD and GDB

Setting up a debugging environment based on GDB and OpenOCD may seem a bit off-putting at first sight (and even at the second), which is why we're going about it gently. **Figure 1** shows the block diagram of the environment we're going to be setting up. Right over on the left, we have a microcontroller board, a Sceptre

Elektor Products & Services

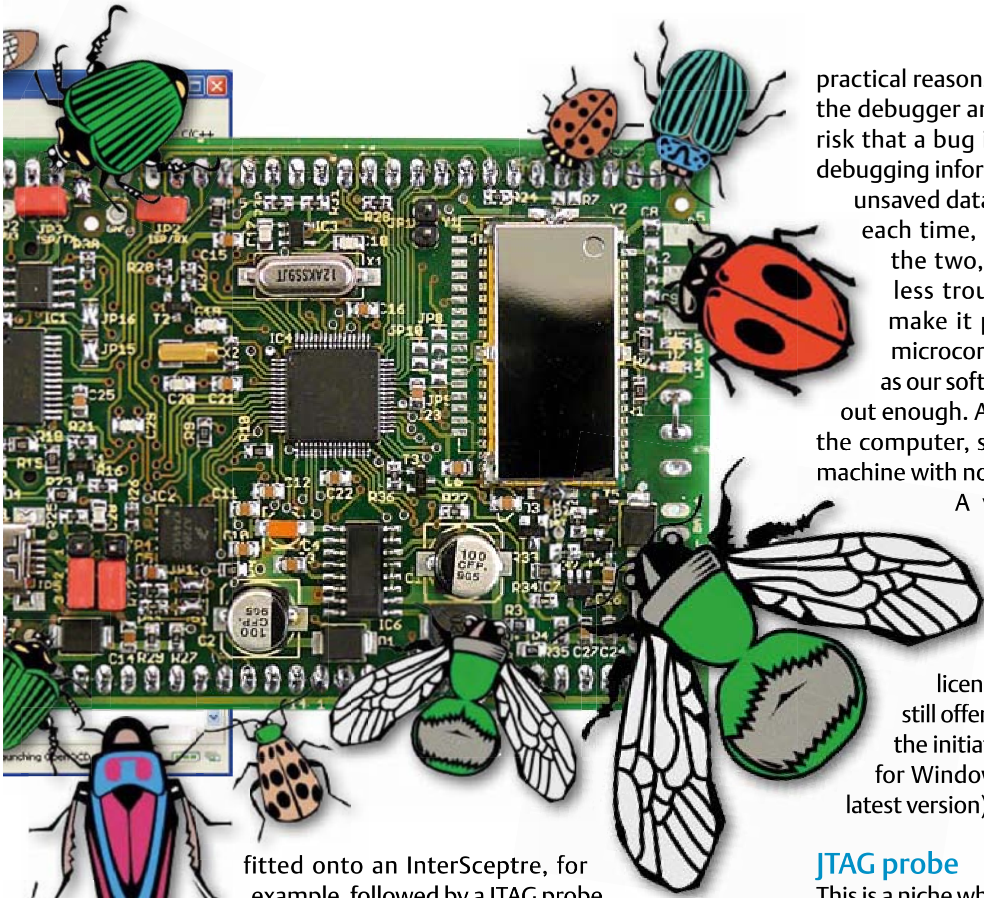
- Test project (free download): 100810-11.zip
- Sceptre board: Elektor # 090559-91
- InterSceptre board: Elektor #100174-71
- Hyperlinks used in article

All items accessible through www.elektor.com/100810/

Also:

- Forum: www.elektor.com/forum
- Blog: http://elektorembedded.blogspot.com/

With the help of OpenOCD, GDB, Insight and Eclipse for Windows



fitted onto an InterSceptre, for example, followed by a JTAG probe. This is a little chunk of hardware that converts the JTAG bus into a USB or parallel port (or even some other type) so as to be able to connect up to a computer. The JTAG probe is driven by a GDB server, a piece of software able to convert high-level debugging commands into low-level JTAG operations. OpenOCD is going to be acting as our GDB server. The GDB client, the GDB software itself, sends debugging requests to the server and processes the replies received. And lastly, a graphical interface completes the environment. This will save you a lot of work by looking after sending the large number of commands needed for debugging a piece of software, and will present the results in a practical, convenient manner.

It may seem odd to split a debugger into a server and a client running on the same computer, but this split has been done for practical reasons. In practice, if the software to be debugged and the debugger are run on the same computer, there is always the risk that a bug in the former may crash the whole system. The debugging information obtained is then lost, along with any other unsaved data, and it will be necessary to restart the computer each time, which ends up getting tiresome. By separating the two, a crash in the system to be debugged is much less troublesome. In our case, this architecture would make it possible to run the GDB server directly on the microcontroller board, but we're not going to do this (yet), as our software to be embedded (still) hasn't been developed out enough. At any rate, if the board crashes, this will not crash the computer, so we can run the server and client on the same machine with no danger.

A version of GDB for Windows is included in programming tool chains for ARM like WinARM, for example (see the article about the Sceptre [1]) and Yagarto [5]. For OpenOCD by Dominic Rath, it's a bit more complicated, because, as the original version used libraries from FTDI without a GPL licence, it is no longer included within Yagarto. WinARM still offers an old version, but a certain Freddi Chopin has had the initiative to produce a version 100 % under a GPL licence for Windows. So download and install OpenOCD 0.4.0 (the latest version) by Freddi Chopin and kick off!

JTAG probe

This is a niche where many electronics merchants are trying to make a bit of money by selling JTAG probes that are more or less powerful. Broadly speaking, the differences between all these probes lie in the maximum speed of communication on the JTAG bus (the faster it is, the more convenient the debugging) and in their ability to program the microcontroller or not. Before you go out and order a probe with all the bells and whistles, be aware that OpenOCD is perfectly capable of programming a large number of controllers and flash memories, including those of the Sceptre, at a perfectly acceptable speed (depending on the probe). The programming option is of particular interest to a production unit that has to program a large number of chips.

The speed of the probe is important for the speed of debugging. There are a lot of bits to be moved around for each JTAG operation, and each debugging requires several JTAG operations. There is an appreciable

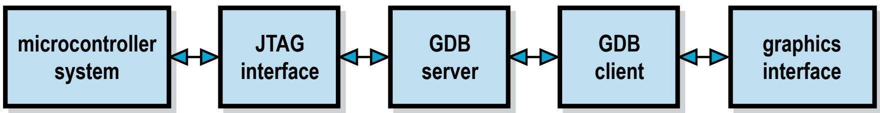


Figure 1. Block diagram of the GDB debugging chain for the Sceptre.

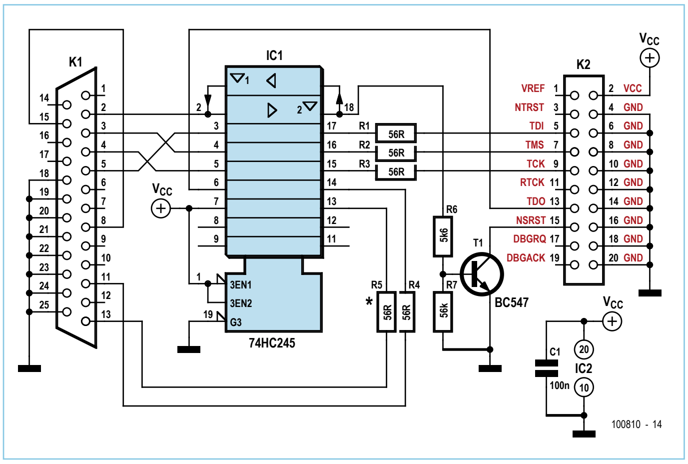


Figure 2. Circuit of a Wiggler clone inspired by the various circuits available on the Internet and the components available. R5 is optional.

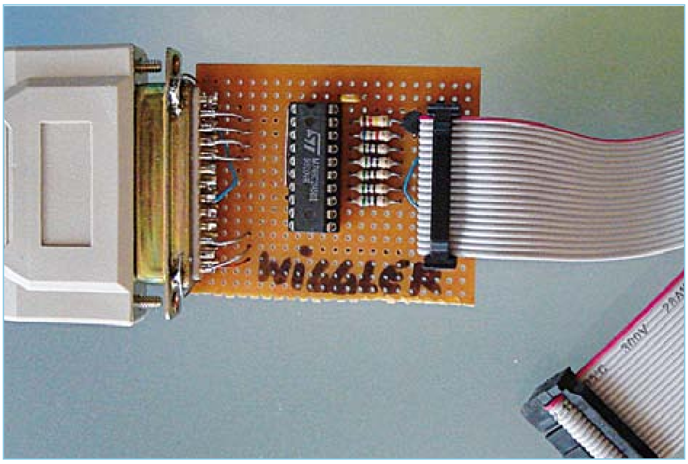


Figure 3. Our Wiggler built on prototyping board.

difference between a probe on the parallel port capable of producing a JTAG clock at 5 kHz and a USB probe that achieves 6 MHz.

We tried three probes: a basic Wiggler-compatible one [4] on the parallel port, the popular ARM-USB-OCD USB probe from Olimex [7], and the J-Link Edu from Segger [8], which is a version of its professional probes cut down for non-commercial use. The Wiggler is easy enough to build yourself (**Figures 2 & 3**), but it's very slow (the test program is loaded at 957 bits/s, compared to 19 KB/s using the Olimex probe; running a simple line of code in C takes around 3 s). To make this probe work, your computer's parallel port must be in EPP mode.

The J-Link Edu from Segger is driven by its own GDB server, which is not entirely compatible with OpenOCD, and certain commands are different. OpenOCD does support a J-Link interface, but unfortunately we didn't manage to make it work with the J-Link Edu. Since the aim of this article is to explain how to debug using only open-source software, we didn't investigate the J-Link Edu option any further.

In what follows, we're going to be using the ARM-USB-OCD from Olimex as our JTAG probe.

Debugging in text mode

Check first that Windows (or you yourself) can find OpenOCD and GDB.

Connect your JTAG probe to the computer and the microcontroller board, open a Command prompt, and run OpenOCD like this:

```
openocd -f interface/olimex-arm-usb-ocd.cfg -f
board/elektor_sceptre.cfg
```

The CFG files specified depend on your hardware. We're using the ARM-USB-OCD probe from Olimex with a Sceptre board fitted to an InterSceptre board.

Open a second Command prompt and execute:

```
arm-none-eabi-gdb
```

Next, we need to execute a series of commands to connect GDB to OpenOCD and to load the executable in the controller's RAM memory (when debugging a program in RAM). The commands must be entered in GDB (indicated by the `(gdb)` prompt):

```
(gdb) target remote localhost:3333
```

This rather strange command pretends that we're going to debug our target remotely (`remote`), although it is actually connected to our computer (`localhost`). If OpenOCD is listening on port 3333 (the default value), you should see appear in the OpenOCD window an `Info` message saying `accepting 'gdb' connection from 0` (**Figure 4**). GDB and OpenOCD will now be able to communicate. Before being able to load the executable into the controller, the latter must be halted. GDB can't stop the processor, but OpenOCD can do it. The `monitor` command makes it possible to execute an OpenOCD command from GDB:

```
(gdb) monitor reset halt
```

If this command, intended to perform a `reset` followed by a `halt`, has worked properly, the microcontroller is now stopped. Before continuing, check that GDB and OpenOCD are displaying the message `target state: halted`. If this is the case, we can load the executable file into GDB and into the microcontroller (if necessary):

```
(gdb) file test_ram.elf
(gdb) load
```

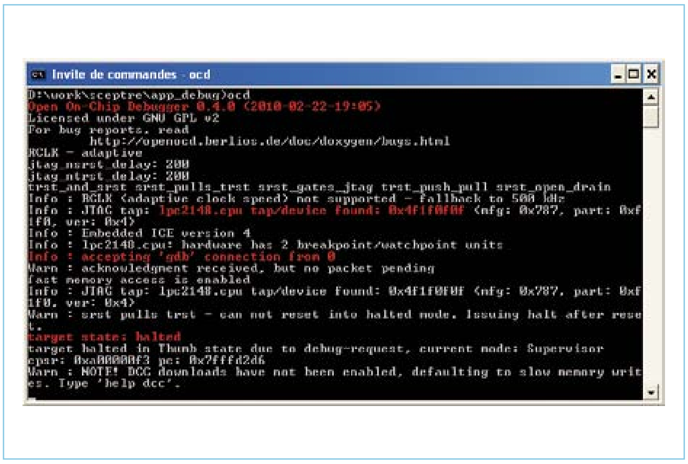


Figure 4. OpenOCD following acceptance of a GDB connection. It's not very clear, so we've put some of the more interesting details in red.

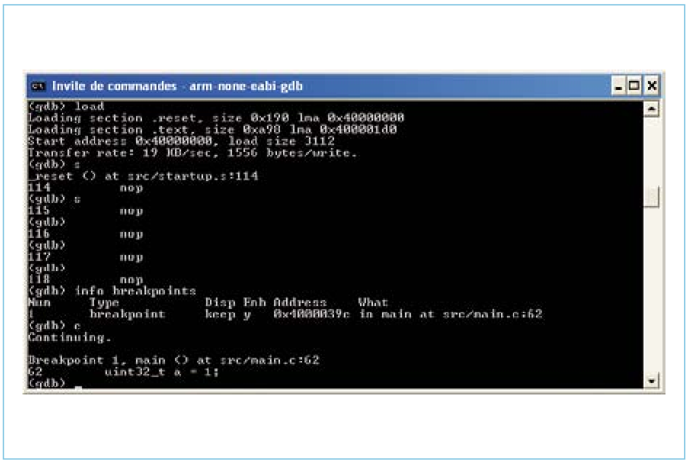


Figure 5. Text-mode debugging using GDB.

The first command reads the file and the second loads it into the microcontroller's RAM memory. Note that you must not execute the `load` command if the program is already loaded into flash memory. We found we were unable to successfully debug a program in RAM while another was already loaded in flash.

The file to be used is in the ELF format (not HEX or BIN), as GDB is going to need additional information. For effective debugging, the program must be compiled with a debugging option, which adds information into the executable that is needed for convenient debugging, like the names of the variables and functions. If you examine the `makefile`, you'll find for the GCC C compiler the option `–gdwarf-2`, which indicates that the compiler will include debugging information in the Dwarf 2 format in the executable. It's thanks to this information that the `step` command (for example) is able to display the source code corresponding to the point you're at in the program.
Now, execute the `step` command a few times, like this:

```
(gdb) step
```

When you want to repeat the previous command, all you have to do is press Enter. After each step, GDB displays the line of the file carrying the next instruction that the microcontroller must execute, along with the current line itself (with comments, if there are any). At the outset, you'll find yourself right at the start of the program, the part that's usually written in assembler that initializes the microcontroller. To jump to the `main` without getting lost in the memory initialization and other loops, insert a breakpoint at the start of the `main` then let the processor run:

```
(gdb) break main
(gdb) continue
```

A little later on, GDB issues a message along the lines of:

```
Breakpoint 1, main () at src/main.c:73
```

and there you are in your code (**Figure 5**).
In principle, we ought to always do what we've just done at the start of a debugging session. Rather than retype the same commands each time you launch GDB, we can insert them in a commands file (a script) that GDB will execute automatically. By default, GDB looks to see if there isn't a file named `.gdbinit` (note this well) somewhere around. If this is the case, it opens it and executes the commands contained in the file. We can also specify a file with another name, thanks to the option `-command`, like this:

```
arm-none-eabi-gdb -command gdb_cmd.txt
```

Be aware that the use of a script file can sometimes lead to difficulties — perhaps because the commands are sent too fast one after the other? If this happens, you need to interrupt GDB using `<ctrl><c>`, stop the microcontroller (`monitor reset halt`) and, in the case of a program in RAM, reload the program using `load`.

Now we're at the start of our program, the debugging proper can start. This is the moment to position some breakpoints, to inspect the variables, registers, memory, or the stack. Unfortunately, the number of hardware breakpoints, i.e. those that are handled by the hardware itself, is limited. The Sceptre's microcontroller, the LPC2148, offers only two hardware breakpoints, which is not really very many. That's why it's worth debugging a program in RAM, which makes it possible to use software breakpoints handled by GDB and not by the hardware. The number of software breakpoints is in theory unlimited.

Table 1. The most-used GDB commands with a short description.
Most of the commands accept any sort of parameters. Consult the help or the Internet for further details.

Command	Description
`apropos`	Lets you search in the help on a keyword.
`backtrace (bt)`	Shows where you are in the program according to the stack.
`break (b)`	Set a breakpoint. E.g. break main
`clear`	Deletes a breakpoint.
`continue (c)`	Continues execution of the program. <ctrl><c> lets you interrupt the program.
`delete (d)`	Deletes one, several, or all breakpoints.
`finish`	Terminates the current subroutine.
`help (h)`	Displays gdb help. help followed by a command name lets you obtain help about that command. E.g. help print
`info (i)`	Displays additional information about something. Must be followed by a parameter, like 'info sources' to display a list of the source files used by the program.
`list (l)`	Lets you display a number of lines (default = 10) of the source code.
`next (n)`	Executes the next line without entering a subroutine, i.e. the subroutine call is treated as a simple line of the program. The subroutine is executed.
`print (p)`	Shows the value of a variable or register.
`quit (q)`	Terminates GDB.
`run (r)`	Starts execution of a program from the beginning.
`set`	Lets you enable or disable a GDB option.
`show`	Shows the status of a GDB option. show without any parameters displays all of them.
`step (s)`	Executes the next line. Enter into a subroutine.
`until`	Continues execution until it reaches a certain line or subroutine. until toto is equivalent to break toto ; continue
`watch`	Stops execution of the program when the specified condition becomes true.
`where`	See backtrace.
`<ctrl><c>`	Forces GDB to stop execution of the program.
`<enter>`	Repeats the last command.

When the program is too large to be run from the RAM (the Sceptre's microcontroller has only 32 KB, to be shared between the program and the data), it has to be loaded into the flash memory and we can't then use software breakpoints. Certain JTAG probes in this case do allow 'hardware' breakpoints to be added. The J-Link Edu from Segger, for example, offers an unlimited number of hardware breakpoints.

The commands used most in a debugging session are probably `step`, `next`, `finish`, `continue`, `break`, `delete`, `list` and `print`. A list of these commands, with a short description, is given in **Table 1**. This table is not exhaustive, practically every command can take several parameters, and still other commands are not included in this list. You'll be able to find several sites on the Internet offering the missing details relating to GDB. Do note however that there are several versions of GDB, which are not necessarily 100 % compatible. Some use a slightly different syntax and others do not have all the commands. So don't be surprised if the explanations on a website don't work with your particular GDB — don't be afraid to research a bit further. Be aware too that certain GDB commands and functions won't work with your hardware, quite simply because it doesn't happen to support them.
Before launching yourself into using a graphical interface, it's worth familiarizing yourself a little with the commands in text mode and using the GDB console. Try, for example, to understand the difference between `step` and `next`, examine the registers, take a look at the stack, etc.

Adding a graphical interface

Even though debugging in text mode is very powerful and instructive, typing in all the commands manually soon gets tiresome. So to make the debugger's life a bit easier, several graphical interfaces have been developed for GDB. One graphical interface (GUI, from Graphical User Interface) lets you see the source code without executing the `list` command, and keeps the variable, register, and stack values automatically updated; it displays the breakpoints clearly, along with the point you're at in the program. Several GUIs exist, but not all of them are suitable for debugging a microcontroller board. The two GUIs most used for this purpose are **Eclipse** and **Insight**. Eclipse is a powerful, sophisticated, free, integrated multi-platform environment. By adding a plug-in, it also be used as a GUI for GDB. But setting up an Eclipse environment for GDB is a bit complicated, so we're going to start by taking a look at Insight, a GUI born and bred for GDB.
Insight is a free, open source Linux application supported by Red Hat. It's difficult to compile under and for Windows, and what's more, it's quite hard to find an Insight executable pre-compiled for Windows. Reason enough for us to abandon this route and

trying something else. So, why go on anyway? Because once you master Insight, it doesn't work badly at all. It's a tool that makes for convenient debugging, without useless or cumbersome options. In fact, an Insight executable for Windows is included in WinARM, the tool chain we've chosen for the Sceptre [4]. Certain distributions of Yagarto include it too, but apparently not the most recent distribution.

Unlike Eclipse, Insight incorporates GDB, so when you already have Insight, there's no point installing GDB as well. In some ways, the tool is a graphical GDB and when you run it, it reads the same initialization script (`.gdbinit`) as GDB. If this script is correct (Insight is a bit quirky and you have to obey the correct order for certain commands), the debugger launches and shows the source code highlighted on the line where the program stopped (if a breakpoint has been set in advance, naturally). Note that it's imperative to launch OpenOCD as previously described before starting Insight.

From the main window (`Source Window`, **Figure 6**), we have access to additional windows for displaying the local variables, registers, stack, memory, breakpoints, and the GDB console. As you move around within your program, all these windows are updated by the software, with the latest changes highlighted. It's a bit more convenient that typing `print` commands after each `step`.
For the GUI to be able to update everything that's displayed, it takes a lot of extra JTAG transactions each time a command is executed. If your JTAG probe is slow, this can take some time, whence the interest of getting yourself a suitable probe.
Insight's GDB console lets you do the same thing as the GDB console described earlier, except that the results are displayed in other windows. Access to the GDB console is very handy when you lose control of the debugging and Insight has to execute commands it doesn't know (for example, `monitor reset`).
It sometimes happens that an operation leads to loss of the connection between GDB and OpenOCD without your realizing — for example, if you load a new file to debug. So do keep an eye on the OpenOCD window to check that the connection is still in place.

