December 2011

elektor

www.elektor.com

Android Tablet
Does time-lapse photography

Pick-proof Code Lock
Using 128-bit AES encryption

USB Data Logger
Store serial data conveniently, safely

Electronic
LED Candle
You can blow it out!

LED Cycle Lamp
High-tech on the trail

US $ 7.95 - Canada $ 7.95

0 74470 24965 8

12

Elektor's air intake

One of the most frequent questions I get, usually by email and occasionally by telephone or letter (hey!) is: "can I contribute to your wonderful publication and if so, what are the requirements and the specific subjects you are interested in?" The answer is invariably, "Yes, please, we're ready to evaluate the publication value of your article proposal. Every edition of Elektor covers a wide variety of subjects and fields of interest. Please review our Author Guidelines, they're available under the Service tab on our website at www. elektor.com. To this I usually add a few encouraging words like "we've been around since 1974" and some guidance on the size of the article. Some authors have been with us for many years and are totally at ease with supplying copy and illustrations in the preferred ways; others need extensive assistance with the file formats, spelling issues, style or the depth of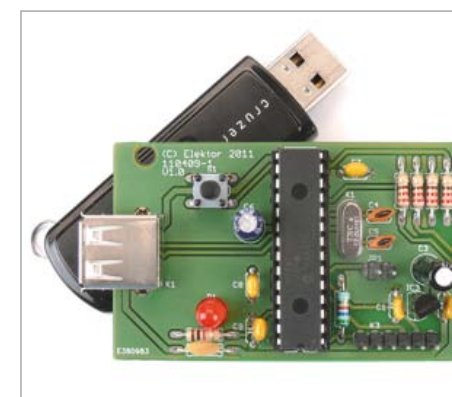 the technical content they can manage. The air intake is actually a manifold — contributions from companies, journalists and specialists in the industry are also welcomed. The approach works well but also creates a queue of articles awaiting publication and in many cases we have to reassure authors that they aren't forgotten and their project is being worked on in the Elektor Lab.

Now, for the solemn bit: about 7 out of 10 article proposals reaching us through all international channels sadly gets rejected for publication. The reasons for the team of editors and designers to be so harsh and unkind to budding authors are diverse: uninventive use of components; *ditto* for obsolete components; ragchewing manufacturer's datasheets or old Elektor articles (!); nebulous circuits nicked from nebulous websites, poor electronic design and attempts to use the magazine to get rid of stock gathering dust. The rest is happily considered for publication or post-engineering by our lab, no matter if the piece is poorly written or the prototype built on breadboard — in general we are good humoured with a keen eye for originality. Even if it takes a while for us to get back to you due to the workload here at Elektor House, give us a try and eventually see your name (and circuit!) in print — it's by no means difficult, we're here to help.

Enjoy reading this edition,
Jan Buiting, Editor

CONTENTS

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

ANALOG • DIGITAL
MICROCONTROLLERS & EMBEDDED
AUDIO • TEST & MEASUREMENT

Elektor PCB Prototyper

➡️ **A professional PCB router with optional extensions!**

NEW!

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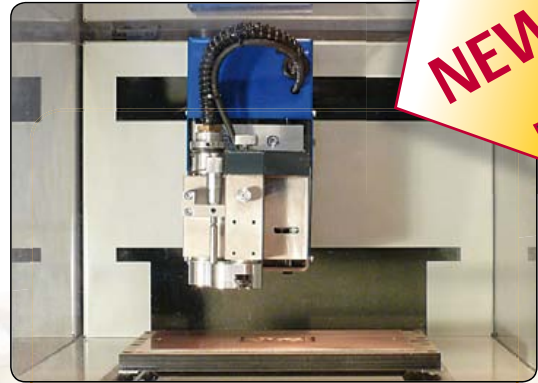
Specifications

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

Ordering

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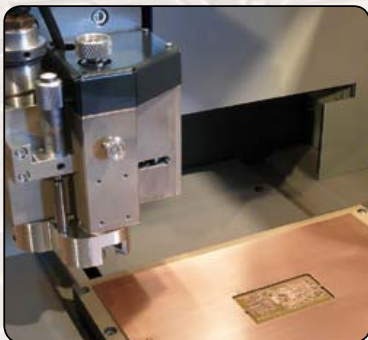
elektor

Further information and ordering at
www.elektor.com/pcbprototyper

DB9 USB-to-digital modules introduced by FTDI

USB solutions specialist Future Technology Devices International Limited (FTDI) has added a series of USB to digital level interface modules, in DB9 form factors, to its product portfolio. The new DB9-USB-D3-F (3.3 V output female connector), DB9-USB-D3-M (3.3 V output male connector), DB9-USB-D5-F (5 V output female connector) and DB9-USB-D5-M (5 V output male connector) enable implementation of USB inter-connection into legacy equipment without the need for alteration of the enclosure or for any noticeable increase in the overall bill of materials.

These modules all utilize FTDI's FT232R USB to serial UART bridge chip, which is capable of supporting data rates of 12 Mbits/s (USB Full Speed). They allow for the quick and easy upgrading of pieces of hardware to today's most widely used serial interface standard in a highly convenient and cost effective manner. Drivers for the modules are available to download direct from the FTDI website.

Supplied in a compact 30.8 mm x 19.8 mm x 11.5 mm package, each of the modules uses a standard USB mini-B connector. They have an operational temperature range of −40 °C to +85 °C, permitting them to be specified for heavy duty industrial applications. The FTDI DB9 to digital modules are each priced at US$15.50 (for 1-9 pcs). The FTDI DB9 to digital modules datasheet is available for downloading.

www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_DB9-USB.pdf   (110675-XVII)

Secure RFID keys for access control, e-Cash, and ID cards worldwide

Maxim Integrated Products (NASDAQ: MXIM) introduces a new line of RFID keys and cards designed for the two-billion-units-per-year automatic identification, access control, and electronic cash (e-cash) markets. This new contactless RFID product family (the MAX66000/020/040/100/120/140) leverages the expertise utilized in the company's popular 1-Wire(R) secure authentication ICs, which protect intellectual property in embedded systems. With a 13.56 MHz interface, these secure keys are ideally positioned to gain market share because 13.56 MHz is becoming the worldwide standard for access control and e-payment applications. Some regions of the world have already begun deploying this RFID technology for passport and national ID cards.

As the critical data found inside many of these RF credentials becomes more and more valuable, efforts to crack, counterfeit, and duplicate cards and credentials will increase. System integrators are already looking for increased security and authentication techniques to protect those assets. Maxim's new RF devices are packaged in a laminated plastic key fob or ISO thin card format and are available in either the ISO 14443B or ISO 15693 HF protocol. Each protocol family offers three products: 64-bit ROM ID only (MAX66000/MAX66100), ROM ID plus 1 Kbit EEPROM (MAX66020/MAX66120), or ROM ID plus 1 Kbit EEPROM and SHA-1 authentication (MAX66040/MAX66140). Custom form factors are also available.

The MAX66040 and MAX66140 employ the secure hash algorithm (SHA-1), a proven technology designed by the NSA for protecting a system's critical data without using expensive encryption techniques or an untested, proprietary protocol. SHA-1 is an ISO standard that is publicly available and has been thoroughly tested in the marketplace. It is designed to maintain the integrity of the stored data so that one can verify the authenticity of any credential. Maxim's RF keys and cards are custom programmable to match the requirements of new and existing tag populations. They work with most 13.56 MHz readers on the market, thus providing an alternative tag source for existing systems.

www.maxim-ic.com/rfid   (110675-VIII)

Granny knows how to keep iPads and Tablets clean

iPads and tablets are sure to appear on holiday gift lists this year again. Music, movies and the internet all available with a simple touch of the screen. And there's one more thing that comes with that simple touch: greasy fingerprints. Thousands of them – all over the screen! Wiping it with a shirt-sleeve isn't effective, and sprays, tissues and cloths are inconvenient and only make the fingerprint mess even worse. Now there's a solution.

SideKick™ by LensPen® is a new screen cleaning tool specially designed to remove fingerprints from iPads and tablet touch-screens. There are no tissues, no cloths, no sprays and no liquids. SideKick features a patented carbon technology that quickly and easily removes the oily fingerprints.

 "It's not high tech, it's old tech," said Peter Meurrens, Vice President of Operations at Parkside Optical and inventor of SideKick. "My grandmother knew how to clean an iPad forty years ago." SideKick's carbon compound is similar to the one found in printer's ink. That carbon compound is why newspapers have been an effective way to clean windows for generations. "When you

get a fingerprint on a lens or a computer screen, it's an accident. But when you get fingerprints on an iPad or tablet, it means you're using it! SideKick is not just an accessory, it's a necessity."

SideKick is available at electronics stores nationwide in the US. Replacement cleaning pads are available in packages of two. Each pad gives 150 cleanings. MSRP for SideKick is $19.95, and $14.95 for the package of two replacement pads.

www.lenspen.com
www.youtube.com/lenspennews
(110675-X)

Cypress: new Gen4 TrueTouch® controller line is ultra noise resilient

Cypress Semiconductor Corp. (NASDAQ: CY) introduced its new Gen4 family of TrueTouch® touchscreen controllers. Gen4 is clainmed to deliver industry-best performance in all categories, including the world's best Signal-to-Noise Ratio (SNR) and unparalleled performance in the presence of all noise sources—the biggest challenge faced by touchscreen designs.

The Gen4 family was designed from the ground up to deliver the world's highest SNR in real world applications. It is the first and only touchscreen IC that delivers built-in 10 V Tx to drive the touch panel at 10 V. Because SNR is directly proportional to the voltage at which the panel is driven, this feature allows Cypress to offer nearly four times the SNR of the next closest competitor.

The Gen4 family further raises the SNR bar as the first touchscreen device family to completely eliminate display noise in hardware. Gen4's patent-pending Display Armor™ offers unprecedented immunity to noise from every type of display, even low

TrueTouch® Gen4 – Rise above the
Revolutionary Touchscreen Solutions from Cypress

cost noisy displays such as ACVCOM LCDs. This allows touchscreen designers to make their products thinner by removing the air gap between the display and the sensor, and also less expensive by removing the shield layer in the sensor. Display Armor allows Gen4 to operate seamlessly with direct lamination, on-cell and in-cell stackups, regardless of the display chemistry.

The Gen4 product family offers the industry's fastest refresh rate of 400Hz, and has the unique ability to scan a capacitive touch panel at 1,000 Hz — both industry-best metrics. This level of performance is enabled by the 32-bit ARM® Cortex™ core at the heart of the Gen4 products. The Gen4 family also provides the industry's best accuracy and linearity of 0.2 mm while boasting the world's lowest active power consumption

of 2 mW, and a deep sleep mode that only draws 1.8 μW with wake-up via address match on the COM port.

Gen4 also offers more capacitive sensing I/O than the competition, while still fitting into the world's smallest touchscreen packages. With up to 40 I/O for mobile phone applications, Gen4 can support up to four standalone CapSense® buttons while still delivering ideal sensor pitch for up to 5-inch screens.

The Gen4 family also offers features that only TrueTouch can deliver, including waterproofing functionality that allows products to meet IP-67 standards; 1-mm stylus support for Asian character sets and accurate handwriting capture; and hover sensing to provide mouseover-like features and true fingernail or thick-glove support in mobile devices.

touch.cypress.com   (110675-XI)
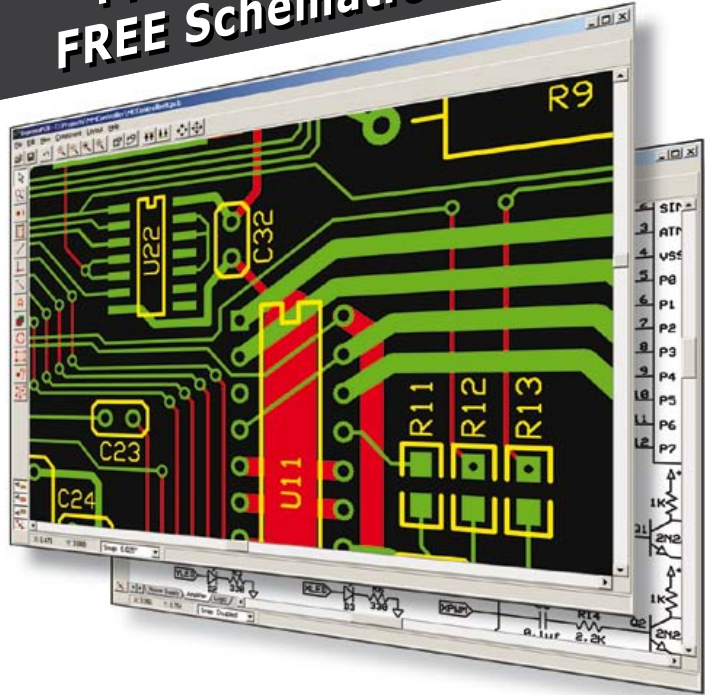
US nationwide rollout of electric vehicle charging stations to begin

Car Charging Group Inc. (OTCBB:CCGI), a provider of electric vehicle (EV) charging services and Central Parking System Inc. and its subsidiary, USA Parking System Inc., the nation's largest parking garage operator, have teamed up to provide EV charging services at locations nationwide in the US via Coulomb Technologies' ChargePoint® Network.

"There are close to 17,000 parking garages in the U.S., and they will play one of the most vital roles in the development of a national EV charging infrastructure," said Brian Golomb, Director of Sales of Car Charging. "By partnering with two of the most important companies in this sector — companies that understand the benefits of electric vehicles —- we will move much quicker in the rollout of this nationwide infrastructure."

Car Charging Group plans to install EV charging stations at sites owned by the two parking garage operators. The operators have 2,200 locations nationwide with 1.1 million parking spaces.

As part of the agreement, Central Parking has the right to purchase five percent of the Common Stock of Car Charging Group.

"We are very excited about this partnership, because it will greatly expand the reach of our nationwide EV charging network," said Michael Farkas, CEO of the Miami Beach, Fla.-based Car Charging Group. "Furthermore, we hope to further enhance our already strong relationship with these companies by giving Central Parking the opportunity to be a shareholder in our company and to take part with us in electrifying the U.S. transportation system."

Central Parking believes that electric vehicles can make a difference in the transportation sector, and they look forward to being a partner in building a nationwide network of EV charging stations.

"Electric vehicles are no longer a mirage — they are becoming an ever increasing presence on our roads and we are proud to be working with such an innovator in the EV sector," said James Marcum, CEO of Central Parking Systems. "By installing EV charging stations in our garages, we will be providing added services to our clients and strengthening our position as an industry leader in environmentally-friendly initiatives."

USA Parking System, a wholly owned subsidiary of the Nashville, Tenn.-based Central Parking System Inc. also aims to benefit both the clients and the environment through its agreement with Car Charging.