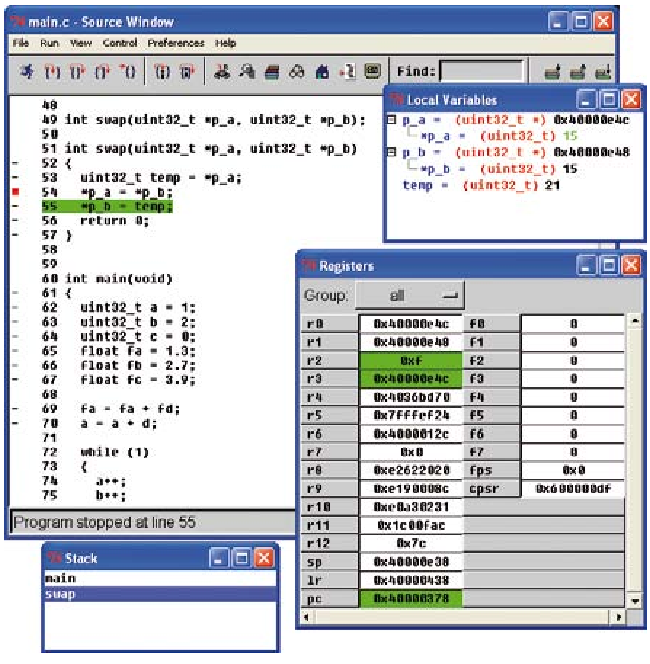


Figure 6. Insight is a graphical interface for GDB.

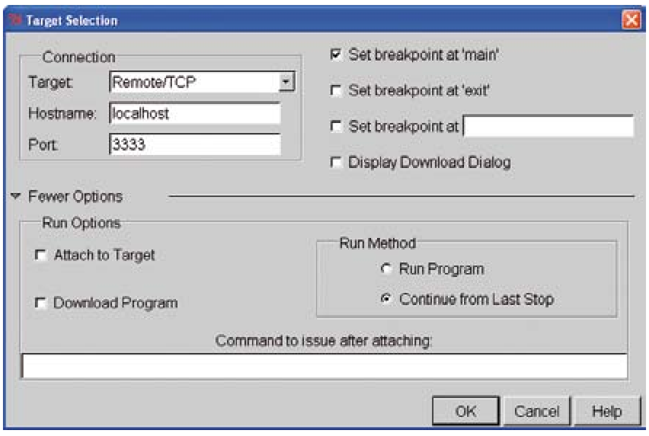


Figure 7. Parameters for Insight that work well.

Let's end this section with a few general comments.

- On our test computer, launching the copy of Insight included in WinARM produces a warning `Unknown ARM EABI version 0x5000000`. Ignoring this warning doesn't seem to stop the tool working properly. If you know where to find a more recent version of Insight pre-compiled for Windows, please let us know.
- **Figure 7** shows how to configure Insight by hand (`File -> Target Settings...`).

Better still

Insight is already a very good tool for debugging an application running on a microcontroller board, but there is better yet. Eclipse (see above) is an integrated development environment (IDE) with all mod cons, written in Java, which lets you not only debug an application, but also edit source code, launch compilation, and start other software and tools, all from the same environment. But there's a price to pay for all this luxury: it's hard to install an Eclipse environment without consulting several websites, as Eclipse is a generic IDE stuffed with options (often incomprehensible) to satisfy the needs of all comers. So we're not going to tell you how to go about it here, but direct you to the Yagarto site [5], for example, where there's a very detailed tutorial on the subject. Do note that it's not enough to just install Eclipse, you also need to install the **CDT** (C/C++ Development Tooling) plug-in that converts Eclipse into an IDE for developing software in C/C++ and adds the debugging function to it.

Once Eclipse/CDT is installed, run it and choose a location for the workspace, the place where Eclipse will go and store the project(s).

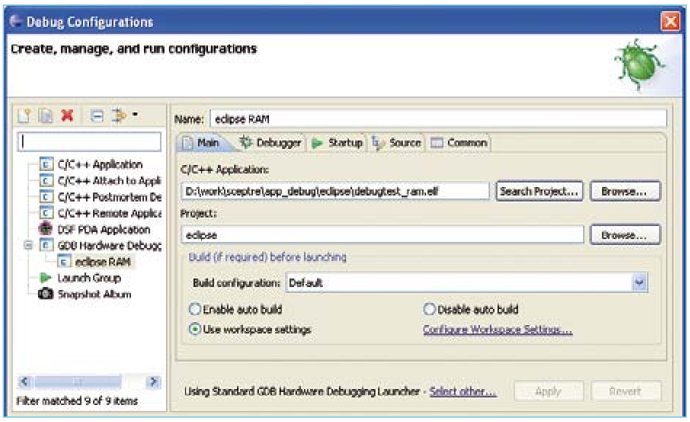


Figure 8. Configuring Eclipse's debugger, first tab.

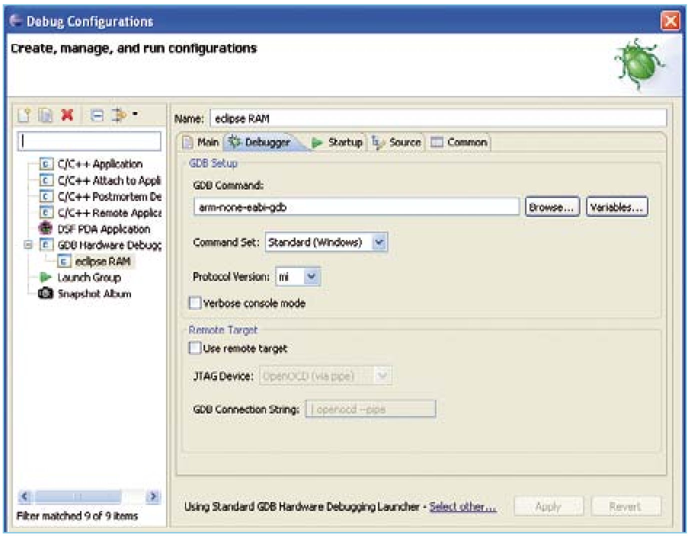


Figure 9. Second tab. Pay attention to the last line Using Standard GDB Hardware Debugging Launcher.

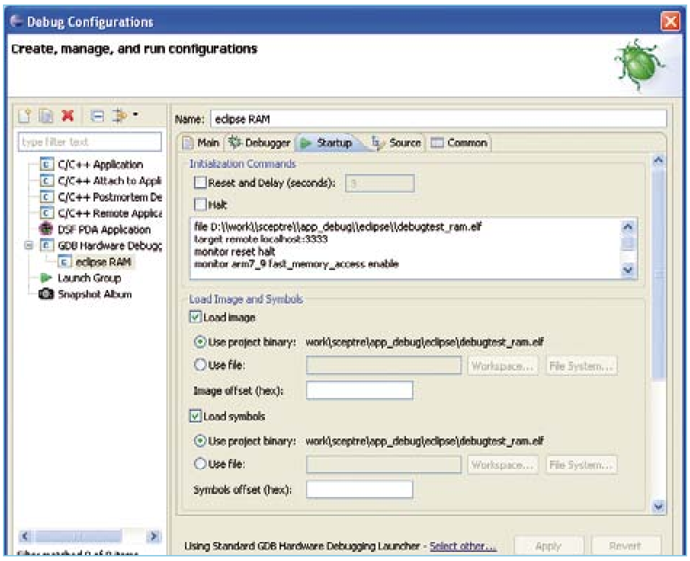


Figure 10. Third tab. The parameters accessible via the scrollbar are blank (by default).

Since we already have the source code for our application, we're going to import everything. For speed, use `File -> New -> Makefile Project with Existing Code`. Then browse your way to your existing project (Existing Code Location) and change the name of the project (Project Name) if necessary. Select `<none>` as Toolchains for Indexer Settings and check the programming language(s) used. Press the `Finish` button, and you're there.
Eclipse offers all sorts of views of a project, called Perspectives. By default, it opens the `Resource` view, but we want the `C/C++` view (`Window -> Open Perspective -> Other`...). You can close the `Resource` view to eliminate one button.
In order to test your project and at the same time the installation of Eclipse, you can try a `make clean` (`Project -> Clean`...), followed by a `make all` (`Project -> Build Project`). Eclipse/CDT presumes that the compilation and GDB tools are present somewhere on your computer. You can indicate the path to GDB, but `make` has to be able to locate it via the 'global' path in Windows. If you use several different compilation chains (WinARM, Yagarto, or others), the simplest thing is to use an adapted `makefile` for each tool chain.

If these two tests were successful, you can move on to debugging. As before, you must run OpenOCD before starting the debugging. It is possible to configure an external tool (`Run -> External Tools -> External Tools Configurations`...) for this purpose, which will let you launch OpenOCD from Eclipse; but just using the Command prompt works too.
Open the `Debug` perspective. The debugger has to be configured before use. You can access debugger configuration via the menu `Run -> Debug Configurations` or via the small arrow to the right of the `Debug` button (with the little beetle). Select GDB Hardware Debugging and click the button `New` (the blank page with a small '+'). There are three tabs to indicate : `Main`, `Debugger` and `Startup`. Refer to **Figures 8, 9, and 10** to find out how to configure your debugger. Those parameters not seen in these figures have kept their default values. Note that the `Startup` tab contains a window where you can enter the commands to be executed when GDB starts up. These are the same commands as those used above and that we have put in our `.gdbinit` file.
Launch the debugger. If this is the first time for the current project, Eclipse won't offer it at the outset and you'll have to start by configuring the debugger by pressing the `Debug` button. The next time, Eclipse will propose the project when you press the button with the beetle. As soon as Eclipse is correctly configured, you get a debugging view like the one in **Figure 11**. Get yourself a big screen, as Eclipse offer lots of windows and you'll need the room to display them all. At the bottom left of this figure, you can see the GDB console where you type in yourself the GDB commands (Eclipse has disabled the `(gdb)` prompt). Don't let yourself be distracted by all the buttons, icons, and tabs that decorate the windows, concentrate first of all on their contents.
You can move around in your program using the F5, F6, and F7 keys and the options offered in the `Run` menu. If you want additional windows, go to `Window -> Show View`.

As in the section on Insight, we're going to end here with a few miscellaneous comments.

- If you can't manage to restart a debugging session, first delete all the breakpoints (`Run -> Remove all Breakpoints`).
- To load the debugging symbols, Eclipse executes GDB's command `symbol-file` instead of the `file` command. As a result, the executable is not loaded by GDB and a subsequent `load` command will fail. So in the case of debugging from the RAM, you'll have to explicitly add the `file` command to the list of start-up commands (or load the program manually in the GDB console). In that case, remember to specify the full pathname, using double slashes '\\' in place of each '\', as in **Figure 10**.

Loading a program into flash memory

When you are the lucky owner of a JTAG probe compatible with OpenOCD, there's nothing to stop you also using it for loading the executable into the processor's or microcontroller board's flash memory. Note that the J-Link Edu won't let you programme the flash memory without an additional licence (unless you can manage to make it work with OpenOCD). Programming by JTAG is especially worthwhile when the application is bulky and the controller's default programming technique is slow, as is the case for the Sceptre's LPC2148, which uses a serial link. Thanks to the USB JTAG probe we used for this article, the Sceptre programming time has been reduced by a factor of nearly ten!

For this to work, you need to configure OpenOCD, for example with the help of OpenOCD's target configuration file, like this:

```
flash bank lpc2148.flash lpc2000 0x0 0x7d00000
lpc2148.cpu lpc2000_v2 12000 calc_checksum
```

The value 12000 corresponds to the processor's clock frequency in kHz. The calc_checksum parameter is required to insert a checksum into the executable, which is obligatory for the LPC2148 to be able to run the program. If your executable already has this checksum, you can delete the parameter and you'll avoid a warning during programming.

From GDB, we now launch the command:

```
(gdb) monitor flash write_image <filename>
```

where `<filename>` indicates the full pathname of the executable file (in BIN, ELF, HEX, etc. format, see the OpenOCD manual) in which every '\' has been replaced by a '/'.

The files used in this article are available from [10].

(100810-I)

Acknowledgement
The ARM-USB-OCD JTAG probe we used for this article was kindly provided by French component supplier Lextronic [9].

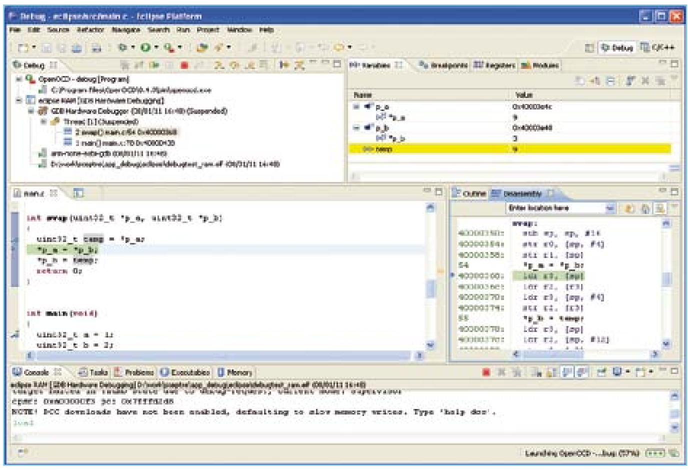
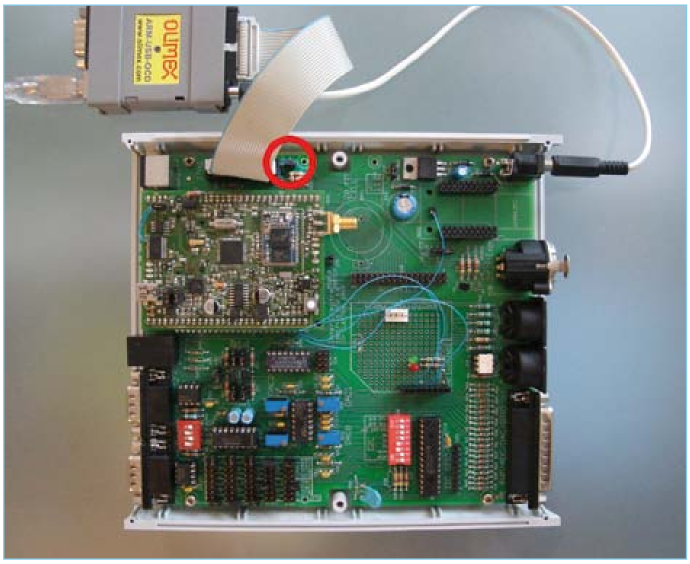


Figure 11. Debugging with all options using Eclipse (Helios 3.6.0 with CDT).

Internet Links

[1] www.elektor.com/090559
[2] http://openocd.berlios.de/web/
[3] www.gnu.org/software/gdb/
[4] www.macraigor.com
[5] www.yagarto.de/howto/yagarto2/index.html
[6] www.freddiechopin.info/index.php/en/articles/34-news/70-openocd-040-instalator-dla-systemu-windows
[7] www.olimex.com/dev/arm-usb-ocd.html
[8] www.segger.com/cms/j-link-edu.html
[9] www.lextronic.fr
[10] www.elektor.com/100810

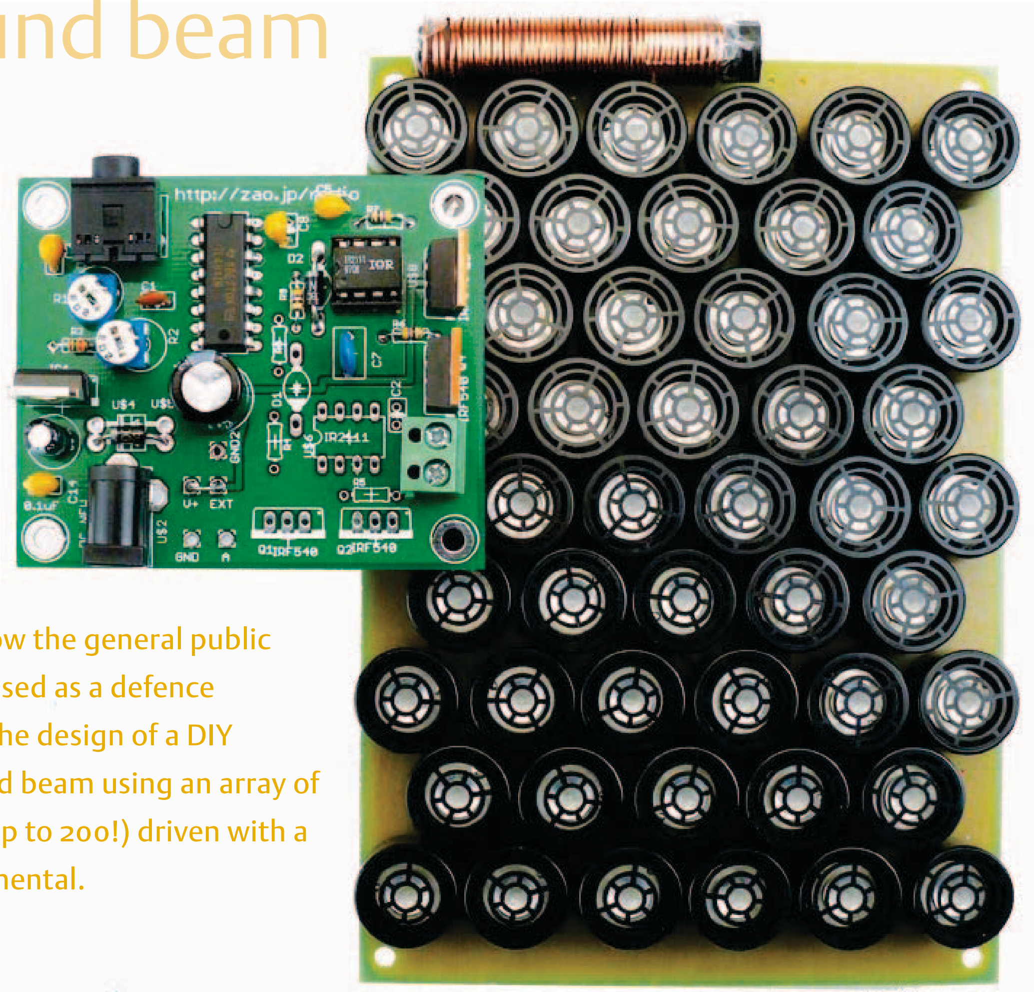


Our Sceptre / InterSceptre system. Remember to fit JP7 on the InterSceptre to enable the JTAG port.

Ultrasonic Directive Speaker

50+ piezo transducers generate audible sound beam

By Kazunori Miura (Japan)



The acronym LRAD sadly and worryingly made it to newspaper stories in connection with Somali pirate attacks on vessels in the Gulf of Aden, and that is how the general public first got word of sound being used as a defence weapon. This article explores the design of a DIY 50-metre range directive sound beam using an array of piezo ultrasonic transducers (up to 200!) driven with a PWM signal, all strictly experimental.

Spurred by the success of their Long Range Acoustic Device® (LRAD) systems, American Technology Corporation changed its name to LRAD Corporation on March 25, 2010 [1]. For non military applications, Audio Spotlight® is a product of Holosonic Research Labs, Inc. [2]. Audio Spotlight produces a very sharp sound beam and has found applications in museums, exhibits and galleries. Those who hear sound from a parametric speaker for the first time are typically surprised and sometimes frightened by the effect. Sounds appear to be heard from extremely nearby, although the person standing right beside you does not hear anything.

Parametric speaker arrays typically employ ultrasonic waves, the same as used in car parking 'radars', distance meters, metal analyzers, etc. However it was not until recently that approaching a real parametric speaker is possible using commonly available components.

Principle of the parametric speaker

A parametric speaker achieves high directivity thanks to the almost line-of-sight propagation of sound waves in the supersonic range. Supersonic is often loosely defined as 'above 20 kHz' because it exceeds the upper frequency limit of human hearing. In practice, 14 kHz is commonly found to be the real limit at least for adults.

So how can humans perceive a supersonic sound wave? Several methods have been devised to convert a supersonic wave into a sound wave you can hear. One method is to passively obtain an audible frequency from two supersonic wave sources with a slightly different frequency. For example, an undulating 1 kHz tone is obtained from two supersonic waves of 40 kHz and 41 kHz. As illustrated in **Figure 1**, where two supersonic waves intersect, a sound within the audible domain is perceived. The disadvantage of this method is that only weak audible sounds are produced, by no means

Note. Readers' Projects are reproduced based on information supplied by the author(s) only.
The use of Elektor style schematics and other illustrations in this article does not imply the project having passed Elektor Labs for replication to verify claimed operation.

enough to stun or incapacitate people like the LRAD.

Other ways of producing an audible sound from supersonic waves include amplitude modulation (AM), double sideband modulation (DSB), single sideband modulation (SSB), frequency modulation (FM) all employ the recently developed parametric speaker system.

Inevitably, a 110 dB+ supersonic wave will be irregular in terms of sound pressure distribution as it propagates through a long air mass, and an audible sound seems to appear by itself owing to these non-linear characteristics. As a result, the audible sound perceived is marked by a fair amount of distortion, which is undesirable for 'narrowcasting' applications like in a museum. Manufacturers typically resort to signal processing using DSPs to reduce distortion to a minimum, often in combination with a highly sophisticated parametric speaker

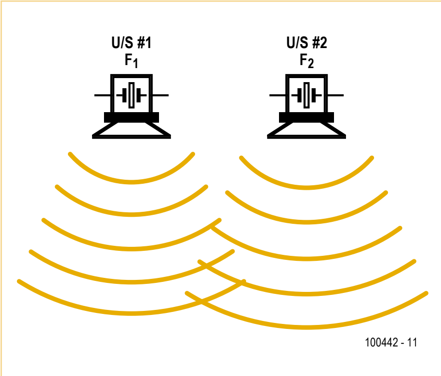


Figure 1. Where ultrasonic waves from sources with frequencies F1 and F2 intersect, audible sounds amounting to F3 = |F1–F2| may be heard.