Car Charging Group provides EV charging stations at no charge to property owners/managers while retaining ownership, thus allowing their partners to offer their customers, tenants and employees charging services without incurring any outlay of capital. In addition, Car Charging Group's partners realize a percentage of the charging revenue generated by the charging services paid for by the EV owners.

www.CarCharging.com   www.Parking.com   www.USAPark.com   (110675-XIV)

SMT Antenna with comprehensive protection for automotive applications

PREMO launches a new family of its 1103 standard, universally adopted by industry. This format provides up to 55 mV/
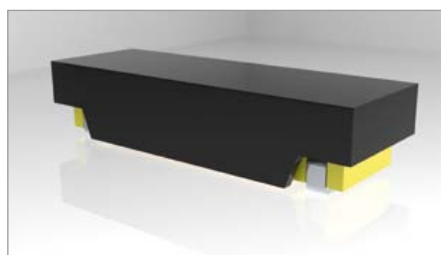
A_{pp}/m sensitivity which gives it the best sensitivity by cm^3 in the market. The new SDTR1103CAP, is a SMD antenna for low frequency 20 kHz-150 kHz receiver applications. This series offers upper and lateral side protection with co-polyamide poly-hexamethylene polymer walls, gamma radiated with high thermal stability (supports up to 290 ºC) and mechanical resistance (exceeds 150 Mpa if mechanical strength). This antenna is equipped with NiZn ferrite core with high surface resistivity (>10 MΩ/mm) that provides a highly stable behavior (rather than ±2%) over a wide temperature range (–40 ºC to 125 ºC).

The new SDTR1103CAP is an SMD antenna with 'Super-Drop-Test-Resistant' technology with an extended range of operating temperature (–40 °C to 125 ºC), which makes it particularly suitable for applications such as TPMS (Tyre Pressure Monitoring Systems) which requires an excellent performance under extreme conditions, according to AEC-Q200 and additional requirements as EU regulations.

PREMO offers four standard values, 2.38 mH, 4.91 mH, 7.2 mH and 9 mH at 125 kHz. Others inductance values and frequencies, from 340 µH to 16 mH, upon request.

Its surface mount (SMT) allows an easy use in the automated process of mounting circuit boards, thus eliminating any manual handling.

www.grupopremo.com/es/file/805
(110675-XIII)

World's first 1 GS/s USB-powered oscilloscope

The three new oscilloscopes in the PicoScope® 2000 Series are the first USB-powered oscilloscopes to offer a real-time sampling rate of 1 GS/s. With two channels, bandwidths ranging from 50 MHz to 200 MHz, a built-in function generator, arbitrary waveform generator and external trigger input, these compact and economical scopes are perfect for engineers and technicians
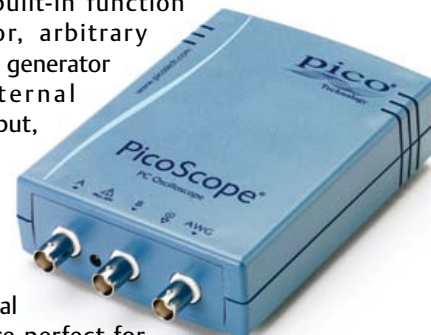
needing a complete test bench in a single unit.

The scopes are supplied with a full version of the PicoScope oscilloscope software. As well as standard oscilloscope and spectrum analyzer functions, PicoScope includes valuable additional features such as serial decoding, mask limit testing, segmented memory and advanced triggers. It provides a large, clear display that shows waveforms in great detail and allows easy zooming and panning. Other advanced features include intensity- and color-coded persistence displays, math channels, automatic measurements with statistics, and decoding of I²C, UART/RS232, SPI and CAN bus data. Free updates to the software are released frequently.

Like all PicoScope oscilloscopes, the new PicoScope 2000 Series devices use digital triggering, which ensures lower jitter, greater accuracy and higher voltage resolution than the analog triggers found on many other scopes. The advanced trigger types include pulse width, interval, window, window pulse width, level dropout, window dropout, runt pulse, variable hysteresis, and logic.

A Software Development Kit (SDK), supplied free, allows you to control the new scopes from your own custom applications. The SDK includes example programs in C, C++, Excel and LabVIEW. The SDK and PicoScope are compatible with Microsoft Windows XP, Vista and Windows 7.

The new PicoScope 2000 Series oscilloscopes are available now from Pico distributors worldwide and from www.picotech.com. Prices range from £349 for the 50 MHz PicoScope 2206 to £599 for the 200 MHz PicoScope 2208, including a 5-year warranty.

www.picotech.com   (110698-I)

New EasyPIC v7 development system

For the first time in EasyPIC's almost 10-year history, EasyPIC v7 has grouped PORT headers, LEDs and Buttons into Input-Output groups, making them easier to use than ever before. The v7 boards come equipped with tri-state DIP switches, so connecting pull-up or pull-down jumpers to desired pins is now just a matter of pushing the switch. Connectivity is the main focus of EasyPIC v7, providing three separate PORT headers in the Input-Output groups
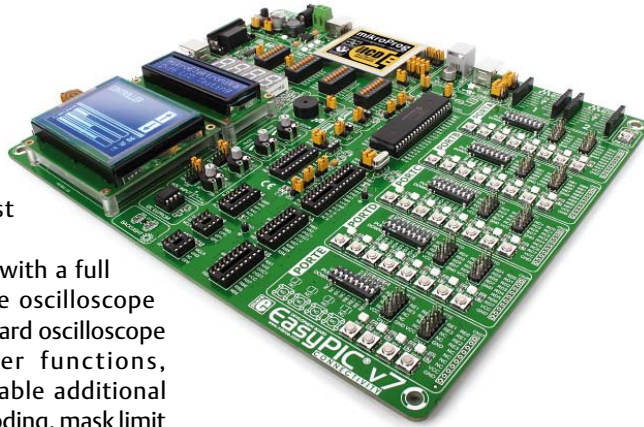
and another one on the opposite side of the board, allowing users to access those pins from any side.

The new board has a dual power supply supporting both 3.3 V and 5 V microcontrollers. It's like having two boards instead of one!

Probably the best feature of the v7 board is its powerful on-board mikroProg programmer and In-Circuit debugger capable of programming over 250 PIC microcontrollers. Debugging is supported with all mikroElektronika PIC compilers — mikroC, mikroBasic and mikroPascal.

7-segment displays have returned at the request of many users, which brings the number of displays on the board to three: GLCD 128x64, LCD 2x16 character and 4-digit 7-segment displays.

EasyPIC v7 is the first board supporting the mikroBUS pinout standard. Mikroelektronika are preparing many mikroBUS compatible Click Boards, which will make development easier then ever. No configuration or jumpering, just plug-n-play.

The new board has the following new modules: Serial EEPROM, Piezo Buzzer and support for both DS1820, and LM35 Temperature sensors.

The EasyPIC v7 User Manuals and schematics haven been redesigned; the documentation is now more informative with a good number of clear photos and expanded text sections.

www.mikroe.com   (110698-II)

Sensirion: first digital temperature sensor

Following the successful market launch of the SHT2x family of humidity and temperature sensors, Sensirion is now launching a sensor designed exclusively for temperature measurement. The new STS21 temperature sensor is based on the same chip as the SHT2x family and is housed in a tiny DFN package measuring only 3x3 mm. This makes the sensor ideal for use in

applications where only very limited space is available. It also delivers outstanding performance and remarkably high accuracy of ±0.22 °C over the temperature range of 5 to 60 °C, combined with very low power consumption — an especially important consideration for battery-operated devices. The STS21 is pin and protocol compatible with Sensirion's standard SHT21 humidity sensor, but it has a different I²C address. This allows the STS21 to be used as alternative to the SHT21 in applications where temperature measurement is required and humidity measurement is optional, since the same hardware can be used with both sensors.

The STS21 temperature sensor is fully calibrated, provides a digital output, and is very competitively priced. It therefore fulfills the stringent requirements of many applications in terms of both performance and cost effectiveness.

www.sensirion.com/sts21-datasheet

(110698-III)

132 LED driver features industry's highest efficiency and smallest size

Austriamicrosystems has announced the AS1130, the most advanced and smallest dot-matrix LED driver (channels vs. PCB space) available. The AS1130 drives 132 LEDs but requires only 5 mm² PCB space, reduces external component count, allows use of cheap connectors and requires fewer PCB (printed circuit board) layers. Benefits for end users include up to 80% longer battery lifetime, more colorful effects and smoother running animations.

Using a 12x11 cross-plexed technique, austriamicrosystems' AS1130 LED driver is targeted for dot-matrix displays in mobile phones, toys, small LED displays in personal electronics, but also non-battery powered household goods, indoor public information displays, and industrial applications such as power meters. The AS1130 drives 132 LEDs, each with an 8-bit dimming control and no external resistor required. Additionally, an 8-bit analog current control allows fine tuning of each current source to compensate for different brightness of different colors, or to adjust the white balance on RGB LEDs. austriamicrosystems' AS1130 incorporates 36 frames of memory for small animations or for use as a buffer to

reduce host processor load, saving energy and processing time. The AS1130 LED driver can also extend battery life by controlling an external power supply (e.g. charge pump) which is required when LEDs need a higher voltage than the battery can supply, allowing continuous operation even under low battery voltage conditions.

"The AS1130 dot-matrix LED driver is designed to make driving LEDs an easier task," commented Rene Wutte, austriamicrosystems' Marketing Manager for Lighting. "It enables driving a large number of RGB LEDs from one IC for creative light designs while providing the highest efficiency available, an important feature for both battery-powered and AC-powered applications. The AS1130's features simplify design and programming, optimize total cost, and allow developers to provide the lighting features required to stay
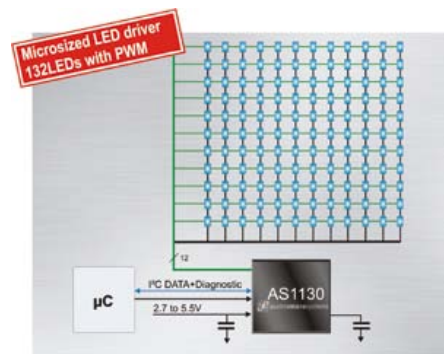
ahead in this market."

In addition to the ultra-small sized CS-WLP-20, the AS1130 LED driver is also available in a gull winged SSOP-28 package, allowing easier handling in applications that are not so space sensitive. This makes the AS1130 a perfect replacement for indoor high pixel density video walls, easily replacing up to eight 16-channel PWM LED drivers, or reducing the complexity of externally (user designed) multiplexed systems.

Only 12 lines are required to drive all 132 LEDs. This is accomplished with austriamicrosystems' multiplexing technique called cross-plexing. It reduces line count on the PCB as well as pins on the connectors, saving space & costs. Other features include control via a 1 MHz I²C compatible interface, open and shorted LED error detection, and low-power shutdown current.

The AS1130 LED driver operates over a temperature range of –40 to +85°C and a wide 2.7 to 5.5 V power supply range.

www.austriamicrosystems.com/LED-driver/
AS1130

(110698-IV)

Vector Fabrics Joins ARM Connected Community

Vector Fabrics recently announced it is now a member of the ARM® Connected Community®, the industry's largest ecosystem of ARM technology-based products and services. As part of the ARM Connected Community, Vector Fabrics will gain access to a full range of resources to help it market and deploy innovative solutions that will enable developers to get their ARM Powered® products to market faster.

Mike Beunder, CEO at Vector Fabrics: "Many of our customers have adopted multicore ARM-based architectures and use our vfEmbedded tool to optimize their software applications for it. That's why vfEmbedded already provides support for the ARM CortexTM-A series of multicore applications processors, including the ARM NEONTM technology. We're excited to be working more closely with ARM to ensure our tools integrate well with the ARM architecture."

The vfEmbedded multicore development tool allows developers to unlock the performance potential of the multicore high-performance ARM Cortex-A architectures. Optimizing software for multicore processors by hand is simply too complex, is error prone, takes too much time, and won't result in an optimal implementation. VfEmbedded thoroughly analyzes the program code, predicts parallel performance and swiftly guides the developer toward an optimal multicore implementation that is free of errors.

The ARM Connected Community is a global network of companies aligned to provide a complete solution, from design to manufacture and end use, for products based on the ARM architecture. ARM offers a variety of resources to Community members, including promotional programs and peer-networking opportunities that enable a variety of ARM Partners to come together to provide end-to-end customer solutions. Visitors to the ARM Connected Community have the ability to contact

members directly through the website.
"The Connected Community is all about companies working together to provide the most complete solutions in the shortest possible time. By joining the Community, which now comprises more than 850 companies, Vector Fabrics increases the large portfolio of skills, products and services that are centered around the ARM architecture, and currently available to developers worldwide," said Lori Kate Smith, Sr. Manager Community Programs for ARM.

Vector Fabrics' vfEmbedded software development tool addresses the hard problem of partitioning and mapping software onto heterogeneous multicore SoC platforms. VfEmbedded lets you model your embedded platform, then takes you through the process of analyzing, parallelizing, and implementing your code. It is the only tool that gives you the precise information you need to make sure that the parallelized software works correctly and has the highest performance on your multicore machine.

http://cc.arm.com
www.vectorfabrics.com/products/vfembedded
(110698-V)

Parallax: laser range finder module

Designed in conjunction with Grand Idea Studio, the Parallax Laser Range Finder (LRF) Module is a distance-measuring instrument that uses laser technology to calculate the distance to a targeted object. The design uses a Propeller processor, CMOS camera, and laser diode to create a low-cost laser range finder. Distance to a targeted object is calculated by optical triangulation using simple trigonometry between the centroid of laser light, camera, and object.
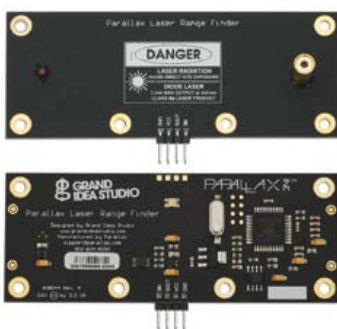
KNX-RF Multi module for building automation

Radiocrafts AS now expand their product line with a new module complying with the **KNX-RF Multi** specification. KNX is the only open international standard for Home and Building Control, used in Smart Home, Building Automation and Building Management Systems.

RC1180-KNX2 is the world's first RF module including a complete KNX-RF Multi protocol stack. The KNX-RF Multi is an extension of the European Norm for building automation adding redundancy and increased reliability. The main features of the new Multi standard is; frequency agility by using up to 5 frequencies, fast link acknowledgement of up to 64 receivers with automatic retransmission, and multi-hop repeaters extending the range by two hops. Battery operated transmitters and receivers are also supported by the new standard. The embedded protocol is backward compatible with KNX-RF 1.1 and KNX Ready, and can be used for unidirectional and bidirectional devices.

The new module is designed for sensors, actuators and other home and building automation equipment. Due to its small size, easy to use interface, complete embedded protocol and low power consumption, it can easily be integrated into any product making a very cost efficient solution.
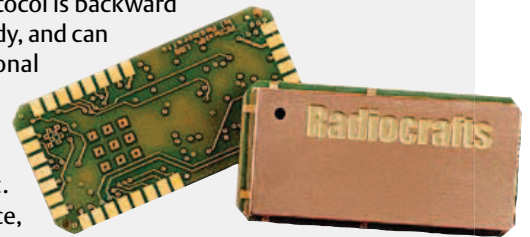
Radiocrafts is a member of the KNX Association and has participated in the development of the new standard. KNX is one of the leading standards for home and building control. The interest for such systems is increasing, meeting the demands for energy saving technology. Studies have shown up to 50% energy savings using KNX technology.

Radiocrafts is also considered as one of the leaders in Wireless M-Bus technology, and the new KNX product series gives the customers an easy transition to KNX with compatible products. Interoperation between Wireless M-Bus and KNX-RF is also possible using the new module.