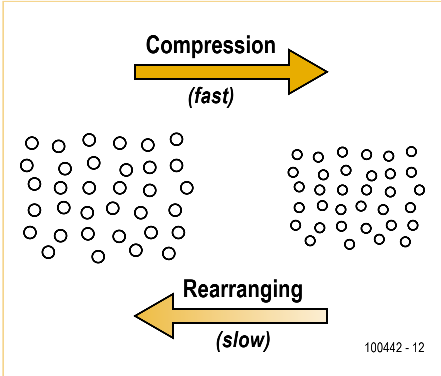


Figure 2. Shock waves come about by air molecules on their way back to their original position colliding with other molecules being compressed at the same time by a sound wave.

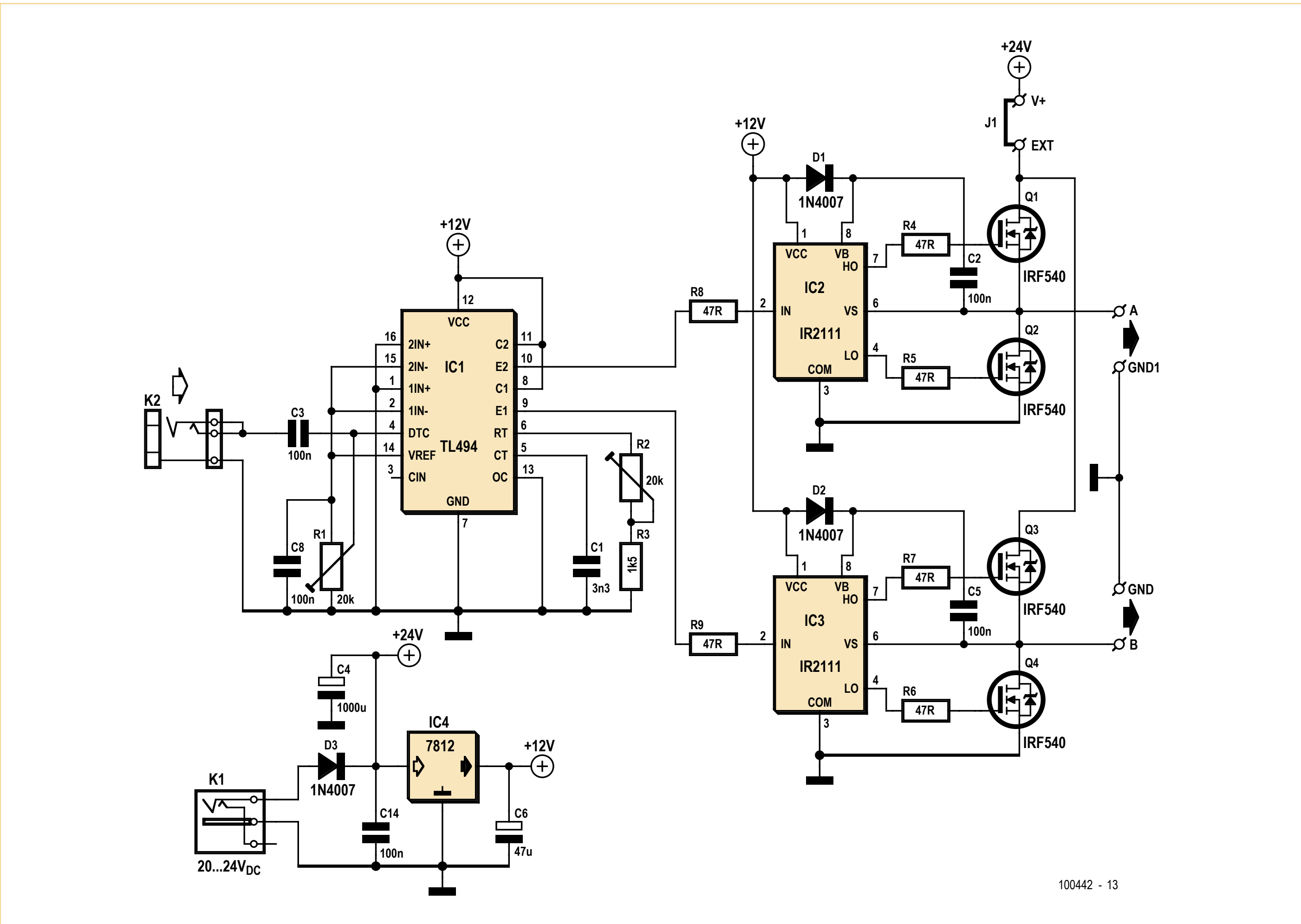


Figure 3. Circuit diagram of the PWM power driver for the ultrasonic parametric speaker unit. The audio input signal is connected to jack socket K2. Channel B is optional.

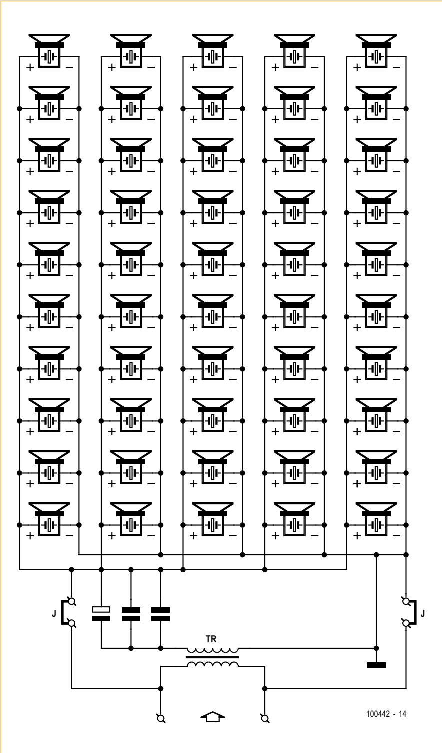


Figure 4. The 'mini' version of the directive loudspeaker consists of 50 ultrasonic piezo transducers connected in parallel and in phase. The step-up transformer and associated resonance caps are optional. The associated PWM driver in its basic form is suitable for up to 200 transducers.

Figure 5. A radio ferrite rod makes a good core for winding a 60-160 µH adjustable inductor that enables the largely capacitive transducers to be peaked at their resonant frequency (usually, 40 kHz).

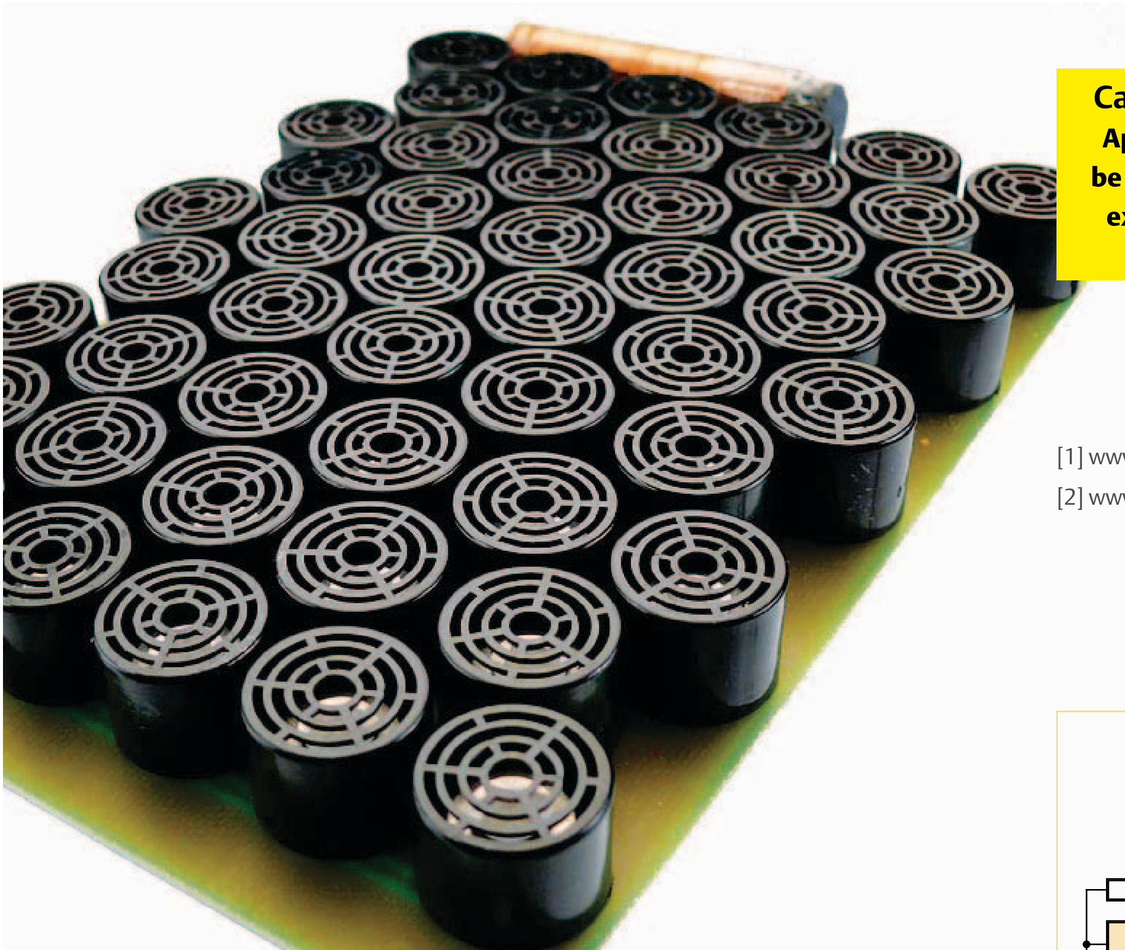
setup.
The 'non linear characteristic' is due to the fact that it takes more time for air molecules to be restored to their original density than to be compressed (**Figure 2**). When the sound pressure is high, and frequency too, a shock wave may be produced by returning air molecules colliding with the ones being compressed. In fact, an audible sound is produced by any molecule not completely 'returning'. When the frequency of the vibration rises, the 'non-linear characteristic" tends to become noticeable by an effect best described as 'air viscosity'.
There is another reason for the high directivity (i.e. small beamwidth) exhibited by a parametric speaker array. The supersonic wave is actually generated by a large number of small loudspeakers called transducers. The piezo-electric transducer is widely used both as a sensor and a transmitting device in car and home automatic systems. The directivity of the piezo transducer by itself is not too high. However, strength is in numbers, meaning the high directivity is due to many small transducers arranged in a plane-like shape. This is essential for making a truly directional speaker unit.

A parametric 2-channel speaker modulator

Double sideband modulation (DSB) is easily implemented using analogue switches. Frequency Modulation (FM) has the same effects basically if you look at the way supersonic sound waves compress air and interact.
The author first attempted a DSB modulator. The result: big sound, lots of distortion and the method might be suitable for a sound beam weapon. Next, a PWM system was built. Looking at the net result, PWM is very similar to FM. The audible sound obtained from PWM is weaker than from DSB, but of a better quality. A PWM modulator may be compared to a class-D amplifier without its low pass filter!
The schematic of a 2-channel PWM modulator is given in **Figure 3**. There are no special components. The TL494 PWM control circuit and the IR2111 half bridge MOSFET driver are used in their standard application circuits. The TL494 has an internal oscillator whose frequency is determined by trimpot R2 and capacitor C1. The basic pulsewidth is adjusted with R1. You need to set up optimum modulation with trimpots R1 and R2.
The audio input signal is connected to K2 (loudspeaker level required, not microphone or line). The board has two outputs, A and B, each driving an array of piezo transducers, optionally through an inductor (see below). Each channel is suitable for up to 200 transducers. The normal supply voltage is 20-24 DVC to K1. The FET stages may be powered by an external supply via the EXT terminal after removing wire link J1. Heatsinks may be required on the IRF540 FETs depending on the supply voltage and the transducers' ratings (up to 60 VDC may be possible).
The U/S speaker schematic is large but unsurprising, see **Figure 4**. It represents one channel and a 'mini' version with just 50 transducers.

Speaker unit and optional coil

There are several type of ultrasonic transducer around. The author used 16 mm diameter devices specified for 40 kHz and 28 kHz. A minimum of 50 transducers is required to make an effective speaker unit. You need more than 100 transducers if you want to the unit to have any sort of range outdoors. All transducers should be carefully distributed to maintain phase. Remember, the wavelength is about 8 mm so a positioning error of 1 mm causes phase errors and loss of SPL.
Ultrasonic transducers are made from piezoelectric ceramic materials. When a voltage is applied to the device, a special type of foil is deformed inside, generating a supersonic sound wave of a specific frequency. Typically, the transducer's sound output reaches 105–120 dB (at 30 cm distance) when a voltage of 10–20 V_{rms} is applied,



correspond-ing to 28–56 V_{pp} and that raises the question if a similarly high supply voltage is needed.

Electrically, an ultrasonic transducer has the properties of a capacitor, which can be made part of a series resonant circuit by putting an inductor in series. Tuning the inductor to about 40 kHz enables the transducer to be driven from a low supply voltage. A step-up transformer as shown in the speaker schematic is another way to get the transducers to operate at resonance.

The resonance frequency f_r may be calculated from

$$f_r = 1 / (2\pi \times L\,C)$$

Each ultrasonic transducer equals about 2,000–3000 pF worth of capacitance. Connecting 50 of them you get roughly 0.1–0.15 µF. To obtain resonance an inductance of about 60– 160 µH is called for, to connected between the driver's A and B outputs and the respective transducer arrays. Fine tuning is required to peak for resonance and the author produced an adjustable inductor from enamelled copper wire and a ferrite rod (**Figure 5**). For a 200-transducer version of the U/S speaker about 55 turns of wire gave best results (60-80 µH).

The ultrasonic transducers need to be checked individually to determine their polarity (phase). This may be done using an oscillator and a 2-channel oscilloscope as illustrated in **Figure 6**. One U/S device is connected to an oscillator (or generator) supplying a 40 kHz source signal that's also fed to one channel of the oscilloscope. The 'receiver' device gets connected to scope channel 2. Now you can view the signal and the timing at a glance (**Figure 7**).

Fun with the parametric speaker

It should be reiterated here that the project is experimental and intended to promote your own experiments. Connect the audio sound source through 3.5 mm jack socket K2, and connect the power supply to K1. You can probably hear a weak sound from the transducer array. Carefully adjust R1 and R2 for optimum sound quality. Check if the sound beam is anything like directive — it should be, even when using one channel (A or B). The author has tentatively indicated a range of about 50 meters (150 ft.) for a 200-transducer (!) system.

(100442)

Caution — Health Hazard. Appropriate measures must be taken to prevent long term exposure to high ultrasonic sound levels.

[1] www.lradx.com

[2] www.holosonics.com

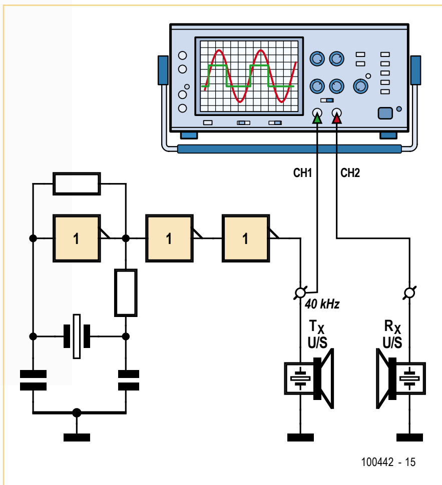


Figure 6. Test arrangement to establish the polarity of each and every piezo transducer used for the parametric array.

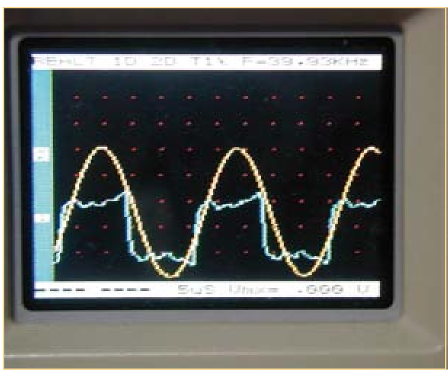


Figure 7. Scope image obtained with correct polarisation of the receiving U/S transducer.

PSoC Evaluation Kits

Cypress Semiconductor offers a wide range of development boards and accessories for its PSoC devices. PSoC evaluation boards are also available from several other sources. Here we present a few interesting boards and briefly describe their features.

By Harry Baggen (Elektor Netherlands Editorial)

The PSoC family from Cypress Semiconductor is based on a general-purpose concept with both analogue and digital programmable functional modules integrated in a single IC, along with a processor core. In contrast to many other SoCs, which are generally targeted to a specific application area (such as a radio or television receiver IC or a multimedia IC), Cypress's PSoCs can be used for virtually any purpose, as long as the computing power and the available analogue and digital modules are sufficient for the intended application.

With the aid of a user-friendly graphic programming environment called 'PSoC Creator', designers can use the modules to quickly configure the desired functionality. There are three families, each based on its own processor core. The PSoC 1 family has an M8C core with a capacity of 4 MIPS, the PSoC 3 family features an 8051 core with a capacity of 33 MIPS, and the most powerful family, PSoC 5, utilises an ARM core with a capacity of 100 MIPS.

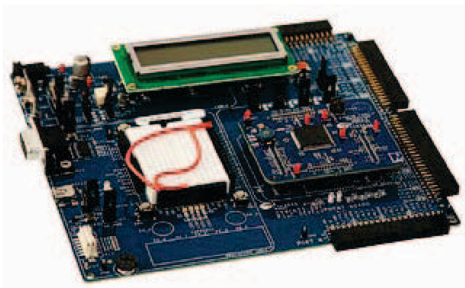
From the wide range of products available from Cypress, we selected a few kits that appear attractive for exploring the world of PSoCs.

Starter kits

The **PSoC 3** and **PSoC 5 FirstTouch Starter Kits** (CY8CKIT-003 [1] and CY8CKIT-014 [2]) are a good choice for initial evaluation of the capabilities of PSoC devices. They are inexpensive (around $50 each), and they are equipped with a variety of sensors and indicators on the board, so you can immediately try out all sorts of example projects. The small boards incorporate a proximity sensor, a thermistor, an accelerometer, several touch surfaces for a CapSense slider and eight LEDs, and of course a good handful of I/O ports. The only difference between the two kits is the PSoC device soldered to the board. The kits come with a USB cable, a 9-V battery and a CD holding the necessary software (PSoC Creator and PSoC Programmer) and various example projects. A separate programming device is not necessary.

General-purpose development kit

If you're looking for a general-purpose development board for all of the PSoC families, you quickly end up with the **PSoC Development Kit** (CY8CKIT-001 [3]). It costs a good deal more than the starter kits ($250), but for that price you get a sizeable board with quite a few features. Furthermore, it comes with three processor modules: one each from the CY8C28, CY8C38

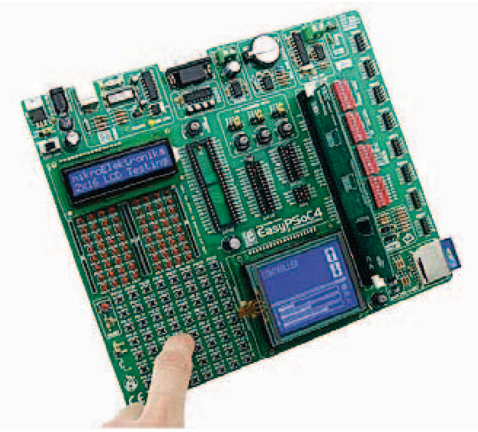


and CY8C55 families. Additional processor modules can be bought separately. This kit also includes a MiniProg3 debugger/programmer. All in all, this is very attractive kit. The evaluation board incorporates an LCD module (2×16), RS232 and USB interfaces, a breadboard area, a potentiometer, several pushbuttons, LEDs, and a CapSense touch area. All I/O lines are fed out to four large connectors. Cypress also offers a variety of expansion modules for this board.

iPod and iPhone app development

Cypress's range of development kits includes diverse kits and expansion boards for specific applications, such as a voltme-

ter kit and an LCD driver kit. The one that grabbed our attention was the **PSoC Expansion Board Kit For iPhone & iPod Accessories** (CY8CKIT-023 [4]), which is designed to be used with the general-purpose development kit described above. This kit enables you to develop all sorts of new accessories for iPods or iPhones, in combination with a PSoC device. You can use an app developed by Cypress and the included example project to communicate with the connected hardware from the iPod or iPhone.
Unfortunately, this module is not available



to everybody; it is restricted to licence holders for the Apple MFi program.

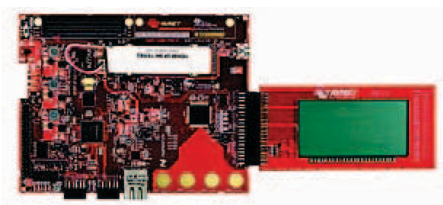
Alternative boards

Several other companies have also developed evaluation boards for PSoC devices. For example, MikroElektronika offers a large development system called **EasyPSoC4 Development System** for $129 [5], which boasts a wide scope of features. The CY8C27643 PSoC is located on a separate, interchangeable MCU board, which allows other types of PSoC devices to be used. The board has numerous interfaces, LEDs, buttons, potentiometers and DIP switches for experimenting with designs. It also includes an SD card connector and a real-time clock, well as an on-board programmer module. Furthermore, there is room to fit a LCD module for text display (2×16) and a graphic LCD module, optionally with a touch screen (all available as separate accessories). MikroElektronika supplies its own software for the programmer module (PSoCprog2). Fifteen simple example applications are available. Each of them focuses on a specific task, such as driving a LED, performing analogue to digital conversion, or driving a stepper motor.

MikroElektronika also offers a separate PSoC programmer with USB interface for $ 89. It contains the same programmer module as the one fitted on the board, but packaged in a separate enclosure.

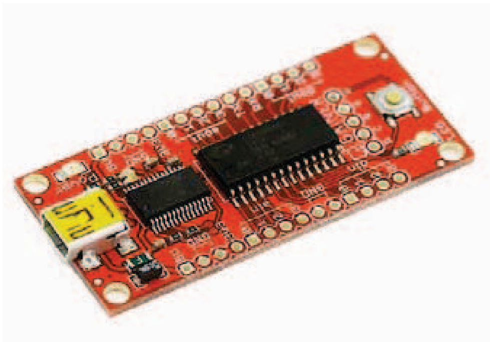
Avnet offers an unusual combination: in cooperation with several IC manufacturers, Avnet has developed a board incorporating a Xilinx Spartan-6 FPGA together with a Cypress PSoC, which goes by the name **Spartan-6 LX16 Evaluation Kit** (part number AES-S6EV-LX16-G [6]). The battery management and voltage regulation elements of this board originate from Texas Instruments.

In addition to the FPGA and PSoC devices, the board has 64 MB of SDRAM, 16 MB of multi-I/O SPI flash memory, Ethernet PHY, a JTAG interface, four CapSense switches, four LEDs, a USB UART, and a lithium-ion battery that powers the entire board. The



kit also includes a display board that can be connected to the evaluation board. The kit price is $225.

Sparkfun offers a very simple evaluation board called **Gainer** [7]. It was developed by Shigeru Kobayashi in cooperation with a team of developers at Gainer.cc [8]. The board is fitted with a CY8C29466, which makes it very suitable for experimenting with A/D and D/A conversion. Device programming via the on-board USB interface is easy thanks to the pre-installed boot loader.



A special feature of this board is that it is suitable for a variety of programming languages, including Flash, Max/MSP and Processing. The price is $35.

All in all, there's plenty to choose from and most of the boards are very affordable, which makes them suitable not only for professionals, but also for hobbyists who want to gain hands-on experience in designing with PSoC devices.

(100856-I)

Internet Links

[1] www.cypress.com/?rID=38235

[2] www.cypress.com/?rID=43674

[3] www.cypress.com/?rID=37464

[4] www.cypress.com/?rID=40218

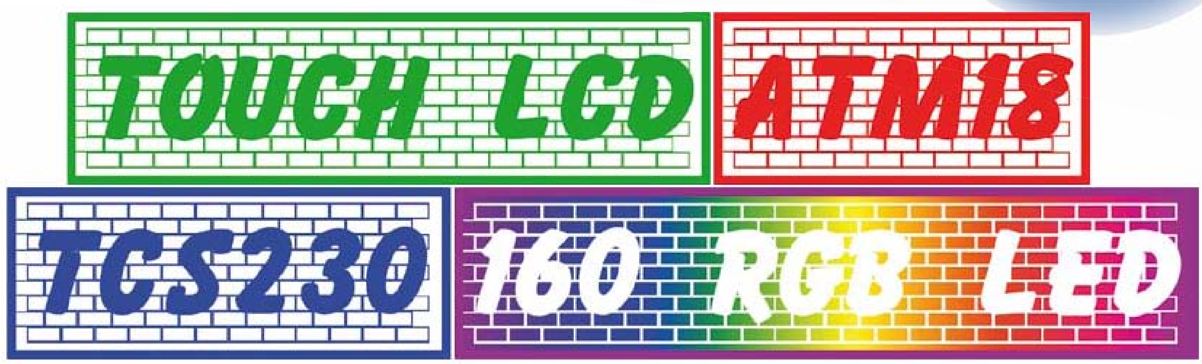
[5] www.easypsoc.com

[6] https://avnetexpress.avnet.com (search for 'AES-S6EV-LX16-G')

[7] www.sparkfun.com/products/8480

[8] http://gainer.cc

A String of 160 RGB LEDs
A colourful display



By Grégory Ester (France)

Colours are not just mood indicators, apparently they can also lift or dampen your general mindset to a degree. If you want to attract attention during some event, decorate a shop window, or add a bit of ambience to a party, you can do all this and more with this high-tech light string — and what's more, it's self-adhesive.
Pale LEDs that just flash, or don't, are out — make way for a new generation of lighting effects!

This project will let you control a string of RGB LEDs using either a touch screen or a colour detector. In the first case, you can use your finger or a stylus; the second will require pieces of red, green, and blue card... Sounds interesting? Then to your boards (ATM18), you have the green light to start wiring!

Block diagram for a colourful connection

This project uses the Elektor ATM18 board [1], a colour detector [2], a 128 × 64 pixel graphical LCD touchscreen [3], the Elektor 2-wire display [4] and a string of RGB LEDs [5]. Provision has been made for two configurations for controlling the string: via the touch screen (without the colour detector or the 2-wire display) or via the colour detector and using the 2-wire display (without the touch screen). **Figure 1** shows how to connect up all the modules so as to test both operating modes. So you can load one or other of the two bits of firmware 74_DOGM_HL1606.hex or 75_HL1606_TCS230.hex without having to modify the wiring. **Table 1** shows you which peripherals are active, depending on the program loaded. The firmware and source code files are available free from [3].

But how can we control these lights?

The string [6] is manufactured by Astro-Fly Lighting Technology in Hong-Kong and is readily available on the Internet ([5], eBay, or search for "HL1606 5050"). It comprises a number of 6.2 cm (2.45 inch) long segments. Each segment includes two RGB LEDs that can be driven independently via an SPI synchronous serial link. The segments also have a self-adhesive strip, so it's easy to stick the string onto any surface.
To be able to control the string in your own fashion and create your own effects, it's necessary to take a closer look at how the whole thing works. So let's light up just the first two blue LEDs and look at the resulting signals on the oscilloscope. But just before we do, let's see how to put together the burst of bits to be transmitted over the serial link in synchronization with the clock signal.

Table 1. Two operating modes.

Configuration	Peripherals used			
Colour detector (75_HL1606_TCS230.hex)	TCS230	ATM18	2-wireLCD	RGB string
Graphical LCD touch screen (74_DOGM_HL1606.hex)	LCD Touch screen	ATM18	RGB string	

Table 2. Summary of the 16 bits for configuring one segment of the string.

First RGB LED								Second RGB LED							
D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
latch	fade speed	blue		red		green		latch	fade speed	blue		red		green	

Each RGB LED is driven by eight bits. Each of its three colour cells (red, green, or blue) is controlled by two bits, allowing four combinations:

- **00**: LED out
- **01**: LED lit, brightness always at maximum
- **10**: the brightness increases gradually from minimum to maximum, faster or slower, depending on the signal S_I
- **11**: the brightness reduces gradually from maximum to minimum, faster or slower, depending on the signal S_I

The 'fade speed' function makes it possible set the speed at which the brightness of each LED fades up or down. One bit is set aside for this:
- **0**: slow fade up/down
- **1**: fast fade up/down

Adding a validation bit (Latch), to let us confirm the whole thing or not, completes our 8-bit byte. If this bit is zero, all the commands are ignored — this is a way of inserting transparent configurations that have no effect on the LED.

Hence it takes two bytes to control the segment's two RGB LEDs. **Table 2** sums up the arrangement of the bits in the two bytes. The MSB (D16) will be transmitted first over the serial link. Once the pair of bytes has been transmitted, the transmission has to be confirmed. Pin L_I receives the confirmation pulse.
A small extract of a few lines from the program written in BASCOM-AVR will illustrate how we make the two blue LEDs in the first segment light at full brightness.

```
Color_array(4) = &B10010000   'BLUE
For X = 1 To 2
```

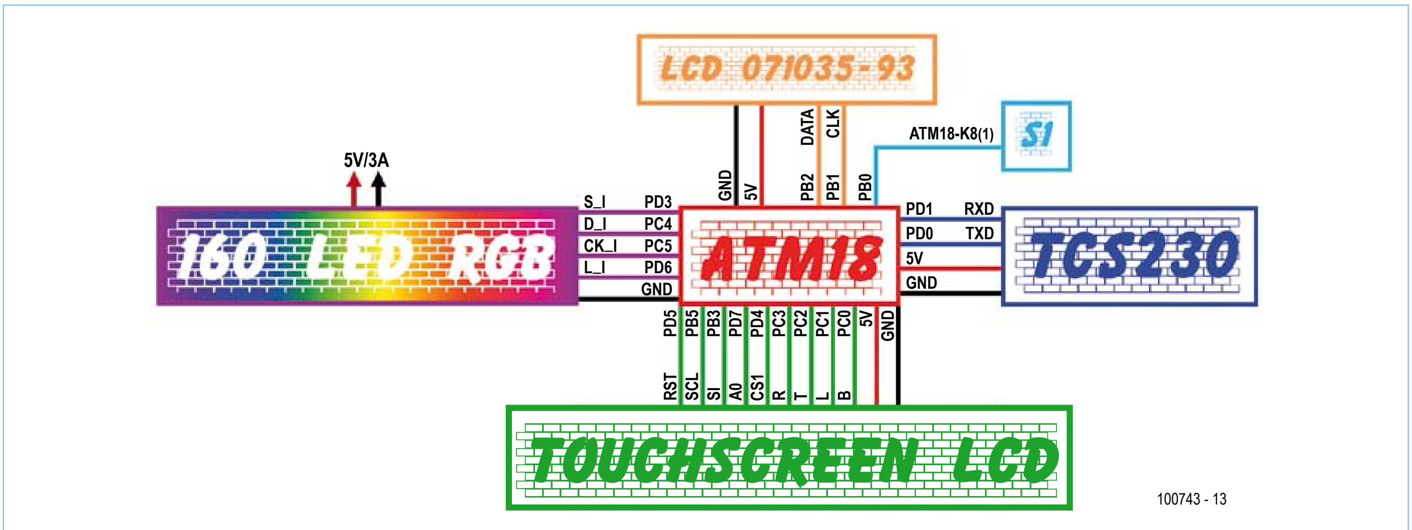


Figure 1. Block diagram of the LED string controller.

Elektor Products & Services

- PCB: Elektor # 100743-1
- PCB design (free download): # 100743-1.pdf
- Firmware & source code (free download): # 100743-11.zip
- ATM18 controller board: Elektor # 071035-91
- ATM18 piggy-back board: Elektor # 071035-92
- 2-wire display: Elektor # 071035-93
- Hyperlinks in article

All items accessible through www.elektor.com/100743

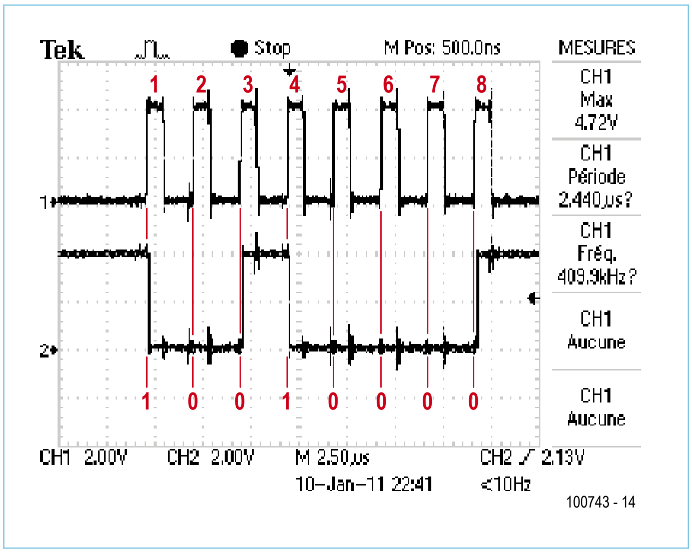


Figure 2. One byte transmitted in sync with the clock.

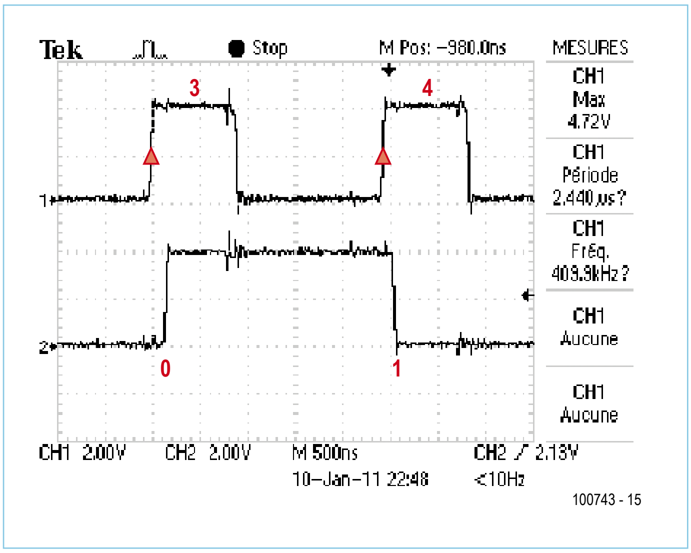


Figure 3. Zoomed detail of the synchronisation.

```
  Spiout Color_array(4) , 1
Next X
Set L
Waitus 3
Reset L
```

Figure 2 shows a graphical display of the first byte; the data is regarded as valid during the rising edge of the clock signal (**Figure 3**).

In **Figure 4**, the LEDs are shining brightly... OK, I admit you can't really see that in this photo, but I can promise you that in reality, they're shining out in topaz blue! By using the method described in the article "LEDs and Illumination – How much light does that LED give?" [7], I obtain an *f*-stop/exposure time combination of *f*5,6/(2 to 4) with sensitivity set at 100 ISO; this corresponds to a lighting level of around 20 to 40 lux in my little workshop.

A transition to create an effect

Now we're going to send 160 bytes that will enable us to obtain a dissolve from green to red on all 160 LEDs simultaneously. The transition will last 1.4 s — with the precision due to Timer0.

Now, going from green through orange to red is all very pretty — but how does it work? The answer lies in the program extract below, which is analysed below.

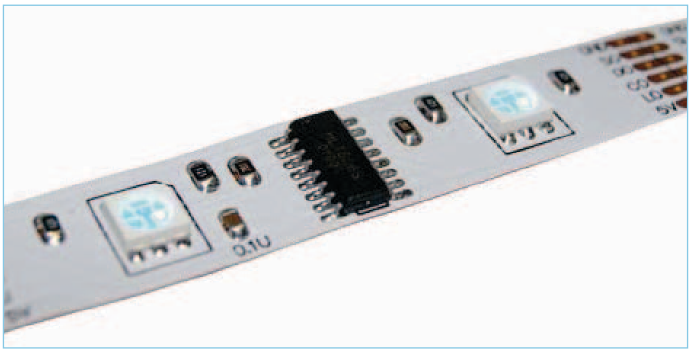


Figure 4. Just one of the 80 segments!

```
Config Timer0 = Timer , Prescale = 1024
On Ovf0 Timer0_isr
...
Color_array(28) = &B10001011
...
Fade_speed = 170
Launch_fade
For x = 1 To 160
  Spiout Color_array(28) , 1
Next x
Latch
Wait 5
...
Timer0_isr:
Timer0 = Fade_speed
Toggle S_i
Return
...
Sub Launch_fade
Enable Timer0
End Sub
```

- 16 MHz / 1024 = 15,625 Hz; Timer0 is enabled at a rate of 15,625 Hz.
- (1/15625) × 2^8 = 16.384 ms; this is an 8-bit counter, so it overflows every 16.384 ms. This is the rate at which the interrupt routine should be executed. Only 'should', since in fact Timer0 is preset to 170 by the variable 'Fade_speed'.
- (1/15625) × (256–170) = 5.5 ms; so S_I is going to change state every 5.5 ms (Toggle S_i), thereby generating the pulses needed to advance the colour dissolve. So one pulse is generated every 11 ms.
- Color_array(28) = &B10001011; the brightness of the red is set at minimum (D12 = D4 = 1, D11 = D3 = 0) while the green (D10 = D2 = 1, D9 = D1 = 1) starts out at maximum. Each pulse will gradually increase the red light and similarly reduce the green. After 128 pulses, the string will be completely red.
- 128 × 11 ms = 1.4 s; so the dissolve will have lasted 1.4 s. If bits D15 and D7 had been set to 1, the transition would have lasted 1.4/2 = 0.7 s.

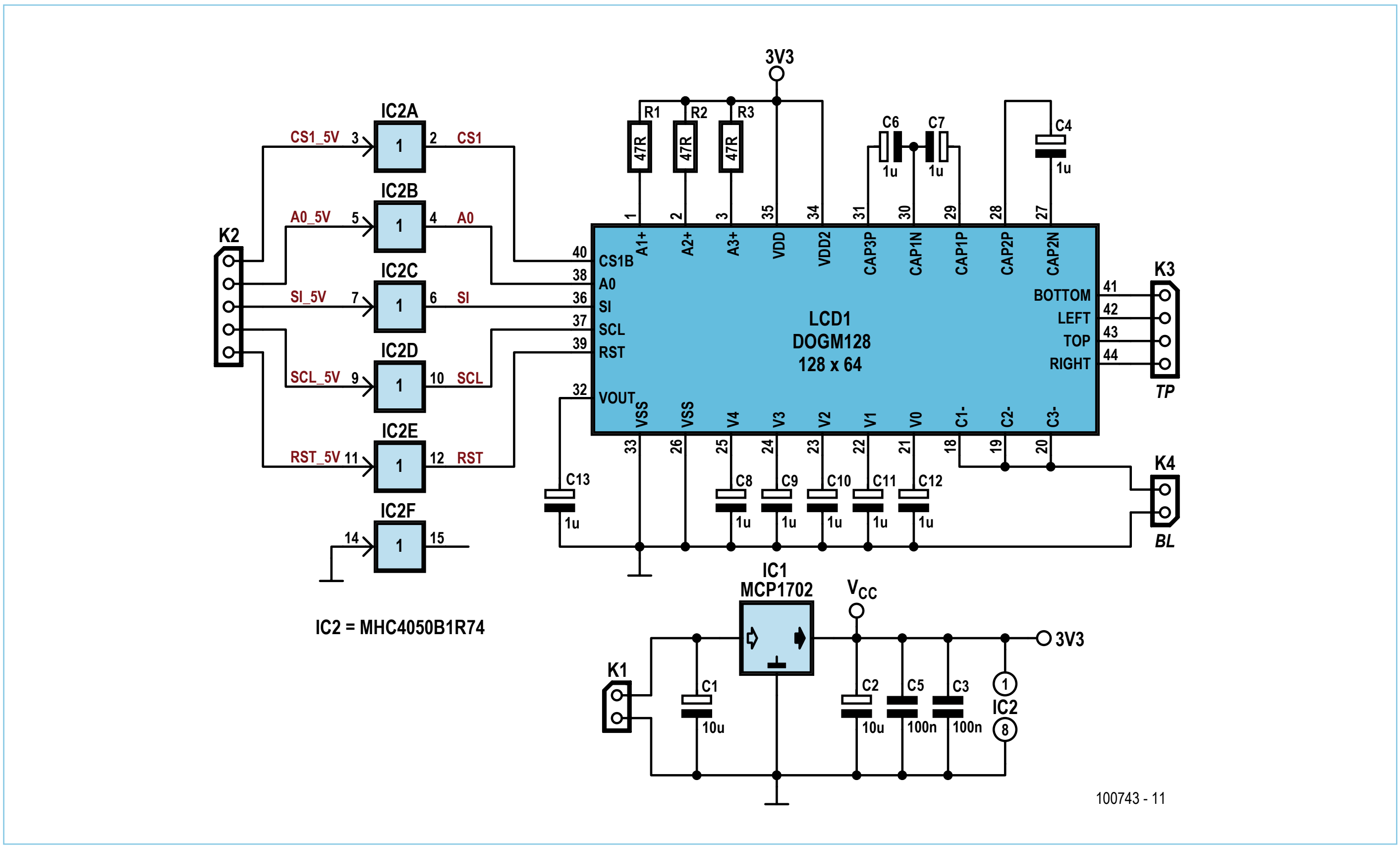


Figure 5. Circuit of the graphical touch screen.

The effects are countless and magnificent, believe me! So we can select some of them, we're going to use two solutions — two different man/machine interfaces.

Beware of the dog

The DOGM128W-6 128 × 64 pixel graphic LCD using the FSTN technique is manufactured by Electronic Assembly [8]. In this application, we're using it sandwiched between the LED backlight module (several colours are available) and the touch panel. This sandwich is not a ready-made module, but an Elektor PCB. The extremely simple circuit diagram is shown in **Figure 5**. The board is available from [3] (Elektor # 100743-1).

This display offers excellent contrast and very interesting software implementation, as BASCOM-AVR has a library for controlling it.

```
Config Graphlcd = 128 * 64eadogm, Cs1 =
Portd.4, A0 = Portd.7, Si = Portb.3, Sclk =
Portb.5, Rst = Portd.5
```

To create your graphical interface, nothing could be simpler. All you have to do is design it in BMP format and then use BASCOM-AVR's built-in graphic converter (**Figure 6**) to create the file with the same name, but with an extension recognized by the compiler — here 'background_1.bgf'.

```
Showpic 1 , 1 , Picture1
Lcdat 6 , 50 , "WELCOME!"
Wait 1
Lcdat 6 , 50 , "          "
Lcdat 6 , 50 , "PROGRAM:"
...
Picture1:
$bgf "background_1.bgf"
```

The touch pad (**Figure 7**) can be likened to two potentiometers whose values are read as follows:

- Apply a voltage of 5 V between the 'TOP' and 'BOTTOM' pins, and between 'LEFT' and 'RIGHT' you will read a voltage proportional to the horizontal displacement of your stylus (Y position).
- Next apply a voltage between 'LEFT' and 'RIGHT', and this time reading between the 'TOP' and 'BOTTOM' pins gives us the X

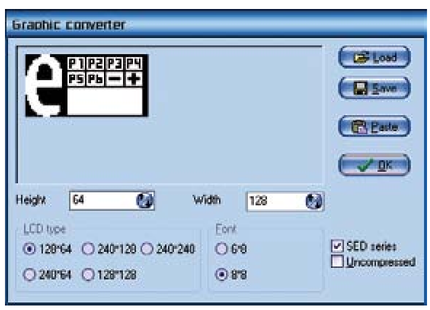


Figure 6. 'Graphic Converter' under BASCOM-AVR.