The module operates in the 868 MHz band, using Listen Before Talk (LBT) and frequency agility to reduce collisions. Up to 5 frequencies are scanned and automatically selected. One receiver can be linked with up to 64 transmitters, enabling very large RF networks. The fast acknowledge and retransmission feature ensure link reliability. Complete repeater functionality is also built in the protocol stack, and can retransmit messages in two hops. It can be used with S, A and E modes of installation. Among other features, the module offer automatic battery supervision and signal strength information.

The new RC1180-KNX2 is a surface-mounted high performance transceiver module measuring only 12.7 x 25.4 x 3.3 mm. A UART interface is used for serial communication and configuration. An antenna is connected directly to the RF pin. The RC1180-KNX2 module is certified for operation under the European radio regulations for license-free use. When used with quarter-wave antennas a line-of-sight range of 800 meter can be achieved.

The module and Demo Kits are available now. The module is delivered on tape and reel for volume production.

www.radiocrafts.com (110698-VI)

Features:
- Compact module with integrated CMOS camera and laser system;
- Optimal measurement range of 6–48 inches (15–122 cm) with an accuracy error <5%, average 3%;
- Maximum object detection distance of approx. 8 feet (2.4 meters);
- Range finding sample rate of 1 Hz;
- Single row, 4-pin, 0.1" header for easy connection to a host system.

The Laser Range Finder is priced at $129.99 and available direct from Parallax, or its national distributors.

www.parallax.com, search 28044
(110698-VII)

The PCB Prototyper in Practice
Perfect for prototypes and small batch production

The PCB Prototyper introduced a year ago by Elektor has now found a place in many labs and companies. This machine allows users who do not have extensive experience with milling machines to mill single-sided or double-sided PCBs quickly and easily. We visited a PCB Prototyper user to learn about their experience with the machine.

By Harry Baggen
(Elektor Netherlands Editorial)

After Elektor presented the PCB Prototyper in December 2010, it didn't take long for the first orders to come in. The manufacturer (Colinbus) was already busy with the first production batch, and after a while the first units were delivered. The responses from users proved to be extremely positive. They hadn't expected to be able to buy such a precise, easy to use machine at a budget price — after all, PCB milling machines are usually much more expensive than the 3,500 euro (ex VAT) price tag of the PCB Prototyper.

Here in the Elektor editorial office we were wondering how this machine is used in practice, so we arranged a visit to Avasto in Oudewater (The Netherlands), a firm that

has been using a PCB Prototyper for a while. Avasto is a multifaceted facility services contractor that is active in construction work, automation of production processes, and the design and maintenance of saunas, whirlpools and the like. Electronic controllers are used in various projects, often based on PLCs. However, this has also involved more and more in-house development of electronic devices in the last while. Their latest product is a slide safety device for swimming pools. This system provides a signal at the top of the slide to indicate when the next swimmer can jump onto the slide. The swimmer can also press a button to start a time measurement. After exiting the slide, the swimmer presses a 'Finish' button in the catch basin, and the elapsed time is displayed on a scoreboard. In addition to preventing blockage of the slide, this system introduces a competitive element that makes the slide more attractive.

Avasto developed the entire system in-house and has already installed several systems. The system is approved by the Dutch Keurmerkinstituut, which is responsible for assessing and certifying product safety. Co-owner Swen van Vrouwerff is a dyed-in-the-wool technology buff who knows absolutely everything about the projects underway in his company, including their mechanical, electrical and electronics aspects. Nowadays the company is developing more and more electronics devices in-house. When the PCB Prototyper was presented in Elektor, he thought it would be the perfect machine for his company, since it would allow them to make PCBs quickly for small product volumes. Although ordering small quantities of PCBs from a PCB manufacturer is also possible, this takes more time and is relatively expensive. Swen is convinced that the investment will pay for itself quickly.
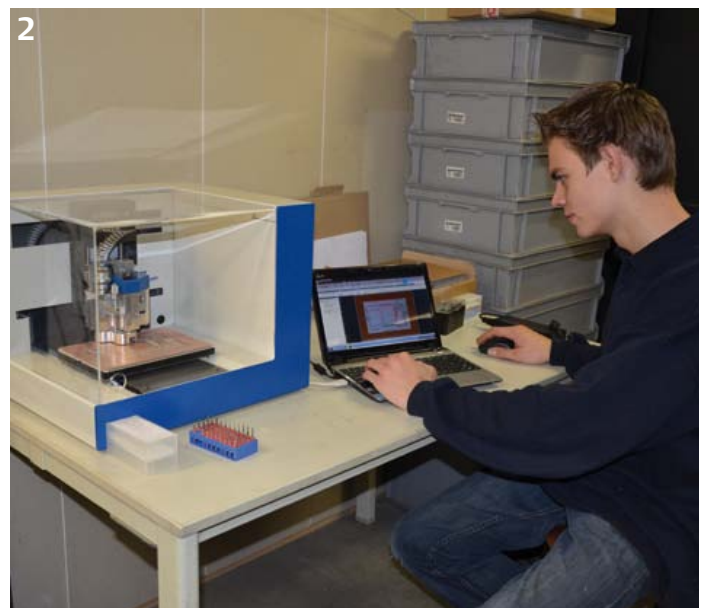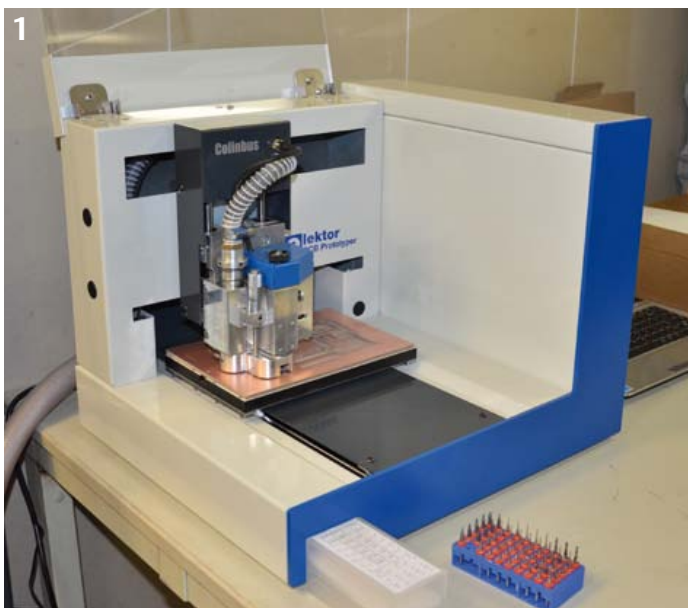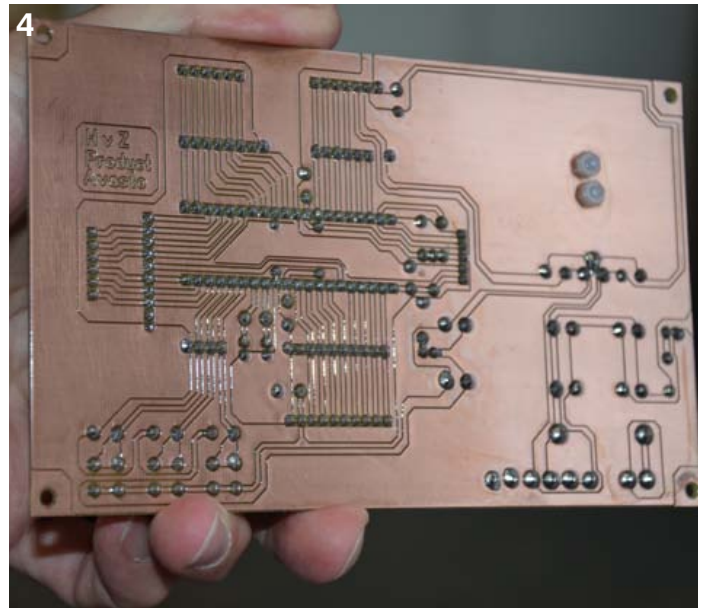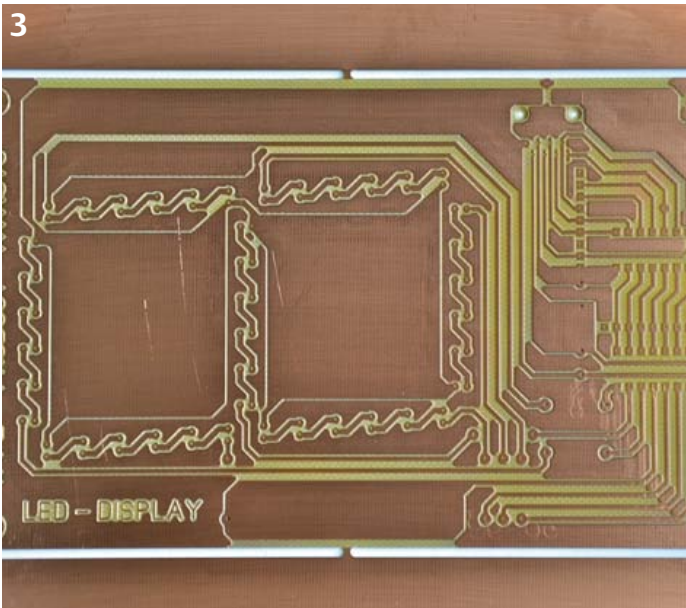
Right now the PCB Prototyper is primarily being used to produce PCBs for the slide safety system described above. All of the circuitry for the system, distributed over nine PCBs, is housed in the robust enclosure of the display unit. All of the PCBs are made on the PCB Prototyper (photo 1). It is controlled by a netbook located next to the machine (photo 2). It's worth mentioning that the user interface for the PCB Prototyper and the structure of the entire slide safety system were designed by two young employees with intermediate vocational school education. They told us that the PCB Prototyper is very easy to use. After spending a day trying it out, they were able to use it properly and they managed to mill top-quality PCBs. Producing a PCB with roughly Eurocard dimensions takes around half an hour. The machine stops automatically when it's time to change the drill bit or milling cutter, so you don't have to be there all the time and you can do other jobs in the meantime. Photo 3 shows the end result — in this case a PCB for a display segment. The PCBs from the machine are assembled and then coated with a thick plastic film on the copper side (photo 4) to protect the copper against the effects of chlorine, which is abundantly present near swimming pools. Photo 5 illustrates the quality of the milled tracks on the PCB. Finally, photo 6 shows the enclosure of the slide safety system with the fitted boards.