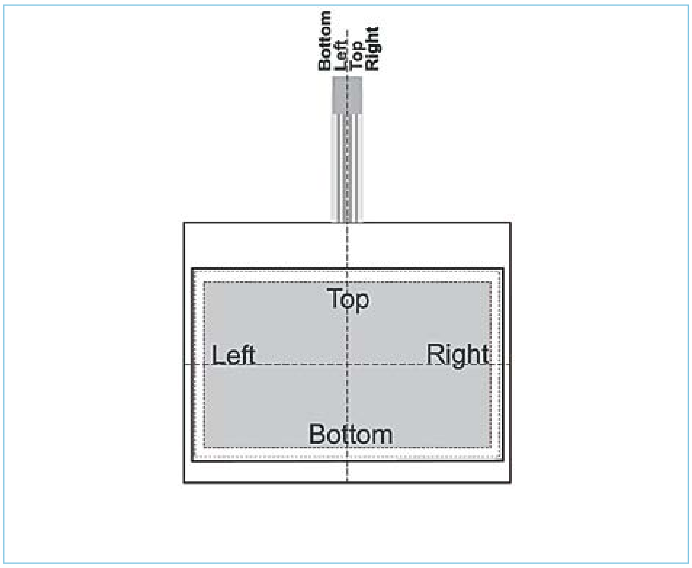


Figure 7. The touch panel.

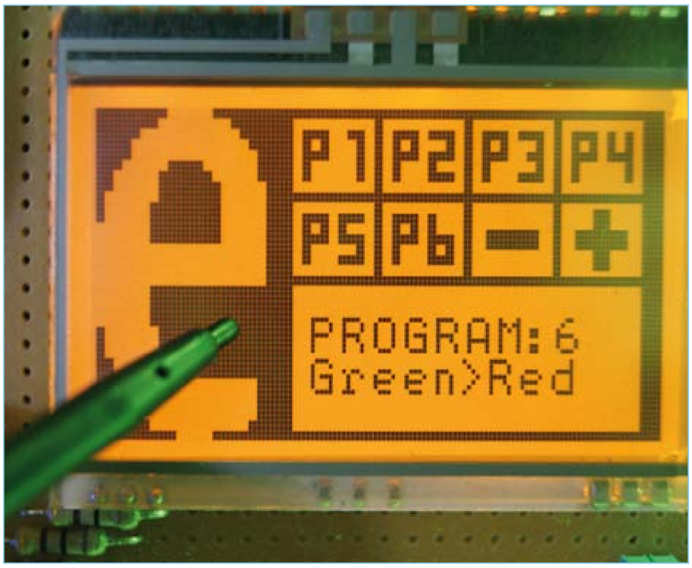


Figure 8. EALED55x46A + EADOGM128W-6 + EATOUCH128-1.

position proportional to the displacement of the stylus from top to bottom.

Hence X and Y will be the co-ordinates of the point where the stylus contacts the touch pad. Nothing could be simpler to handle, using our microcontroller's built-in ADC.
In this way, our program recognizes nine zones (**Figure 8**). Six of these correspond to six programmes from P1 to P6 that are directly accessible. Pressing with your finger or a stylus in the relevant zone displays the name of the programme. You can also move around from 1 to 10 using the + or − 'buttons'. Programme 7, for example, can only be accessed this way. Confirm by pressing the 'e' for 'Elektor'.

A digital retina

The TCS230 module [2] incorporates the TAOS optical sensor of the same name, which lets us measure colours with wavelengths from 350–750 nm for light levels of at least 100 lux. We have three possibilities for recovering digital or analogue information that represents the colours present in front of the 6 mm diameter lens:

- A variable linear voltage that reflects the red, green, or blue values;
- An SPI link for reading digital information in the form of bytes;
- An asynchronous serial link and a syntax in the form of ASCII characters. In this instance, we've opted to communicate using the UART that is physically present in our ATmega88.

After loading the firmware '75_HL1606_TCS230.hex', you can calibrate the sensor by pressing S1 at start-up or after a reset, with the sensor's lens aimed at a white object:

```
If S1 = 0 Then Print "$sure wb" ; Chr(_cr) ;
Chr(_lf) ; 'Start White Balance
```

The surface viewed at this moment will henceforth become the sole reference so that the module will then be able to break down colours viewed into the three colours red, green, and blue.

The 2-wire display

Once the white balance has been performed (**Figure 9**), the 2-wire display ([4], Elektor # 071035-93) displays, in rotation, three bytes in decimal (0–255) representing the colours measured. Here are the five operating modes the program offers, depending on the colour seen by the lens:

– Red card: the string lights up red
– Green card: the string lights up green
– Blue card: the string lights up blue
– White: the string lights up with all lights blazing
– Black (the cap is fitted over the lens): lights out!

An example for green (**Figure 10**): unless your card is a perfect green, in which case there will be no ambiguity in detecting it, it will be necessary to adjust the limits so as to filter out the other colours:

```
If Var_green > 150 And Var_blue < 140 And Var_
```

Table 3. The effects available. Programmes 8–10 are free, the effects they produce will depend on your own imagination.

Programme	Description
1 (P1)	Red snake followed by a blue snake
2 (P2)	« Crazy snake »
3 (P3)	Red or green? Green or red?
4 (P4)	Succession of colours
5 (P5)	A symphony of colours!
6 (P6)	Green to red via orange — are you following?
7 (P7)	Progressively towards blue
8 (P8)	Free
9 (P9)	Free
10 (P10)	Free

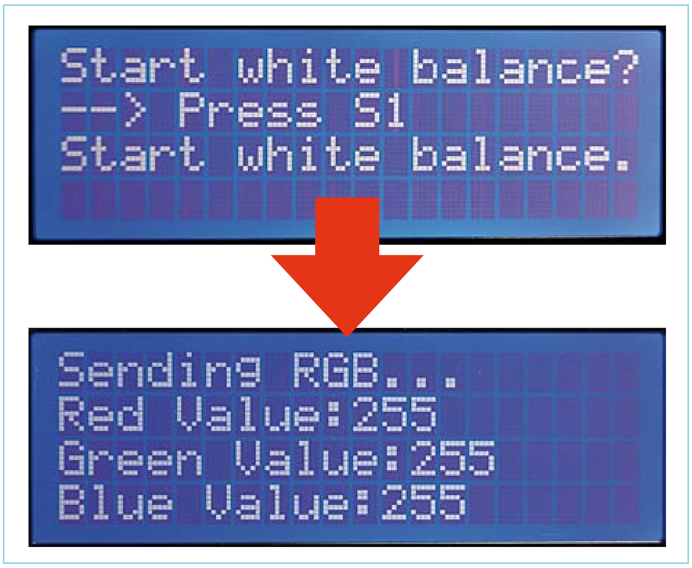


Figure 9. White balance.

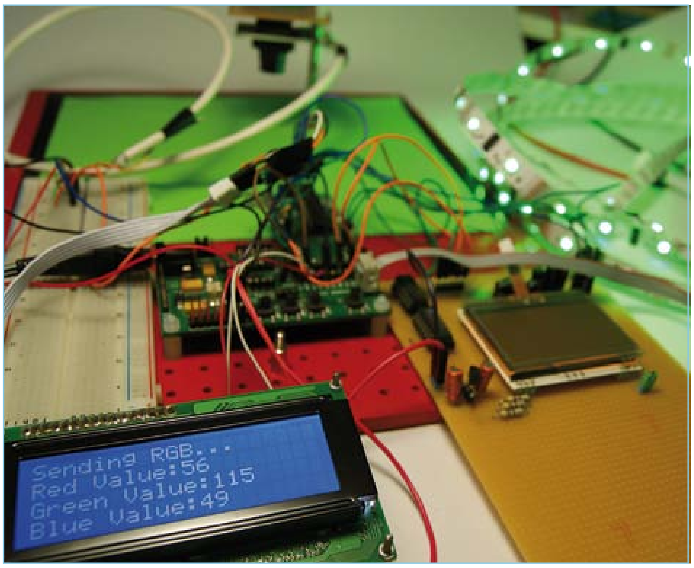


Figure 10. Rather green!

```
red < 140 Then
  For X = 1 To 160
    Spiout Color_array(2) , 1
  Next X
  Latch
  Wait 1
End If
```

If in the rarest of cases the string itself fails to draw attention, this novel way of controlling it will, albeit from an different audience.

(100743)

Internet Links

[1] www.elektor.com/atm18

[2] www.sureelectronics.net/goods.php?id=959

[3] www.elektor.com/100743

[4] www.elektor.com/071035

[5] www.ledlight-lamp.com/cp/html/?310.html or www.lextronic.fr/P19361-flexible-a-leds-RVB.html

[6] www.ledlight-lamp.com/cp/html/?310.html

[7] www.elektor.com/100621

[8] www.lcd-module.com/products/dog.html

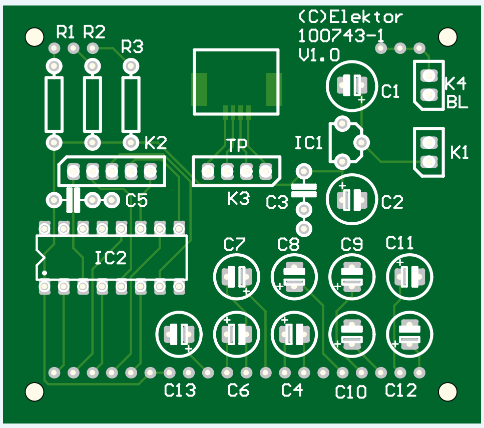
COMPONENT LIST

Resistors
R1, R2, R3 = 47 Ω

Capacitors
C1,C2 = 10µF 25V radial, lead pitch 2.5mm
C3,C5 = 100nF, lead pitch 5mm or 7.5mm
C4,C6–C13 = 1µF 16V radial, lead pitch 2.5mm

Semiconductors
IC1 = MCP1702-3302E/TO (TO-92)
IC2 = 74HC4050N (DIP-16)

Miscellaneous
LCD1 = LCD, graphic, Electronic Assembly type EA DOGM128X-6
Touchscreen, Electronic Assembly type EA TOUCH128-1
K5 = ZIF connector for touchscreen, Electronic Assembly type EA WF100-04S
Backlight module = Electronic Assembly type EA LED55x31-W (W=white, other colours availabe, see LCD1 datasheet [8])
K1 = 2-way pinheader socket, lead pitch 0.1 in. (2.54mm)
K2 = 5-way pinheader socket, lead pitch 0.1 in. (2.54mm)
K3 = 4-way pinheader socket, lead pitch 0.1 in. (2.54mm)
K4 = wire link or 2-pin pinheader with jumper, lead pitch 0.1 in. (2.54mm)
Jumper or switch for K4.
PCB # 100743-1, see [3]

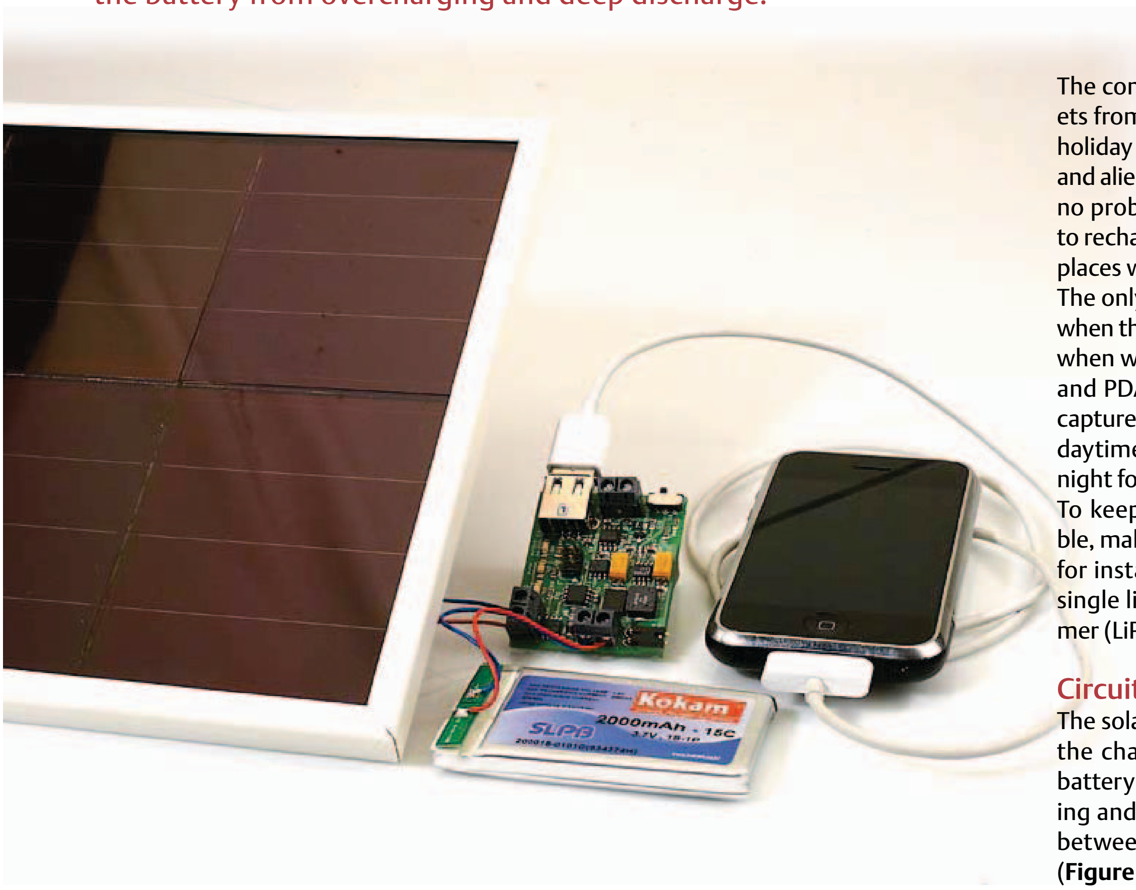


Solar Charger

Portable energy for people on the move

by Martin Kiel (Germany)

With renewable energy high on the agenda, this little project will appeal to everyone who would feel better charging their mobile or PDA from solar sources. A lithium-ion cell stores the sun's energy in between charging sessions. Smart circuitry in the solar charger monitors the battery voltage and protects the battery from overcharging and deep discharge.



Features

- Internal Lithium-Ion battery for storing solar energy
- Versatile: your choice of batter and solar panel sizes
- Direct charging/operation of USB devices.
- Two switchable charging outlets:
  - constant voltage output (5 V, max. 500mA)
  - constant current (max. 150 mA)
- Battery management for internal battery:
  - Overload protection
  - Under-voltage protection with hysteresis regulator and load dumping
- Watchdog dropout protection
- Compact dimensions
- Complete firmware with source code downloadable from the Elektor website

The concept of recharging portable gadgets from the sun is by no means new [1]. On holiday any undetermined AC grid voltages and alien-looking power outlets would pose no problem, whilst we would also be able to recharge these essential gadgets even in places where there is no mains electricity. The only disadvantage is that the daytime, when the sun in question is available, is also when we most need to use mobile phones and PDAs. So the aim of this project is to capture and store those sun rays during the daytime so that we can put them to use at night for charging our gadgets.
To keep this circuit as portable as possible, making it useful on a long walking tour for instance, the energy store chosen is a single lithium-ion cell of the lithium-polymer (LiPo) type.

Circuit

The solar charger consists of two modules: the charge regulator for the lithium-ion battery and a DC-to-DC converter for raising and stabilising the battery voltage (of between 3.0 and 4.15 V) to a higher value (**Figure 1**).
The heart of the circuit in **Figure 2** is an ATtiny13 microcontroller from Atmel, which monitors the battery voltage and controls the output of the solar cells. The solar charge regulator is arranged as a shunt regulator, which short circuits the solar cell if the battery voltage gets too high. As solar cells are short circuit-proof this arrangement does not pose any problems and offers the bonus that the current flowing through the feeder leads is not cut off abruptly. An economic advantage of this scheme is that T1 can switch without the need for an additional driver stage. A MOSFET IRF7413 is our choice for T1, which is definitely a bit of overkill for this applica-

tion (I_{Dmax} = 13 A) but assures reliable activation by TTL level voltages without any problems. Acceptable activation is possible even at a reduced battery voltage of 4.1 V.

The charge voltage reaches the battery from the solar panel via diode D1. The choice of this diode comes down to the solar panel used and the prototype used a 1N4007. However it is better, based on the voltage produced by the solar panel, to use a Schottky diode (e.g. BAT85), since these exhibit a lower voltage drop, raising the overall efficiency of the circuit.

The battery in turn feeds the boost converter, which is equipped with an LT1302 from Linear Technology. The buffer choke L1 used has an inductance of 10 µH.

In standby mode and with the DC-to-DC converter enabled, the circuit draws barely 30 mA of current. Despite this you have the option of removing jumper link JP1 to disable the DC-to-DC converter altogether.

On the ground side the boost converter (and thus the output) is separable by a further IRF7413 MOSFET, meaning that the battery can be disconnected from the output for the sake of deep discharge protection (load dumping).

Since mobile handsets do not all exhibit the same charging characteristics, the boost

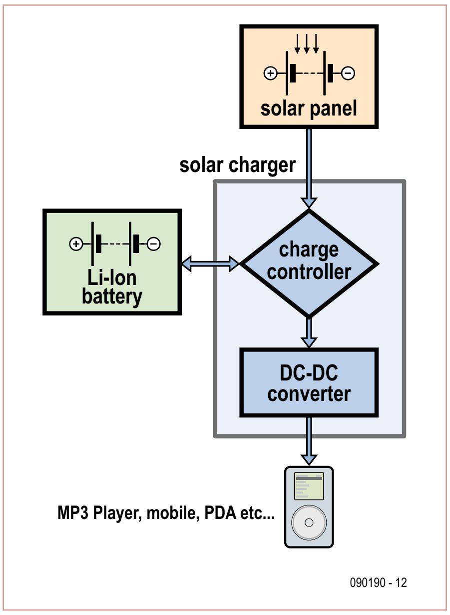


Figure 1. Functional diagram of the Solar Charger. Current from the solar panel is stored in a Li-Ion battery.

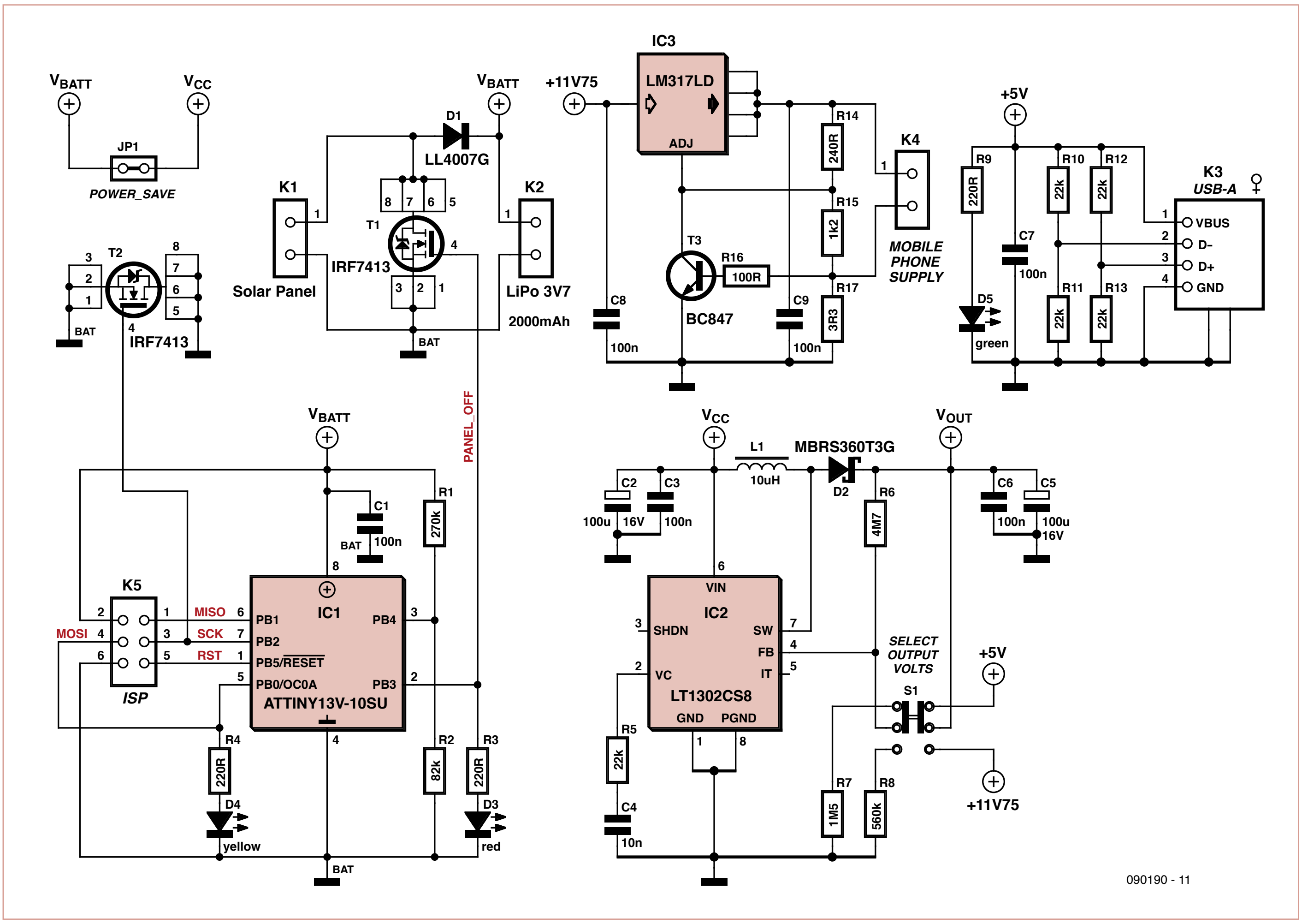


Figure 2. Circuitry of the Solar Charger. An ATtiny microcontroller monitors charging and discharging of the buffer battery.