This is just one example of the many potential uses of the PCB Prototyper. Presently Avasto produces mainly single-sided PCBs for leaded components, but if the company wishes to switch to SMDs in the future, the PCB Prototyper can easily mill PCBs for them as well. Furthermore, a variety of extension options for the PCB Prototyper to make it even more versatile will be available in the near future.

(110694)

Internet Links

www.elektor.com/projects/
    pcb-prototyper-(100619).1599728.lynkx

www.avasto.nl
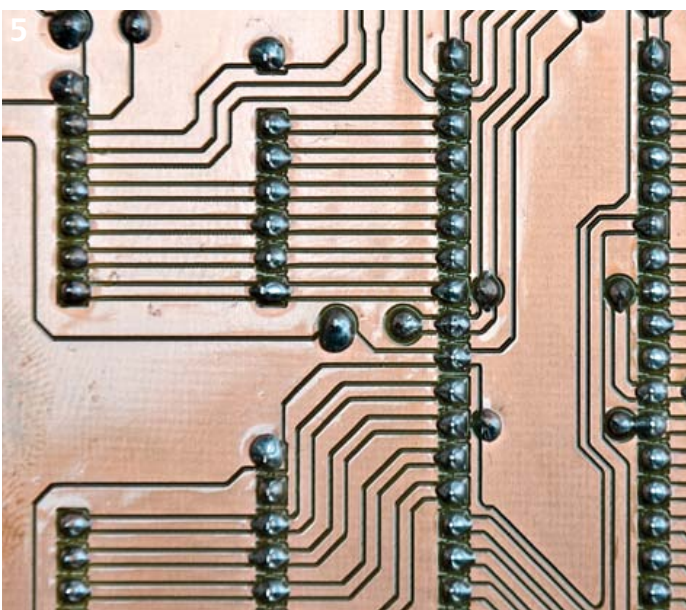
Android as a Development Platform

Using low-cost tablet PCs in embedded applications

By Elbert Jan van Veldhuizen (The Netherlands)

Electronics designers are accustomed to developing their systems from scratch or using development boards. An interesting alternative is to use a tablet PC as an embedded device.

Tablet computers (PCs) running the Android operating system are now available for under 100 dollars. They are packed full of electronics and are visually attractive. Many functions that cost considerable effort to implement in an embedded environment are standard features in tablet computers. In this article we examine the factors involved in using tablets in electronics projects.

Hardware

What can you expect to find in a tablet computer priced below 100 pounds/euros? To start with, a nicely finished enclosure. A tablet is perfectly at home in you living room or mounted on the dashboard of your car. It can also be fitted on the front panel of an enclosure to serve as a control panel with a professional look, which is one of the major potential uses.

A tablet has a backlit touchscreen display, which make it an ideal platform for a user interface, and in particular a graphic user interface. The combination of a screen and a touchpad allows you to create virtual buttons (which can even be context sensitive) and entry fields, and you can present elaborate menu structures. Tablets are also suitable for multimedia applications, allowing you to display live imagery from a camera (with or without an IP interface) or play instruction videos. The displays of the low-cost tablets have diagonal dimensions of 7 to 10 inches (18 to 25 cm) and resolutions of 800 x 480 to 1024 x 600 pixels. Most low-cost models have a resistive touchscreen, which allows only one touch at a time, but this is generally good enough for our purposes.

You might be thinking that tablets can only be used as a sort of dumb terminal, but this is by no means true. Even the low-cost models have an ARM11 processor with over 600 MIPS of computing power. This is more than 10 to 100 times faster than the microcontrollers we normally used in our designs, and it means that tasks that need a lot of processing power can run without major problems on a tablet. Some examples include computations that are normally executed by a DSP, such as Fourier transforms and filtering. Android is not a real time operating system, so a tablet cannot be used for real-time computations. However, Android can handle near real time tasks with response times in the seconds range.

Programming Android apps was described in detail in this year's June 2011 edition [1]. Android is a multitasking operating system, which means that different applications for different purposes can run in parallel.

Another important aspect of tablet computers is Internet connectivity. All tablets have a WiFi port for connecting to the Internet. Various Internet apps are available for Android devices, including a web server, an FTP server and a variety of e-mail applications. These apps are able to run concurrently, including in the background. This makes it easy to use an Internet connection to view or download data remotely or to control electronic devices remotely. Some tablets (typically the more expensive models) also have a 3G mobile telecommunication port. This allows them to be used in areas outside the range of a WiFi router, as long as mobile phone coverage is available. Tablets have integrated non-volatile (flash) memory, and most of them also have a micro-SD slot for memory expansion. This allows up to 32 GB of memory (with current devices) to be added. A typical application for this is data logging. This amount of memory is sufficient to store a year's worth of data at a data rate of 1000 samples per second. This memory can also be used for the previously mentioned multimedia applications.