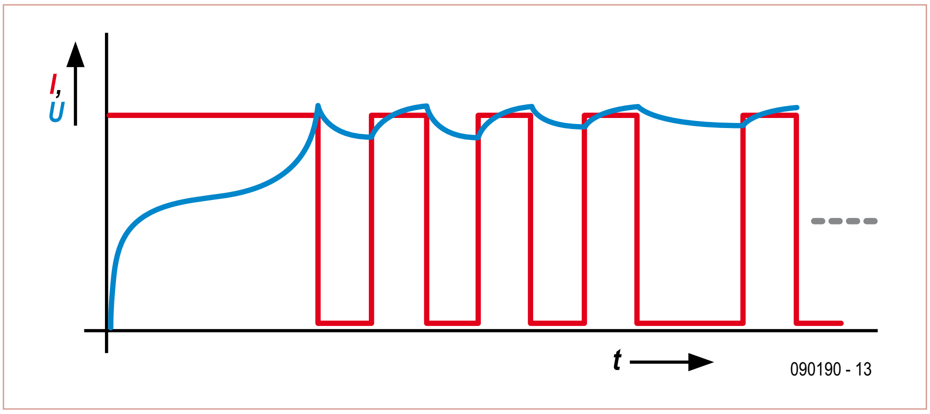


Figure 3. The charge regulator short-circuits the solar panel and cuts out the charging current (red), as soon as the voltage at the battery (blue) reaches the permitted maximum.

converter can be operated in two different modes, selected by a switch (S1). The first of these modes delivers 5 V to the USB connector, so that any devices that are charged from the USB supply can be charged in this way. The LT1302 is equipped with internal overload protection and switches off automatically if it overheats [2]. All the same, you should not allow the charge current for the USB device to exceed 500 mA. All devices that conform to the USB standard fulfil this requirement without exception [3].
The second mode of operation is intended for devices that require a constant current source for charging (for example some Nokia handsets). The author uses a Siemens BenQ S68 and this model requires a charging voltage of approx. 7 V to start the charging process. Subsequently it expects a constant charging current, until the mobile's battery reaches a voltage of approx. 4 V. At this point the handset disconnects the charge automatically.
This charging mode is achieved by a further stage that follows the boost converter. The output voltage of the boost converter is raised across R6 and R8 to 11.75 V. This voltage is then passed to an LM317 linear regulator, which provides an output voltage of close to 7 V and also limits the charging current [4]. Using a value of 3.3 Ω for R17 we get a charging current of approx. 150 mA, which should not be exceeded, in order not to overload the boost converter LT1302 [2].

Programming and regulation

Overall regulation of the circuit is handled by a microcontroller. The scheme uses two regulators with an interrupt-driven program; one looks after the charging terminal voltage and the other controls load dumping. The complete program flow is controlled by an interrupt occurring once every second. At the start of each interrupt LED D4 (yellow) is illuminated. Following this the existing battery voltage is compared against the stated limits for over and undervoltage states. Afterwards a new A-to-D conversion is initiated and LED D4 extinguished.

Charging LiPo batteries

The regulator for overvoltage short circuits the solar panel via T1 when the predefined maximum voltage of 4.15 V is reached and thus prevents overcharging the lithium-ion cell. To protect the battery from destruction the voltage of the cells must never exceed 4.2 V. For this reason the terminal voltage for charging is set at 4.15 V.
Figure 3 clarifies the circuit of the charging system. The red curve shows how the charging current of the solar panel works, simplified by assuming that the value of the current never varies. The blue curve represents the battery voltage. As seen, the charging current flows until the maximum permissible battery voltage is reached. At this point the solar panel is shunted and the battery voltage drops again.
At the next analogue-to-digital conversion the controller checks that the voltage of the battery is below the maximum permitted and allows charging current to flow once more. The battery voltage rises again now, occasionally even above the permissible limit, since the controller can measure the battery voltage only within the time windows defined. As the charge state of the battery drops, its voltage decreases constantly during the off-period of charging, so that the intervals between times when current flow is permitted again are constantly increasing.
In practice the battery is completely charged when LED D3 (red) remains on all the time.

Deep discharge protection by load release

The second regulator, for load dumping, takes the form of a two-position controller with hysteresis. If the lowest permissible voltage is crossed during discharge of a lithium-ion cell, the DC-to-DC converter is disconnected from the battery by T2. The battery voltage then recovers slightly until the next interrupt occurs. If the load is now reconnected immediately, then the charging process sees the exact reverse scenario: the battery voltage would drop constantly, taking with it the charge state of the battery.
Lithium-ion batteries must not be discharged too deeply, as this causes permanent damage. For this reason the terminal voltage during discharging is set here as 3.0 V. When this is reached, the hysteresis regulator for load dumping waits until the battery voltage is again at a higher level (e.g. 3.5 V) before load dumping is deactivated again.

Construction, commissioning and calibration

The PCB of the solar charger (**Figure 4)** uses predominantly surface-mount components (SMD devices). All of these are installed on the upper side of the board apart from up to four resistors. The software for the microcontroller, including source code, can be downloaded from the Elektor website [6]. If

COMPONENTS LIST

Resistors

SMD0805, 0.125W, 1%, if not shown otherwise)
R1 = 270kΩ
R2 = 82kΩ
R3,R4,R9 = 220Ω
R5,R10–R13 = 22kΩ
R6 = 4.7MΩ
R7 = 1.5MΩ
R8 = 560kΩ
R14 = 240Ω
R15 = 1.2kΩ
R16 = 100Ω
R17 = 3.3Ω

Capacitors

C1,C3,C6–C9 = 100nF (SMD0805, 10%)
C2,C5 = 100µF 16V, 10%, tantalum, SMD
C4 = 10nF (SMD0805, 10%)

Inductor

L1 = 10µH, 2.47A, 0.066Ω, 20%, ferrite core (e.g. Coiltronics DR74-100-R)

Semiconductors

D1 = 1N4007 (MELF), e.g. LL4007G or BAT85 (see text)
D2 = Schottky diode 3A, 60V (e.g. MBRS360T3G, On Semiconductor)
D3 = LED red, 25mA, SMD1206
D4 = LED yellow, 25mA, SMD1206
D5 = LED green, 20mA, SMD1206
T1, T2 = IRF7413 (International Rectifier)
T3 = BC847 SMD (e.g. BC847CLT1G, On Semiconductor)
IC1 = ATtiny13V-10SU (Atmel), programmed, Elektor Shop # 090190-41 *
IC2 = LT1302 (Linear Technology)
IC3 = LM317LD (e.g. from ST Microelectronics)

Miscellaneous

K3 = USB-A connector, SMD (e.g. Lumberg 2410 06)
S1 = Micro slide switch, 2-pole changeover (e.g. Multicomp MCLSS22)
K5 = Connector strip 6-pin, twin-row, 2.54mm spacing (e.g. Tyco/AMP 1241050-3)
JP1 = 2-pin pinheader with jumper, 0.1 in. lead pitch
LiPo battery 2000 mAh, 15C, 3.7V, e.g. Kokam 200015-0101G (834374H)
ASI-OEM solar panel 4.8V/80mA or 5V/81mA or similar
Schottky diodes BAT85 for parallel connection of the panels (see text)
PCB, order code 090190-1 *

* Available from the Elektor Shop and at www.elektor.com/090190

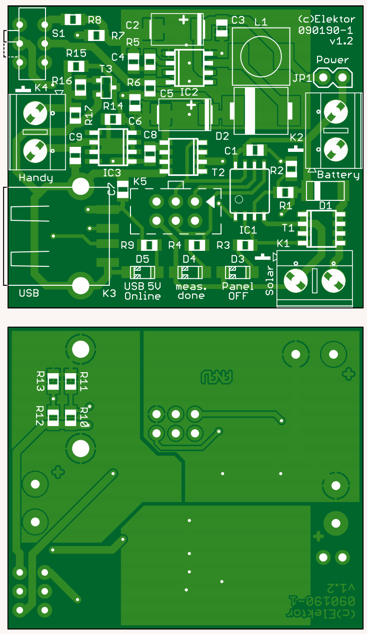


Figure4. The PCB of the Solar Charger uses mainly surface-mount components that are easy enough to solder. Four of the resistors are located on the underside of the board.

you don't feel inclined to program the ATtiny yourself, a ready programmed controller can be bought from the Elektor Shop.

As with any other project, a functional test is the next step after construction. This consists primarily of testing for effective protection against overvoltage, undervoltage and deep discharge. For this you will need a programmable power supply, which is connected in place of the Li-Ion battery.

The voltage is first set to 3.5 V and the function of the DC-to-DC converter checked (output current and voltage). After this the voltage is raised slowly until the red LED D3 begins to light. T1 should now short circuit the solar panel.

Now the voltage is slowly lowered until the green LED D5 goes out. For this exercise switch S1 in the DC-to-DC converter should be in 5 V (USB mode) position. With this load dumping test carried out successfully, the voltage is raised again until the green LED lights up.

The data sheet of the Atmel controllers gives the internal reference voltage of the controller as from 1.0 V to 1.2 V, meaning that the controller needs to be calibrated for the exact voltage limits.

The software provides three variables for this (SolarCharger.h):

- MEAS_BATT_MAX: gives the maximum battery voltage for overload protection.
- MEAS_BATT_MIN: gives the lower voltage limit for load dumping.
- MEAS_BATT_MIN_MAX: gives the upper limit for reconnecting the load.

Benchmark values for these limiting values are given in **Table 1** and are calculated as follows:

The A-to-D converter of the ATtiny has a resolution of 10 bits, i.e. 1024 separate values. The internal voltage source is indicated as a nominal 1.1 V. Using the given values of the voltage divider R1 and R2 and a maximum battery voltage of 4.72 V, the A-to-D converter will deliver a value of 1024. From this we deduce that one bit of the converter corresponds to 4.6 mV. In this way we can calculate all values for the voltage limits. The values in **Table 1** do not correspond to the exact value, however, on account of variance in the reference voltage. For this reason during the functional test it is important to note at which voltage each limit is reached. The correct value for the respective voltage limit can be calculated as follows:

$$Limit_{new} = \frac{U_{target}}{U_{actual}} \times Limit_{actual}$$

Solar panel and battery sizes

The prototypes used Kokam brand LiPo cells with 2 Ah capacity. These batteries, widely used by aircraft modellers, have the advantage of being flat and thus space-saving. Their high performance and discharge current make them relatively expensive, however. As high currents are not involved in

Table 1. Battery voltage limits relative to microprocessor internal reference. The exact value must be determined by calibration (see text).

		U_{ref} [V]		
		1.0	**1.1**	**1.2**
U_{batt} [V]	**4.15**	990	900	825
	3.50	835	759	696
	3.00	716	651	596

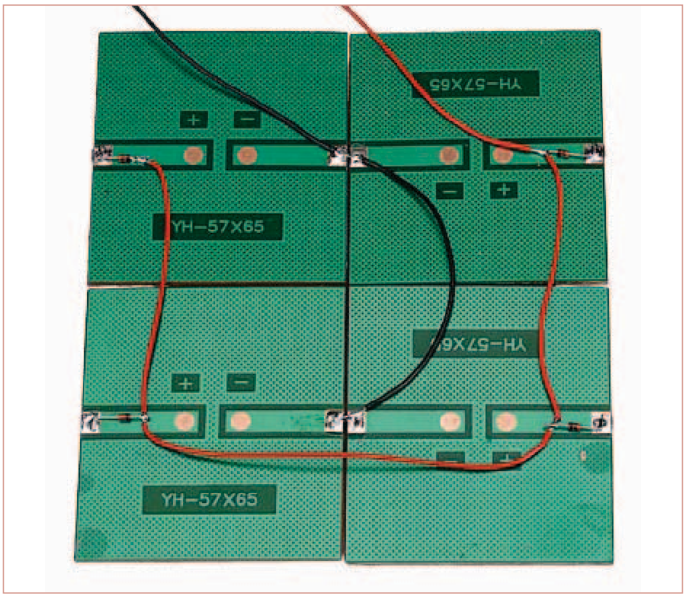


Figure 5. Cabling of the four solar modules, which are connected in parallel.

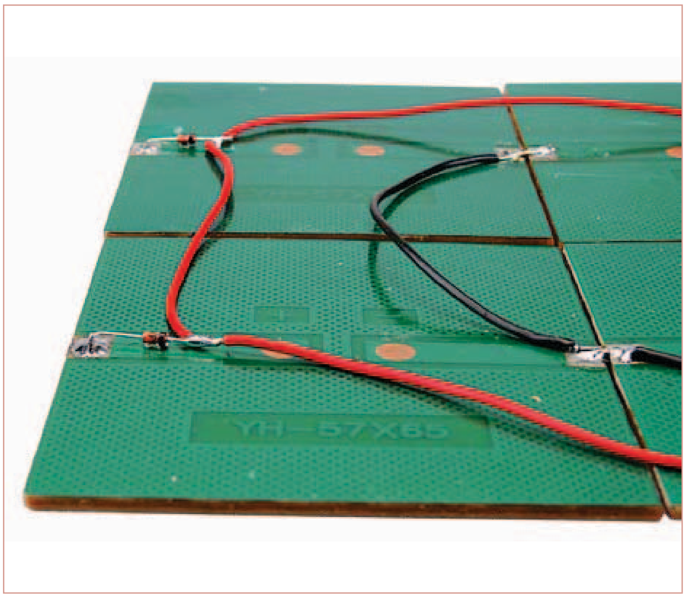


Figure 6. Schottky diodes ensure that no reverse current can flow through a module that is in shadow.

our application we can get away with more economical batteries, for instance the type 18650 round cells used in laptop batteries. The size of battery used in the charger is determined chiefly by the load created by (in other words the capacity of) the mobile phone battery to be recharged. The latter varies between 600 mAh (e.g. Siemens BenQ S68) and 1.6 Ah (e.g. Apple iPhone). If we start with the assumption that the boost converter of our solar charger has an efficiency of 80% and the battery has adequate capacity, then in order to fully recharge a 1.2 Ah mobile phone battery the charger battery needs to have a minimum capacity of 1.44 Ah. Taking this further, accepting that the battery in the solar charger will not always be fully charged, this means that a 2 Ah battery would be a safe choice. Whatever these values, it is clear that the storage battery in the solar charger must always have greater capacity than the one in the device being recharged.

The capacity of the battery will then determine the size of the solar panels. In the prototypes the solar panel was assembled from four solar modules wired in parallel, giving a nominal voltage of 5 V at a nominal current of 81 mA. This is a common size in trade catalogues.

In **Figure 5** you can see how we wired the four solar modules at Elektor Labs. Each of the positive connections was connected to the common +ve bus via a 200 mA Schottky diode (BAT85) arranged to permit current

Figure 7. Solar module arrangement in the author's prototype charger.

Figure 8. Internal view of the first prototype. The PCB is connected directly to the solar modules here.

Literature and Links

[1] Portable Solar Panels – Portable Energy On The Move, Elektor June 2009
[2] Data sheet LT1302, Linear Technology, http://cds.linear.com/docs/Datasheet/lt1302.pdf
[3] Universal Serial Bus specification, Revision 2.0, 27 April 2007, www.usb.org
[4] Data sheet LM317, Linear Technology, http://cds.linear.com/docs/Datasheet/lt0117.pdf
[5] Data sheet Attiny13(A), Atmel, http://www.atmel.com/dyn/products/product_card.asp?PN=ATtiny13A
[6] Project page with software download and ordering details at www.elektor.com/090190

The author

Martin Kiel (29) is a member of the scientific staff at the Institute for Rectifier Technology and Electrical Drive Trains (RWTH University, Aachen) in Germany. His specialism is measurement and diagnostics technology for batteries. He became a licensed radio amateur in 1996 and spends his spare time in electronics.

flow from the module to the bus (see close-up photo **Figure 6**). These diodes block any backward current flow through individual modules when they are in shadow or are delivering a reduced voltage to the others for some other reason. This set-up provides a total charging current of 324 mA, meaning that the 2 Ah storage battery should be fully charged in six hours (in theory). These diodes were omitted in the author's original prototype (**Figure 7** and **Figure 8**), which differs in a few details from the Elektor version presented here. The reverse current through an obscured (or underperforming) module is not really critical but it will reduce the output current and hence the performance of the solar panels. The Schottky diodes block the reverse current but also introduce a permanent loss in performance on account of the voltage drop of about 0.4 V at 80 mA, equivalent to around 8 % at maximum output of the solar modules used here. Nevertheless Elektor Labs recommend using these diodes.

In principle larger solar modules, as described in [1], can also be used, for example those with a voltage of 12 V. The controller ensures that the voltage does not exceed critical limits and thus protects the battery. All the same, a panel of such large size would never be able to deliver full performance, as the voltages will always be well below the optimal operating point.

(090190-I)

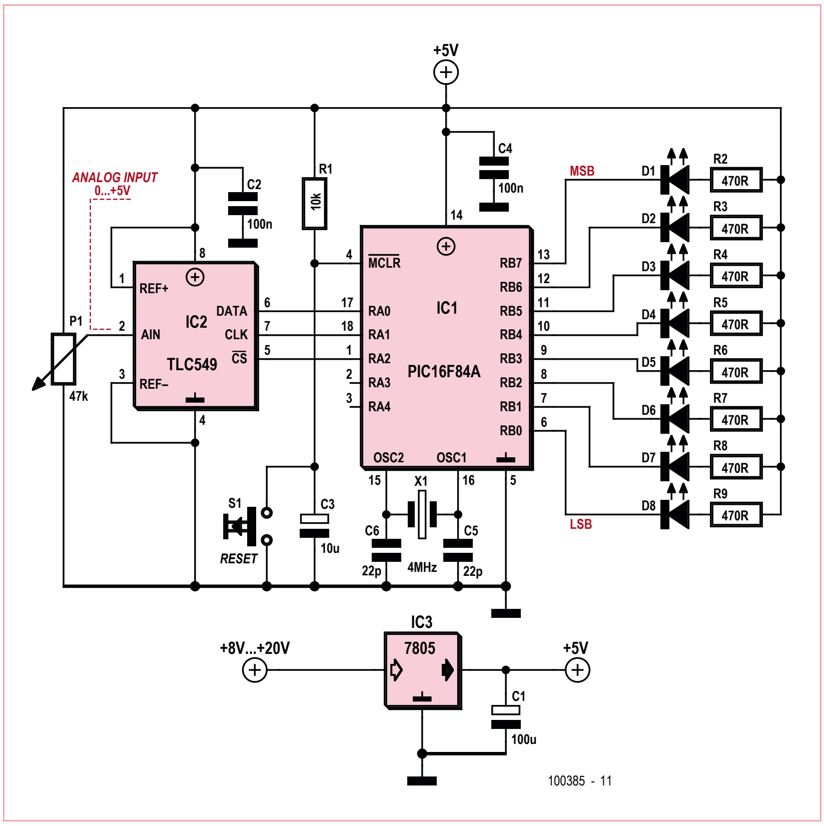
ADC for the PIC16F84A

By Eric Vanderseypen (Belgium)

The good old PIC16F84A does not have an analogue to digital converter (ADC) on board. A good solution to this problem is offered by the TLC549 serial ADC made by Texas Instruments. The TLC549 uses only 3 of the I/O pins of the controller, is very compact and is also readily available.

The TLC549 control lines $\overline{CS}$ and I/O-Clock are controlled from the PIC. The result of the conversion is available via the serial output of the ADC (Data Out), one bit at a time, and is stored into a byte of RAM in the PIC. You can find a detailed description of how this works in the Texas Instruments data sheet: http://focus.ti.com/lit/ds/symlink/tlc549.pdf. The program shows how the 8 bits in the RESULT byte are fetched in (lines 10 to 25).

For the sake of clarity it was decided to show each and every step in full. The clock signal for the ADC is generated by the subroutine IOCLOCK (lines 34 to 38). The chip select input of the ADC is controlled by program lines 09 and 26. The operating sequence diagram (datasheet page 3) clearly shows how the $\overline{CS}$ has to be controlled. The TLC549 will not operate correctly when the $\overline{CS}$ is permanently connected to ground.



The routine SHIFTIN takes care of assembling the RESULT byte. RA0 (Data Out) is first copied to the carry bit (31). The carry is subsequently left shifted into the RESULT byte (32). Since the order of the data bits from the conversion result is MSB first and LSB last (see datasheet), this ensures that the conversion result ends up the correct way around in the RESULT byte, after going through a complete cycle.

In the schematic you can see that PORTB is used to visualise the result of the conversion using LEDs. If you have another use for PORTB then you may omit program lines 27 and 28. An analogue signal for the input of the ADC is simulated with potentiometer P1. The value is not critical; use a higher value to avoid unnecessary loading of the power supply.

When flash programming the PIC client, the reset circuit (R1, C3 and RST) needs to be disconnected from pin 4 ($\overline{MCLR}$).

(100385-I)

Hexadoku ‘Digest’

Win a Cypress PSoC 5 Starter kit!

This month’s Hexadoku is unusual and pays homage to our long term readers and faithful subscribers. To be able to complete the grid and find the solution you’ll need the solutions to previous Hexadokus dug out from your archives or cleverly found using the Elektor website. Enter the right numbers in the puzzle, send the ones in the grey boxes to us and you automatically enter the prize draw for one of 10 Cypress FirstTouch PSoC 5 starter kits. Have fun!

Hexadoku ‘Digest’ contains 16 horizontal blocks of 5 cells each marked yellow. Each block takes the solution of a Hexadoku puzzle that appeared in one of 16 editions of Elektor magazine in the period **January 2009 to June 2010**. The July & August 2009 double edition is excluded. If you do not have these editions (printed copies or digital), the solutions may be found via the ‘Magazine’ tab at www.elektor.com. The Hexadoku puzzle employs the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that’s 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

Correct entries received enter a prize draw. All you need to do is send us **the numbers in the grey boxes**.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for one of 10 **PSoC 5 FirstTouch starter kits (CY8KIT-014)** kindly donated by Cypress and representing a value of £45.00 each (approx. rrp).