Furthermore, most tablets have speakers (as well as an audio out connector), a microphone and a webcam built in. Tablets can provide a regulated 5 V supply voltage (from the USB port), taken from the integrated (or replaceable) rechargeable battery. Although this is not the main reason for using a tablet in an electronics project, it's a handy bonus.

Incidentally, tablets are not the only Android devices worth considering. Android

Figure 1. USB Micro B to Type A bus cable for use with a tablet operating in host mode.

smartphones are also available for under 150 dollars. Although they have smaller displays, they are equipped with 3G functionality as standard. If 3G functionality is more important than display size, an Android smartphone is an excellent choice. These smartphones also support Bluetooth wireless communication. The Amarino project [2] uses the Bluetooth interface to link an Arduino board to an Android smartphone.

USB interface

Every tablet has a USB port, which appears to be the logical way to connect the tablet to external circuitry. In theory this is a simple task, but in practice it is full of pitfalls. First of all, you need to realize that Android tablets have evolved from Android mobile phones. The USB port of a mobile phone acts as a USB device ('slave' mode) and is intended to be used for purposes such as connecting the phone to a PC, and the USB ports of Android tablets are also device ports. This means that the connected circuitry must acts as a USB host ('master' mode), which generally requires a relatively complex USB controller. However, many tablets are able to switch the USB port to host mode (which requires a special conversion cable) or have a second port that operates in host mode. This port is intended to be used for connecting a mouse, keyboard or memory stick. Simple USB controllers with a USB device port can also be con-

nected to this second port. Microchip provides code that can communicate with an Android unit configured as a USB device. A number of development boards are also available [3].

In addition to the hardware, you need the right driver software, which is not so easy. Android does not have any standard support for USB devices other than those mentioned above. Although the most recent versions of Androids (2.3.4 and 3.1) have enhanced capabilities, using them is not straightforward. In order to install the right drivers, you must first install a modified version of Android. This requires not only the software (the 'ROM'), but also root access permission in the tablet. Apps that can provide this access are available for low-cost tablets with a standard Android installation, but creating a new ROM is a task for advanced users, and it is essential to have all of the necessary drivers available. This is a lot of work for a single project, but if a particular tablet can be used in various projects, it may be worth considering.

Fortunately, it appears that this problem will be remedied in the future. Google, the maker of Android, is hard at work on version 4 of Android, named Ice Cream Sandwich, which combines version 2 (for mobile devices, including low-cost tablets) and version 3 (for fancier tablets). One of Google's

explicit goals is to allow Android devices to communicate with many different types of accessories. The software class 'accessory' [4] has been developed specifically for this purpose. Although this functionality is implemented in Android versions 2.3.4 and 3.1, there is no standard support for it in the ROM. The USB port is configured as a host with the new class, which allows simple USB controllers to be connected to the port. This simplifies the connection of USB devices to Android tablets. Support for this will probably be provided in the standard ROM version, eliminating the need for upgrades. Android version 4 is expected to be available in late 2011. The system requirements for the new version exceed the capabilities of current low-cost tablets. It will probably take a while before low-cost tablets that support this version become available, since the prices of processors with sufficient computing power are presently too high.

Architecture

If you use a tablet in an electronics project, the place where you put the software that provides the 'intelligence' (either the tablet or the external circuitry circuitry) is a significant architectural issue. The advantage of putting the intelligence in the tablet is that you can work in a well defined programming environment with an API that provides direct access to the various components of the tablet. The tablet also has high process-
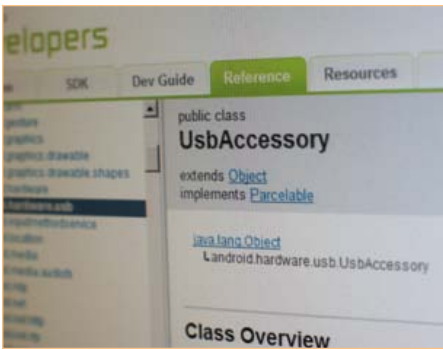
Figure 2. The Android API is extensively documented.

ing power and large memory capacity. However, real-time applications are not possible. The main reason for placing the intelligence in the connected microcontroller is that the application requires real-time control of the circuitry and/or short response times.

In practice a hybrid solution will usually be the most appropriate choice, with the app that runs on the tablet handing the tasks closely related to the peripheral devices of the tablet, while the connected microcontroller provides direct control of the I/O.

Applications

To give you an idea of the realm of possibilities, here we outline a number of potential applications and briefly describe the neces-

Figure 3. Using a tablet in a home automation system.

sary architectures:

• Data logger with configuration settings entered over the GUI. The data can viewed in graphic form on the GUI or on a web page. The raw data can be uploaded over the Web or via FTP. The hardware requirement for this is an ADC connected to the USB port. On the software side, three apps would be necessary: a web server and an FTP server (both of which are available as standard apps), and a user-written app for reading and storing the data — plus a configuration settings screen and graphic output.

• Telemetry from a (race) car, with live data displayed on the GUI and 3G link capability for reading out data in real time and storing it on a flash memory card. This requires a USB to OBD2 interface and software to handle the interface, configuration settings, and data readout and storage. If the data only needs to be read out once per lap, a standard FTP server app would be sufficient. If real-time or near real-time readout is required, a new app must be developed for this purpose (perhaps based on a serial communication app).

• Home automation: controlling a solar water heater and a central heating system with the aid of weather forecasts on the Web, with remote control via a website and a GUI on the tablet. This requires the same configuration as the data logger mentioned above. An additional feature here is that decisions regarding how the solar water heater is used can be taken based on information available on the Web, such as the forecast hours of sunshine and outdoor temperatures. This requires writing a special app to obtain this information, for example by reading RSS feeds and extracting the necessary data.

• Digital oscilloscope with settings and waveform display on the GUI, and potentially the ability to send oscillograms by e-mail. The configuration described for the data logger is necessary here as well. In this case this case the majority of the intelligence must be located in the connected circuitry due to the required signal speeds.

• A coffee machine control panel that sends

an e-mail to the maintenance department when the coffee needs to be refilled or a problem occurs. Here the tablet can serve as the heart of the system, but I/O circuitry is necessary for controlling and reading data from all of the components.

• A standard maintenance console. A tablet is very suitable for use as an external console for troubleshooting, configuration and maintenance. In its simplest form it can communicate with the target system over the USB port, using a terminal emulator app on the tablet. This could be used to view start-up messages or display the commands sent from the terminal. A more complex option is a console connected over a USB to JTAG converter. Thanks to its low cost, a console of this sort could kept on hand at every customer site.

• Prototyping: the ease of writing apps makes a tablet suitable for developing prototypes and proof-of-concept designs quickly and efficiently. Thanks to the large number of standard functions in Android and the wealth of available apps, considerable functionality can be deployed in a very short time.

Conclusion

Low-cost tablet computers can be effective, affordable and quickly implementable components of electronic projects. In addition to a visually attractive design, they have many readily accessible functions. At present it takes a good deal of effort to use the tablet's USB port for connection to external circuitry, but with the advent of Android version 4 in the coming year this will become easier. This will open the way for a host of new applications.