Participate!

Before April 1, 2011, send your solution (the numbers in the grey boxes) by email, fax or post to

Elektor Hexadoku – 1000, Great West Road – Brentford TW8 9HH United Kingdom.

Fax (+44) 208 2614447 Email: hexadoku@elektor.com

Prize winners

The solution of the January 2011 Hexadoku is: **B278F**.

The £80.00 voucher has been awarded to: Walter Rothleitner (Germany).

The £40.00 vouchers have been awarded to: Ludovic Robichon (France), Chris Smith (UK), Claude Guyon (France).

Congratulations everyone!

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

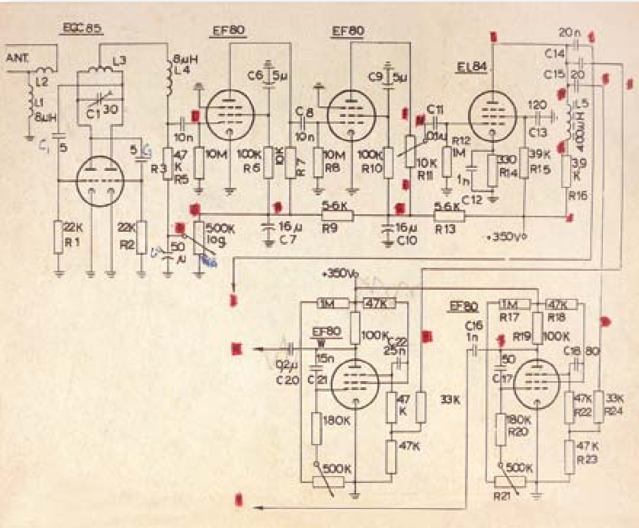
F	7	3	A	6	0	5	C	2	1	9	D	B	8	4	E
C	D	1	4	A	B	2	7	8	F	E	5	9	0	3	6
B	E	2	0	D	9	8	1	C	6	4	3	F	7	A	5
9	5	6	8	E	F	3	4	7	A	0	B	1	C	2	D
D	F	A	3	7	C	E	B	5	9	6	1	8	2	0	4
5	8	C	2	1	A	F	0	D	B	7	4	3	6	E	9
E	6	7	1	2	4	9	D	3	0	8	A	5	F	B	C
0	4	9	B	8	3	6	5	E	C	F	2	D	A	1	7
1	2	B	5	F	7	4	E	0	8	A	C	6	9	D	3
3	9	D	C	B	8	A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

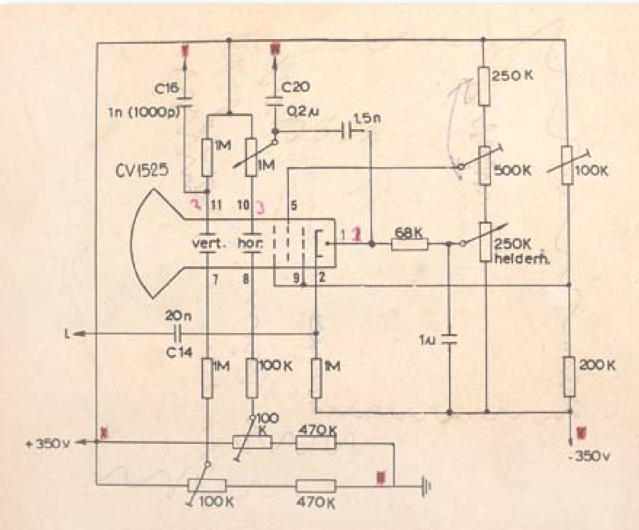
The Worst TV Set Ever (1962)

By Karel Walraven (The Netherlands) [1]

In the 1950s there were lots of DIY electronics designs for sale. They were printed on large sheets of paper (easily up to a metre in width), with precise descriptions of how to build the project — such as an audio amplifier or a radio receiver. In the Netherlands, such 'plans' or 'blueprints' were published by various companies, including Amroh, originally a transformer manufacturer that found this a good way to boost their turnover. *Electronica Wereld* magazine, the precursor of *Elektuur,* was launched in 1961 [2]. The founder and editor in chief, Bob van der Horst, liked to publish articles on as many different topics as possible, and one of them was a DIY television set [3] with the simplest possible design and consequently low specs like 7 cms screen diagonal, 1 MHz bandwidth and 312 picture lines. The Worst TV Set Ever (in Dutch: De Slechtste TV Ooit) was published in a number of instalments starting in *EW* issue # 7. Alternatively, by December 1962 you could obtain 'the plans' of the project by post for two and a half Dutch guilders [4].

I was 14 or 15 years old at that time. I knew almost nothing about electronics, but the idea of building a television set appealed to me. Only a couple of people on our street had a television set then – you could see who they were from the aerial masts on the roofs, rising to a height of five metres or more. Not hindered by any knowledge of the subject, I began collecting the necessary components, which was not so easy because I lacked not only expertise, but also money. Fortunately, I had an abundance of radio valves, which I 'found' behind our house. Valve manufacturers such as Philips and Telefunken were paranoid that people would stuff old, used valves into the (unsealed) cartons they used to package their radio valves and sell them as new. For this reason, the valves and packages were not allowed to be disposed of as normal rubbish; everything had to be destroyed by the dealers. The radio & TV dealer who lived behind my parents' house did this in a tidy manner: he tossed the rubbish in his back garden and burned it every Friday afternoon. When you're 14 years old, you don't let yourself be stopped by a couple of fences, so I always had an ample supply of valves. To my amazement, most of them were not actually bad; apparently a lot of valves were replaced for 'preventive' reasons (although I suppose their cathode emission was probably weak).





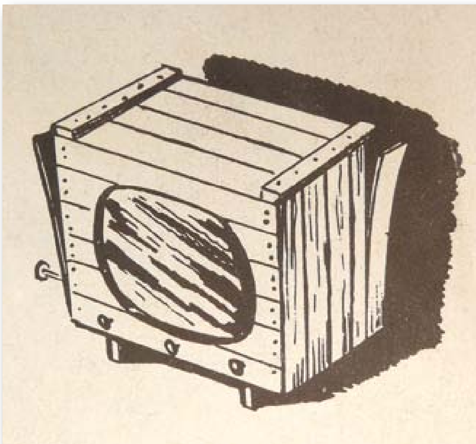
The construction of the television set was described very clearly with three-dimensional drawings (nowadays we would use a PCB layout), and putting it together was actually not especially difficult. It was composed of several small modules, each with one valve and a few components, so everything was easy to follow. You could buy ready-made assembly boards, but I couldn't afford them. For 1 Dutch guilder you could buy a piece of Pertinax panel the size of an A4 sheet, and that's what I used to make my modules. All I had to do was drill the holes and fit them with hollow rivets and solder lugs, which cost almost nothing.

Winding the coils went very well, but I had to buy the choke (L5), and things were no better then than now: the sales clerk in the local electronics shop didn't know anything about coils and was obviously very nervous.

The only expensive component was the type CV1525 CRT, which you could pick up at an NATO surplus store for 15 guilders or so. I would have rather had a DG7-32 because of its lower operating voltage and higher sensitivity, but it was way beyond my budget. I therefore worked with a potentially lethal voltage of 700 V, but thanks to my good instincts I managed to survive. I can still remember that I used to have a valve receiver on my bed right next to the pillow — with no enclosure, of course.

I quickly reached the point where I had a 'picture' on the screen. It wasn't a real television picture, but rather a bunch of narrow stripes on the screen that could be regarded as a green rectangle if you were feeling generous. The instructions were also clear on this point: you couldn't expect high quality. That didn't bother me at all; the idea that I had something as unattainable as a television set within my grasp was fantastic, and I was thrilled.

The instructions mentioned that 'the high-frequency portion is expected to cause problems for many builders'. Unfortunately, this proved to be all too true. I was fairly certain that most of the circuit worked properly, because a

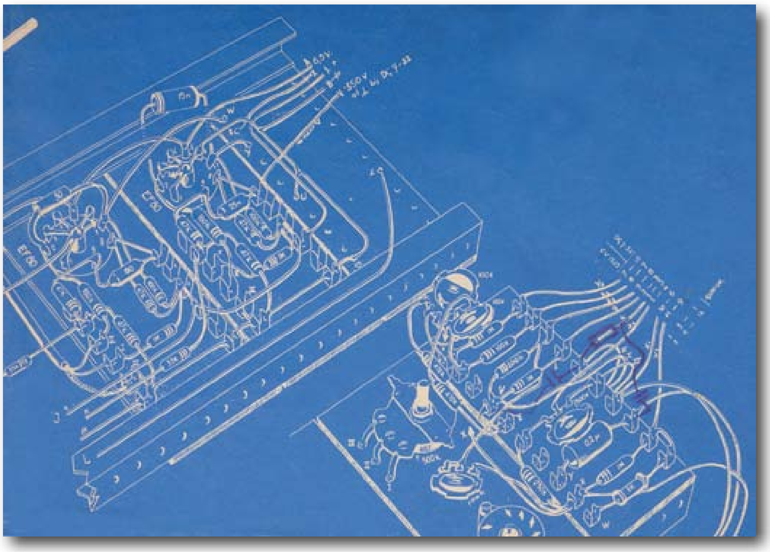


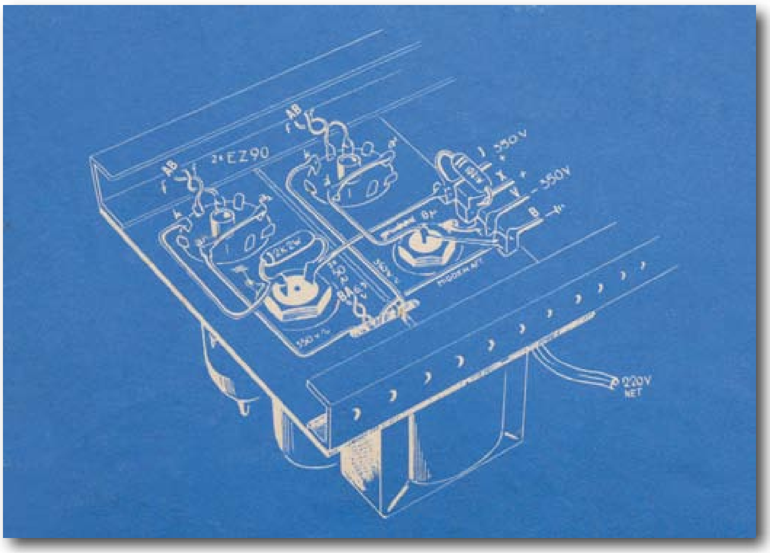
test with a square-wave generator produced a nice black & white (actually black & green) screen. However, I couldn't get the receiver portion to work. The first bit of trouble came when I adjusted the 500-kΩ potentiometer that was supposed to set the receiver to the point where it was just about to oscillate. This regularly resulted in the potentiometer burning out. Another thing that wasn't very encouraging was that the instructions included a lot of solutions for problems you might encounter, and in one corner there were several corrected component values. Accordingly, I tried all of these solutions and changed the component values, but to no avail. Probably the worst thing was that I had absolutely no idea whether the receiver was actually tuned to the 'Lopik' transmitter on VHF TV channel 4 [5]. The instructions said something in passing about pressing the coils together or spreading them apart to obtain the right frequency. I also lived in an unfavourable location; Lopik was nearly 70 miles away and the signal was bound to be so weak that it's a good question whether that simple receiver would have been able to receive it, even if it had worked properly. To make things even worse, I did not have access to an aerial on a tall mast. The best I could manage was a length of flat cable on a board outside my bedroom window. It's therefore hardly surprising that I never saw any picture, despite all my efforts.

However, I didn't suffer from this; it was all very exciting, and I soon switched my attention to the next project. I also learned a lot from it, since the fact that it didn't work forced me to think and to try to understand the circuit. I can still remember very well how fantastic it was to see a moving point of light on the screen – proof that invisible particles were in fact striking the phosphor layer. My mother had less appreciation for the greatness of this event; she kept asking, "Do you really need all that just to make a little bit of light?"

Years later, I used the CRT and the power supply to build a DIY oscilloscope, the 'Glow Worm' (Glimworm), published in a competing magazine.

(100748)





For historical interest the original plans of the Worst TV Ever have been scanned and may be downloaded free of charge from www.elektor.com/100748. The published material is in Dutch.

Editorial Notes

[1] Former International Co-ordinating Editor and Head of Lab. This article is published in honour of Karel's massive contribution to Elektor over the period 1975-2006.

[2] Name change to *Elektuur* as of November 1964 after legal advice from the publishers of the English-language *Electronics World*. *Elektuur* is the mother of all international editions of Elektor, celebrating its 50th anniversary this year.

[3] Valve complement: ECC85, EF80 (4x), EL84, ECF80, CV1525.

[4] Slightly less than the cost of two editions of the magazine.

[5] Actual location: Ijsselstein. The only high power TV broadcast transmitter in The Netherlands at the time. Vision carrier 67.250 MHz, antenna height approx. 250 m ASL.

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

To book your showcase space contact Elektor International Media

Tel. 0031 (0) 46 4389444 | Fax 0031 (0) 46 4370161

ASTROBE V3.0

www.astrobe.com

Windows Development System for LPC2000 microcontrollers.

- Develop high-and low-level software without C or assembler
- Generate fast 32-bit native ARM code
- No special programming hardware required
- Personal, Standard and Professional Editions

ATOMIC PROGRAMMING LTD

www.atomicprogramming.com

- AP-114 ISP/JTAG Programming System
- JTAG Programming and Testing
- Boundary Scan Testing
- Universal In-System Programming
- EEPROM and SPI Flash Out-of-Circuit Programming
- Generic GDB Proxy Server
- Jennic JN5148 ZigBee Development Applications
- Training Platform available

AVIT RESEARCH

www.avitresearch.co.uk

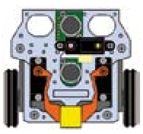
USB has never been so simple...
with our USB to Microcontroller Interface cable. Appears just like a serial port to both PC and Microcontroller, for really easy USB connection to your projects, or replacement of existing RS232 interfaces.

See our webpage for more details. From £10.00.

BLACK ROBOTICS

www.blackrobotics.com

Robot platforms and brains for research, hobby and education.



- Make your robot talk!
- TalkBotBrain is open-source
- Free robot speech software
- Robot humanisation technology
- Mandibot Gripper Robot

TO BOOK YOUR SHOWCASE SPACE CONTACT ELEKTOR INTERNATIONAL MEDIA

Tel. 0031 (0) 46 4389444
Fax 0031 (0) 46 4370161

CEDA

www.ceda.in

ceda@vsnl.com

CEDA e-learning
PADS | OrCAD | ALLEGRO
PCB layout@ $5 Hourly

- PCB Layout & library service @$5 Hourly
- PCB Reverse Engg, MCU Code/ckt. Extraction service
- Learn Signal Integrity with Hyperlynx, PCB Design with PADS & Allegro. Support by email & web meeting

DESIGNER SYSTEMS

http://www.designersystems.co.uk

Professional product development services.

- Marine (Security, Tracking, Monitoring & control)
- Automotive (AV, Tracking, Gadget, Monitoring & control)
- Industrial (Safety systems, Monitoring over Ethernet)
- Telecoms (PSTN handsets, GSM/GPRS)
- AudioVisual ((HD)DVD accessories & controllers)

Tel: +44 (0) 845 5192306

EASYSYNC LTD.

www.easysync-ltd.com/

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

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

ELNEC

www.elnec.com

Europe's leading device programmers manufacturer:

- reliable HW: 3 years warranty for most programmers
- support over 58.000 devices
- free SW updates
- SW release: few times a week
- excellent technical support: Algorithms On Request, On Demand SW
- all products at stock / fast delivery

EMBEDDED ADVENTURES

www.embeddedadventures.com

From news and tutorials to modules, components and kits, we have everything for your next microcontroller based project.
Your embedded adventure starts here.

embedded adventures

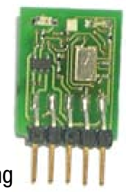
FIRST TECHNOLOGY TRANSFER LTD.

http://www.ftt.co.uk

First Technology Transfer Ltd.

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

FLEXIPANEL LTD



www.flexipanel.com

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

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

FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD.

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

FTDI specialise in USB silicon, hardware and software solutions.

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

USB MADE EASY

HAMEG® Instruments
A Rohde & Schwarz Company

- ☑ Oscilloscopes
- ☑ Power Supplies
- ☑ Spectrum Analyzers
- ☑ RF Instruments
- ☑ Programmable Measuring Instruments

Great Value in Test & Measurement

www.hameg.com

products and services directory

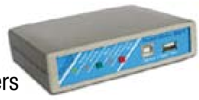
HEXWAX LTD

www.hexwax.com

World leaders in Driver-Free USB ICs:

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

MQP ELECTRONICS



www.mqp.com

- Low cost USB Bus Analysers
- High, Full or Low speed captures
- Graphical analysis and filtering
- Automatic speed detection
- Bus powered from high speed PC
- Capture buttons and feature connector
- Optional analysis classes

ROBOT ELECTRONICS

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

Advanced Sensors and Electronics for Robotics

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

ROBOTIQ

http://www.robotiq.co.uk

Build your own Robot!

Fun for the whole family!

Now, available in time for X-mas

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

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

STEORN SKDB LITE

Join the SKDB Lite, the place to understand, discuss and experiment with magnetics.

- Learn more about magnetics and electromagnetics
- Participate in developer forums and discussion surrounding magnetics and related topics.

For FREE access to SKDB Lite: https://kdb.steorn.com/ref25

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.

www.
elektor.
com

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 0031(0)46 4370161

COMPANY NAME ....

WEB ADDRESS ....

30-WORD DESCRIPTION ....

....

....

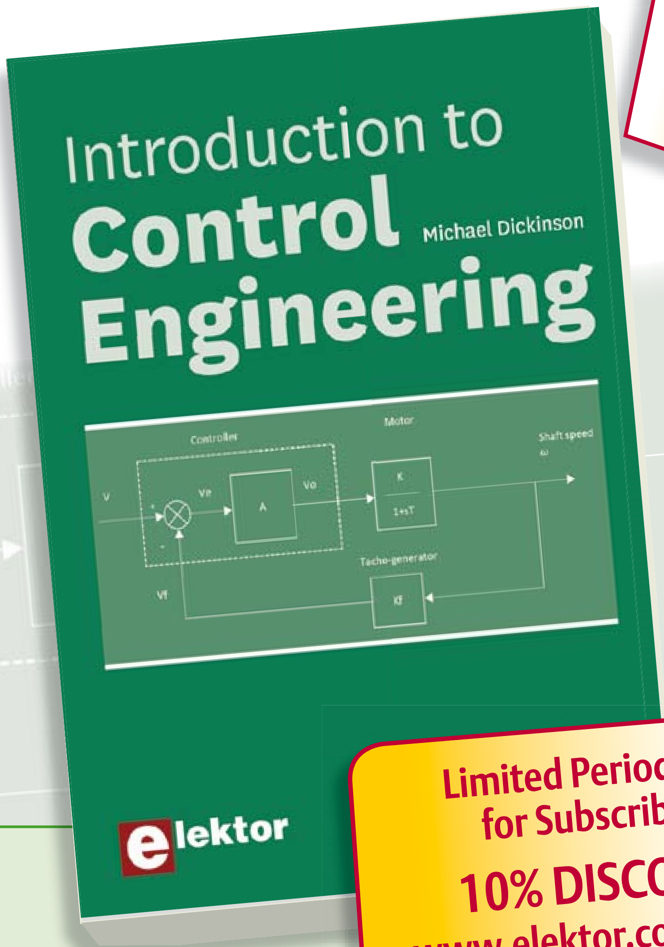
....

....

Going Strong

A world of electronics from a single shop!

NEW!



Limited Period Offer for Subscribers!
10% DISCOUNT
www.elektor.com/march

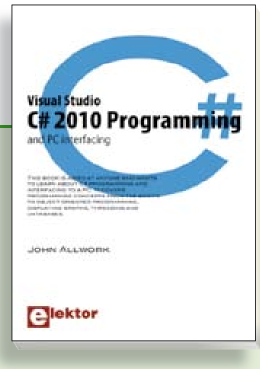
Books

A sourcebook of solutions for control system applications

Introduction to Control Engineering

This book is intended as a source of reference for hardware and software associated with instrumentation and control engineering. Examples are presented from a range of industries and applications. Throughout the book, circuit diagrams and software listings are described, typical of many measurement and control applications. The hardware and software designs may be used as a basis for application by the reader. The book contains examples of PIC, PLC, PAC and PC programming. All code samples are available to download free of charge from the support website. After an introductory section on control theory and modelling, the text focus is upon software for control system simulation and implementation, with appropriate reference to interfacing, electronic hardware and computing platforms.

164 pages • ISBN 978-0-905705-99-6 • £27.50 • US $44.40

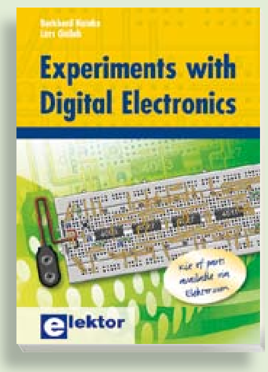


Visual Studio

C# 2010 Programming and PC interfacing

This book is aimed at anyone who wants to learn about C# programming and interfacing to a PC. It covers programming concepts from the basics to object oriented programming, displaying graphs, threading and databases. The book is complete with many full program examples, self assessment exercises and links to supporting videos. All code examples used are available – free of charge – from a special support website. Professional quality software tools are downloadable –also free of charge– from Microsoft. The Microsoft Visual Studio 2010 environment is extensively covered with user controls and their properties, methods and events.

306 pages • ISBN 978-0-905705-95-8
£29.50 • US $47.60

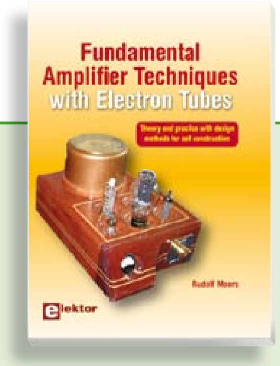


An introduction to digital control electronics

Experiments with Digital Electronics

This book presents fundamental circuits using gates, flip-flops and counters from the CMOS 4000 Series. Learning these fundamentals is best done using practical experiments. Each of the 50 experiments presented in this book has a circuit diagram as well as a detailed illustration of the circuit's construction on solderless breadboard.

176 pages • ISBN 978-0-905705-97-2
£26.50 • US $42.80



The ultimate tube amplifier reference book

Fundamental Amplifier Techniques

The aim of this book is to give the reader useful knowledge about electron tube technology in the application of audio amplifiers, including their power supplies, for the design and DIY construction of these electron tube amplifiers. This is more than just building an electron tube amplifier from a schematic made from the design from someone else. No modern simulations, but because you first understand the circuit calculations, then you can work with your hands to build the circuit.

834 pages • ISBN 978-0-905705-93-4
£65.00 • US $104.90

Associated starter kit available

ARM Microcontrollers

This is the perfect book for people who want to learn C and who want to use an mbed ARM microcontroller in an easy and fun way. The mbed NXP LPC1768 uses cloud technology. This means you do not need to install software on your PC in order to program the mbed! The only thing you need is a browser and a USB port on your PC. No previous experience or knowledge required. You can get access to your project from any PC anywhere in the world and continue working on it. When you are done a few mouse clicks transfer the program to your mbed hardware.

250 pages • ISBN 978-0-905705-94-1
£29.50 • US $47.60

Enhanced second edition

Design your own Embedded Linux Control Centre on a PC

The main system described in this book reuses an old PC, a wireless mains outlet with three switches and one controller, and a USB webcam. All this is linked together by Linux. This book will serve up the basics of setting up a Linux environment – including a software development environment – so it can be used as a control centre. The book will also guide you through the necessary setup and configuration of a webserver, which will be the interface to your very own home control centre. New edition enhancements include details of extending the capabilities of your control center with ports for a mobile phone (for SMS messaging) and the Elektor "thermo snake" for low-cost networked real-time thermal monitoring of your house and outbuildings. Now you can additionally also send all kinds of useful temperature and sensor warnings to a mobile phone. All software needed will be available at the Elektor website.

416 pages • ISBN 978-1-907920-02-8
£34.50 • US $55.70

More information on the Elektor Website:

www.elektor.com

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

elektor

CD/DVD-ROMS

More than 75 power supply designs

CD The Power Supply Collection 1

This CD-ROM contains more than 75 different power supply circuits from the volumes 2001-2005 of Elektor. Highlights include the Cuk Converter, Automatic Battery Switchover, Battery Voltage LED, Digital Benchtop Power Supply, Lithium-Ion Charger, Electronic Fuse, High Voltage Regulator, Power Supply for USB Devices, Step-up Converter for White LEDs, Vehicle Adapter for Notebook PCs and much more. Using the included Adobe Reader you are able to browse the articles on your computer, as well as print texts, circuit diagrams and PCB layouts.

ISBN 978-90-5381-265-5
£17.90 • US $28.90

75 Audio designs for home construction

DVD The Audio Collection 3

This DVD contains more than 75 different audio circuits from the volumes 2002-2008 of Elektor. The articles on the DVD-ROM cover Amplifiers, Digital Audio, Loudspeakers, PC Audio, Test & Measurement and Valves. Highlights include the ClariTy 2x300 W Class-T amplifier, High-End Power Amp, Digital VU Meter, Valve Sound Converter, paX Power Amplifier, MP3 preamp and much more. Using the included Adobe Reader you are able to browse the articles on your computer, as well as print texts, circuit diagrams and PCB layouts.

ISBN 978-90-5381-263-1
£17.90 • US $28.90

CD/DVD-ROMs

RFID, NFC, Zigbee, GPS and more

DVD Wireless Toolbox

On this DVD-ROM you'll find a number of technical documents and tools that will enable you to add wireless data exchange to your electronics systems. The choice of equipment depends on the transmission distance: a few centimetres using Near Field Communication (NFC) or Radio Frequency Identification (RFID), tens of metres with the Bluetooth, Wi-Fi or ZigBee systems, or indeed thousands of kilometres using a module for receiving GPS data. The DVD contains technical documentation (spec. sheets, application notes, user guides, etc.) on various devices according to the frequency and/or protocol used. All of the documents are PDF files (in English).

ISBN 978-90-5381-268-6
£28.50 • US $46.00

110 issues, more than 2,100 articles

DVD Elektor 1990 through 1999

This DVD-ROM contains the full range of 1990-1999 volumes (all 110 issues) of Elektor Electronics magazine (PDF). The more than 2,100 separate articles have been classified chronologically by their dates of publication (month/year), but are also listed alphabetically by topic. A comprehensive index enables you to search the entire DVD.

ISBN 978-0-905705-76-7
£69.00 • US $111.30

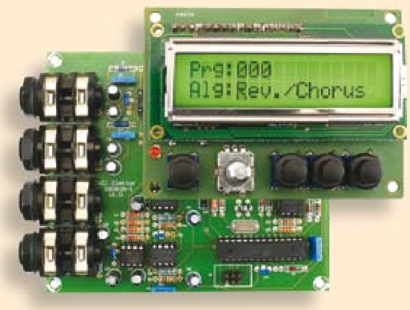
NetWorker

(December 2010)

An Internet connection would be a valuable addition to many projects, but often designers are put off by the complexities involved. The 'NetWorker', which consists of a small printed circuit board, a free software library and a ready-to-use microcontroller-based web server, solves these problems and allows beginners to add Internet connectivity to their projects. More experienced users will benefit from features such as SPI communications, power over Ethernet (PoE) and more.

Module, ready assembled and tested

Art.# 100552-91 • £53.00 • US $85.50



Digital Multi-Effects Unit

(September 2010)

It's a simple fact: every recording sounds better with the right sound effects. Here we prove that it's possible to generate a variety of effects digitally, including hall, chorus and flanger effects, without having to work yourself to the bone with DSP programming. The circuit is built around a highly integrated effects chip and features an intelligent user interface with an LCD. The result is a treat for the eye and the ear.

Kit of parts including PCBs, programmed controllers and EEPROM

Art.# 090835-71 • £165.00 • US $266.20

The Elektor DSP radio

(July/August 2010)

Many radio amateurs in practice use two receivers, one portable and the other a fixed receiver with a PC control facility. The Elektor DSP radio can operate in either capacity, with a USB interface giving the option of PC control. An additional feature of the USB interface is that it can be used as the source of power for the receiver, the audio output being connected to the PC's powered speakers. To allow portable 6 V battery operation the circuit also provides for an audio amplifier with one or two loudspeakers.

PCB, assembled and tested

Art.# 100126-91 • £149.00 • US $240.40

Reign with the Sceptre

(March 2010)

This open-source & open-hardware project aims to be more than just a little board with a big microcontroller and a few useful peripherals — it seeks to be a fast prototyping system. To justify this title, in addition to a very useful little board, we also need user-friendly development tools and libraries that allow fast implementation of the board's peripherals. Ambitious? Maybe, but nothing should deter you from becoming Master of Embedded Systems Universe with the help of the Elektor Sceptre.

PCB, populated and tested, test software loaded (excluding Bluetooth module)

Art.# 090559-91 • £89.00 • US $143.60

Kits & Modules

Product Shortlist

			£	US $
March 2011 (No. 411)				
+ + + Pr oduct Shor tlist M arch: See www .elektor.com + + +				
February 2011 (No. 410)				
Gentle Awakenings				
080850-1	Printed circuit board		28.90	47.10
080850-41	ATmega168-20PU, programmed		8.75	14.20
Ultimatic CW Keyer				
100087-41	PIC16F688-I/P, programmed		8.75	14.20
Educational Expansion Board				
100742-1	Printed circuit board		26.00	41.90
Contactless Thermometer				
100707-1	Printed circuit board		20.50	33.10
100707-41	PIC16F876A DIL28, programmed		13.35	21.40
TimeClick				
100371-1	Printed circuit board		36.00	58.00
100371-41	ATtiny861-20SU, programmed		10.60	17.10
MIAC Controlled Underfloor Heating System				
MI0235	MIAC-PLC		154.00	248.40
MI1472	MIAC and Flowcode 4		275.10	447.90
MI3487	3 x MIAC and Flowcode 4		596.30	971.00
Linux'ed Telephone-to-VoIP Adapter				
100761-1	Printed circuit board		8.15	13.20
100761-41	PIC18F2550-I/SO, programmed		13.25	21.40
January 2011 (No. 409)				
Nixie Tube Thermometer				
090784-1	Printed circuit board		12.40	20.00
090784-41	Programmed controller AT89C2051/24PU		8.75	14.10
Flight Data Recorder				
071035-91	ATM18 controller module		7.30	15.40
090773-91	PCB, populated and tested with programmed bootloader		56.00	90.00
100653-1	Printed circuit board		12.95	20.90
Low-cost Headphone Amp				
100500-71	Elektor Project Case		16.80	25.80
100701-1	Printed circuit board		8.75	14.10
Wireless ECG				
080805-1	Printed circuit board		8.75	14.10
Support Board for Arduino Nano				
100396-1	Printed circuit board		18.00	29.00
December 2010 (No. 408)				
NetWorker				
100552-91	Module, ready assembled and tested		53.00	85.50
Heating System Monitor				
090328-41	ATmega328-20AU (TQFP32-08), programmed		11.00	17.80
Stroboscopic PC Fan				
100127-1	Printed circuit board		4.50	7.30
100127-41	ATtiny 2313, programmed		14.20	8.75
ARM Freephone Control				
080632-91	ECRM40 module, ready assembled and tested		32.00	51.70
Modular LED Message Board				
100664-41	MC9S08SH32CWL, programmed		8.75	14.20
Speed Controller for Small DC Motors				
100571-41	ATtiny44-20PU, programmed		8.75	14.20
November 2010 (Nr. 407)				
Micro Fuel Cell Measures Oxygen Concentration				
090773-91	PCB, populated and tested with programmed bootloader		56.00	90.40
The 5532 OpAmplifier (2)				
100124-1	Amplifier board (one channel)		23.00	37.10
100124-2	Power supply board		17.95	29.00
Camera Interval Timer				
081184-41	PIC16F886-I/SP, SPDIP28, programmed		8.00	12.90

Bestsellers

				£	US $
Books	1	ARM Microcontrollers	ISBN 978-0-905705-94-1	£29.50	US $47.60
Books	2	C# 2010 Programming and PC interfacing	ISBN 978-0-905705-95-8	£29.50	US $47.60
Books	3	Fundamental Amplifier Techniques with Electron Tubes	ISBN 978-0-905705-93-4	£65.00	US $104.90
Books	4	Experiments with Digital Electronics	ISBN 978-0-905705-97-2	£26.50	US $42.80
Books	5	Power Electronics in Motor Drives	ISBN 978-0-905705-89-7	£29.50	US $47.60
CD/DVD-ROMs	1	DVD Wireless Toolbox	ISBN 978-90-5381-268-6	£28.50	US $46.00
CD/DVD-ROMs	2	CD The Power Supply Collection 1	ISBN 978-90-5381-265-5	£17.90	US $28.90
CD/DVD-ROMs	3	DVD Elektor 1990 through 1999	ISBN 978-0-905705-76-7	£69.00	US $111.30
CD/DVD-ROMs	4	DVD The Audio Collection 3	ISBN 978-90-5381-263-1	£17.90	US $28.90
CD/DVD-ROMs	5	Masterclass DVD High-End Valve Amplifiers	ISBN 978-0-905705-86-6	£24.90	US $40.20
Kits & Modules	1	NetWorker	Art. # 100552-91	£53.00	US $85.50
Kits & Modules	2	Reign with the Sceptre	Art. # 090559-91	£89.00	US $143.60
Kits & Modules	3	Elektor DSP radio	Art. # 100126-91	£149.00	US $240.40
Kits & Modules	4	MIAC-PLC	Art. # MIO235	£154.00	US $248.40
Kits & Modules	5	Digital Multi-Effects Unit	Art. #090835-71	£165.00	US $266.20

Order quickly and securely through

www.elektor.com/shop

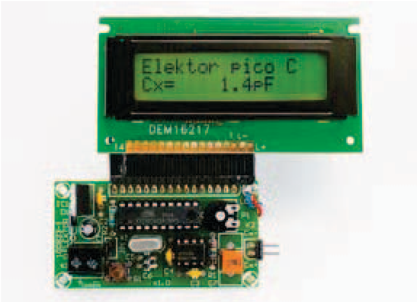
or use the Order Form near the end of the magazine!

elektor

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

RF Frequency and Signal Level Meter

This tester is designed for measurements on RF circuits. The input frequency range is from 50 MHz to 3 GHz, and the signal level range from +10 dBm down to –40 dBm. The measured values appear on a three-line LCD. The frequency measuring part is built around a CPLD chip from Altera, while the level measurement employs a logarithmic detector IC from Linear Technology. All of the digital processing and displaying of data is handled by a DSPIC microcontroller.



Pico-C Meter

Nearly every multimeter of the slightly more elaborate class has a built-in capacitance meter for a wide range of capacitors. Sadly the measurements are rather coarse, and any level of accuracy is hard to discover with capacitances in the picofarad range. Using this simple capacity meter built around an ATtiny microcontroller you can accurately and quickly measure really small capacitance values. Pico-C can measure down 0.1 pF!

Tested and Compared: Infrared Thermometers

IR thermometers are widely used by engineers to reliably measure the temperature of an object from a distance. That's very helpful, but often measurements are not carried out properly, producing large errors confusing the user. In this article we not only describe what to watch out for when buying such an instrument but also tested twenty meters of different brands in a price range up to about £200.

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

Elektor UK/European April 2011 edition: on sale March 17, 2011. *Elektor USA April 2011 edition: published March 14, 2011.*

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

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, DVDs, kits, modules, equipment, tools and books. A powerful search function allows you to search for items and references across the entire website.

Also on the Elektor website:

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

Order Form 03-2011

Description		Price each	Qty.	Total	Order Code
Introduction to Control Engineering	NEW	£27.50			
DVD Elektor 2010	NEW	£17.50			
DVD Wireless Toolbox1	NEW	£28.50			
CD The Power Supply Collection 1		£17.90			
Fundamental Amplifier Techniques with Electron Tubes		£65.00			
ARM Microcontrollers		£29.50			
		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.

Name

Address + Post code

Tel. *Email*

Date – – *Signature*

EL03

METHOD OF PAYMENT

(see reverse before ticking as appropriate)

☐ **Bank transfer**

☐ **Cheque**
(UK-resident customers ONLY)

☐ 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 180
Vernon CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: sales@elektor.com

Subscription Order Form 03-2011

Yes, I am taking out an annual subscription to Elektor and receive the E-book Microcontroller Basics totally free!*

(Please fill in your emailaddress below)

I would like:

☐ **Standard Subscription (11 issues)**

☐ **Subscription-Plus**
(11 issues plus the Elektor Volume 2011 DVD-ROM TIP
+ exclusive access to www.elektor-plus.com)

* *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*

EL03

METHOD OF PAYMENT

(see reverse before ticking as appropriate)

☐ **Bank transfer**

☐ **Cheque**
(UK-resident customers ONLY)

☐ 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. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London.
IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP).
Please ensure your full name and address gets communicated to us.
Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.
GCredit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in this magazine. If difficulties in the supply of components are envisaged, a source will normally be advised in the article. Note, however, that the source(s) given is (are) not exclusive.

TERMS OF BUSINESS

Delivery Although every effort will be made to dispatch your order within 2-3 weeks from receipt of your instructions, we can not guarantee this time scale for all orders. **Returns** Faulty goods or goods sent in error may be returned for replacement or refund, but not before obtaining our **consent.** All goods returned should be packed securely in a padded bag or box, enclosing a covering letter stating the dispatch note number. If the goods are returned because of a mistake on our part, we will refund the return postage. **Damaged goods** Claims for damaged goods must be received at our Brentford office within 10-days (UK); 14-days (Europe) or 21-days (all other countries). **Cancelled orders** All cancelled orders will be subject to a 10% handling charge with a minimum charge of £5.00. **Patents** Patent protection may exist in respect of circuits, devices, components, and so on, described in our books and magazines. Elektor does not accept responsibility or liability for failing to identify such patent or other protection. **Copyright** All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, diskettes and software carriers published in our books and magazines (other than in third-party advertisements) are copyright and may not be reproduced **or transmit**ted 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 2011

SUBSCRIPTION RATES FOR ANNUAL SUBSCRIPTION

	Standard	Plus
United Kingdom & Ireland	**£51.00**	**£63.50**
Surface Mail Rest of the World	**£65.00**	**£77.50**
Airmail Rest of the World	**£82.00**	**£94.50**
USA & Canada	See www.elektor.com/usa for special offers	

HOW TO PAY

Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London.
IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L.
Currency: sterling (UKP). Please ensure your full name and address gets communicated to us.
Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.
Credit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

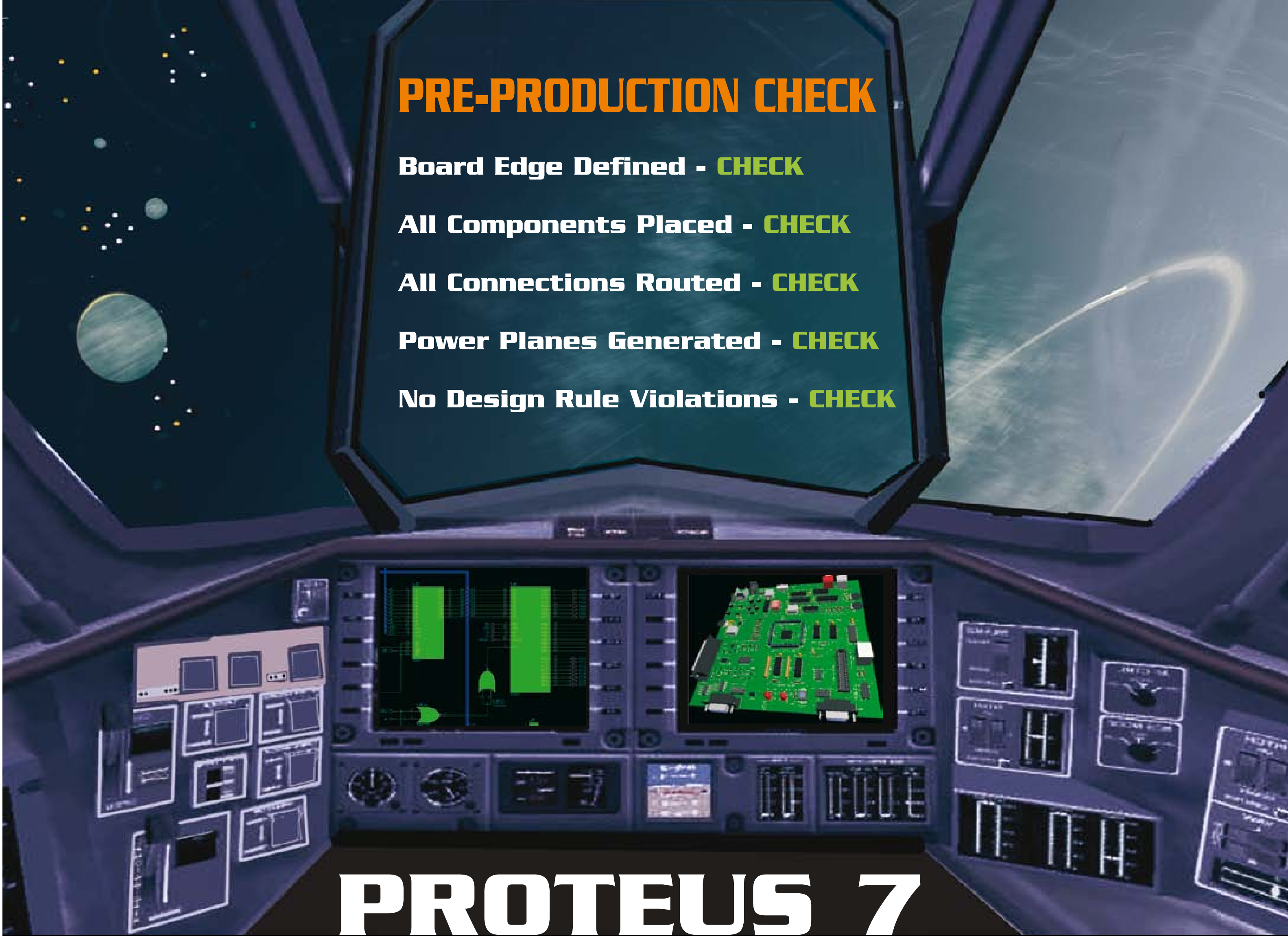
SUBSCRIPTION CONDITIONS

The standard subscription order period is twelve months.
If a permanent change of address during the subscription period means that copies have to be despatched by a more expensive service, no extra charge will be made. Conversely, no refund will be made, nor expiry date extended, if a change of address allows the use of a cheaper service.

Student applications, which qualify for a 20% (twenty per cent) reduction in current rates, must be supported by evidence of studentship signed by the head of the college, school or university faculty.
A standard Student Subscription costs £40.80, a Student Subscription-Plus costs £53.30 (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 2011


PRE-PRODUCTION CHECK
Board Edge Defined - CHECK
All Components Placed - CHECK
All Connections Routed - CHECK
Power Planes Generated - CHECK
No Design Rule Violations - CHECK
PROTEUS 7