(110667)

Internet Links

[1] www.elektor.com/110265

[2] www.amarino-toolkit.net

[3] www.microchip.com/android

[4] http://developer.android.com/guide/
     topics/usb/accessory.html

Elektor Academy Webinars in partnership with element14

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

Webinar programme:

Let's Build a Chaos Generator
Date: Thursday December 15, 2011
Time: 15:00 GMT (16:00 CET)
Presenters: Maarten Ambaum and R. Giles Harrison (Reading University)

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

Here comes The Elektor Bus!
Date: Thursday January 19, 2012
Time: 15:00 GMT (16:00 CET)
Presenter: Jens Nickel (Elektor)

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

Now available to view on demand at www.element14.com:

E-Blocks, Twitter and the Sailing Club
Presenters: Ben Rowland and John Dobson (Matrix Multimedia)

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

Platino – an ultra-versatile platform for AVR microcontroller circuits
Presenter: Clemens Valens (Elektor)

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

FREE PARTICIPATION
Register now!

elektor ACADEMY
the school of electronics

element 14
www.element14.com

Time-lapse Photography with an Android Tablet

Using a servo-operated shutter release

By Elbert Jan van Veldhuizen (The Netherlands)

With an Android tablet and a handful of hardware, you can put together a remote control for a still camera for time-lapse photography. It also allows the picture settings to be configured from a web browser on a PC, which sends to them to the tablet via the local WiFi network.

The design described here can be used with a still camera to cause it to take pictures at regular intervals. If you make a film from these pictures, the result is what is called a time-lapse film in which hours or even days are reduced to a few seconds. A similar project for cameras with an external shutter release input was published in a previous edition of Elektor. The product described here operates the camera button mechanically using a servo of the sort used in RC (remote controlled) models.

A special feature of this project is that an inexpensive Android tablet (as described elsewhere in this edition is used to provide the GUI (graphic user interface) and enable operation over the Internet. This article is handy for learning more about the basic building blocks for programming Android tablets in embedded systems.

Design and user interface

This design is intended to operate a still camera mechanically with the aid of an RC servo, which makes it usable for all types of still cameras. The author first tried to access the shutter button leads of a compact digital camera in order to operate the camera electronically, but the components are so small that it is virtually impossible to do this without destroying the camera.

As can be seen in **Figure 1**, the servo actuates a lever with a plunger that presses the shutter button on the camera. Other mechanical arrangements for operating

Figure 1. A prototype put together with a few Meccano parts.
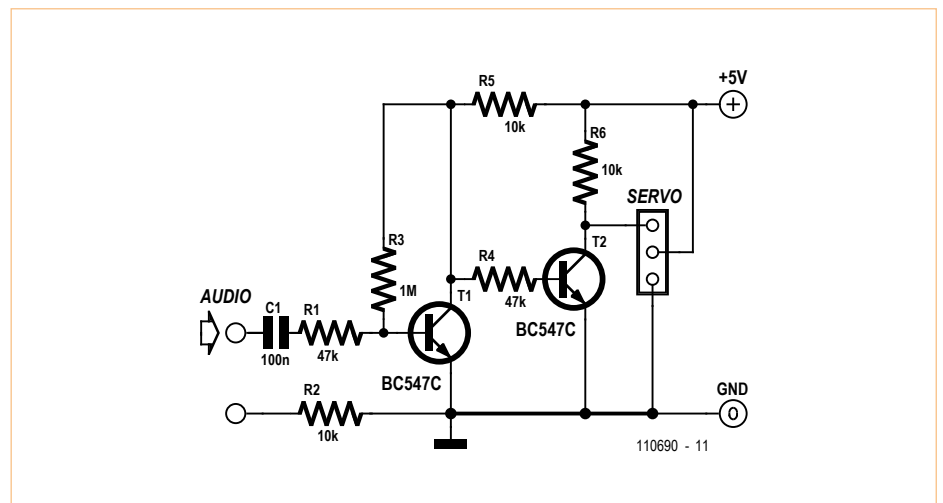
Figure 2. This circuit boosts the audio output signal to 5 V TTL level. A small 5-V AC power adapter provides an adequate source of power.

the shutter button with a servo are also conceivable.

As described in the article on Android tablets elsewhere in this issue, it is presently not easy to use the USB port of an Android tablet. In practical terms, doing so would mean that for each different type of tablet, different instructions and software would be necessary to make the design described here work properly. For this reason, the author decided to use the audio output to drive the servo. The position of a servo is determined by pulses having a width between 1 and 2 ms, or with some servos between 0.5 and 1.75 ms. These pulses are repeated every 20 ms. A pulse signal of this sort can be generated directly using the tablet's audio output. The pulse amplitude at the audio output is typically under 1 V, so this signal needs to be converted to TTL level. This is done by the circuit shown in **Figure 2**. Transistors T1 and T2 amplify the signal. As the zero level of the audio output signal is not necessary the same as the circuit ground level (it may be floating), quasi-DC coupling is provided by R1 and C1. None of the component values here is critical.

If no modifications to the software are necessary, the program can be installed by copying the file `timeLapse.apk` (in the binary folder of the zip file) to the tablet over the USB port or from a flash memory card. Sending the file to the tablet as an e-mail attachment doesn't work. The installation program included with the tablet must be used to install the program. Under Settings / Application settings / Unknown sources, enable 'Allow installation of non-market applications'.

The circuit can be operated in two different ways: either via a GUI on the tablet (see **Figure 3**) or via an external web interface (see **Figure 4**). On the GUI the number of pictures and the time between pictures (in seconds, with a minimum value 3 s) can be best in the two text boxes. Three different servo positions can be set with the three

sliders. The first position is with the camera at rest; the second position is for pressing the shutter button halfway for focusing; and the third position is for taking a picture. Check to make sure that the slider settings are right, since the position between 0.5 ms and 2 ms varies. For some servos this is outside the working range, and they may be damaged if they are driven with corresponding pulses for a prolonged time.

The Phase button selects either positive or negative pulse output. Depending on the tablet, it may be necessary to invert the pulses (this doesn't matter for audio output). Finally, the Start button starts and stops a time-lapse sequence.

The application can also be operated over the web. It listens to port 8090 at the IP address of the tablet. Start by accessing the tablet from your browser (on another computer), for example by entering the address `http://192.168.1.101:8090/`. The current status will be shown. New values can be entered in the form, and the time-lapse sequence can be started and stopped with the links. The logo on the web page (at the top left) must be entered by the user by placing a `logo.jpg` file in the `/scard/webserver_TL` folder on the tablet. This folder is created automatically when the application is launched for the first time.

GUI programming

The basic aspects of programming Android devices and the necessary resources were described in the June 2011 edition. The author also used Eclipse [1] and the Android Development Toolkit (ADT) [2] for this purpose.

The GUI can be designed in a fully graphic environment in Eclipse (see **Figure 5**). Buttons, entry fields, text fields and so on can be dragged from the toolbar to the virtual tablet screen. This generates a file named `main.xml` containing descriptions of all of the elements, including their unique IDs.

The main routine is located in the `TimeLapseActivity.java` file (created by Eclipse) under the class `TimeLapseActivity`. This class pro-

Figure 3. The GUI for camera operation using the tablet.

Figure 4. The accompanying web interface on the computer monitor.

vides the interaction with the buttons. To access a button, you have to link an object to the previously mentioned ID as illustrated by the following example:

```
ToggleButton mtoggleButton1 =
(ToggleButton) findViewById(R.
id.toggleButton1);
```

Buttons can be used in two ways: actively or passively. An object's functions can be used to read buttons and set values. This is passive use. If a function of the class should be actively invoked when a button is pressed (or when the status changes),
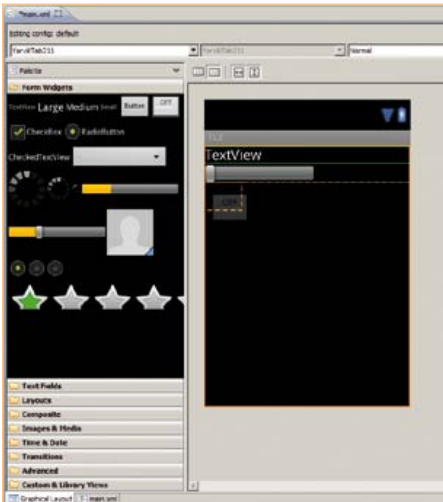
Figure 5. The graphic design in Eclipse.
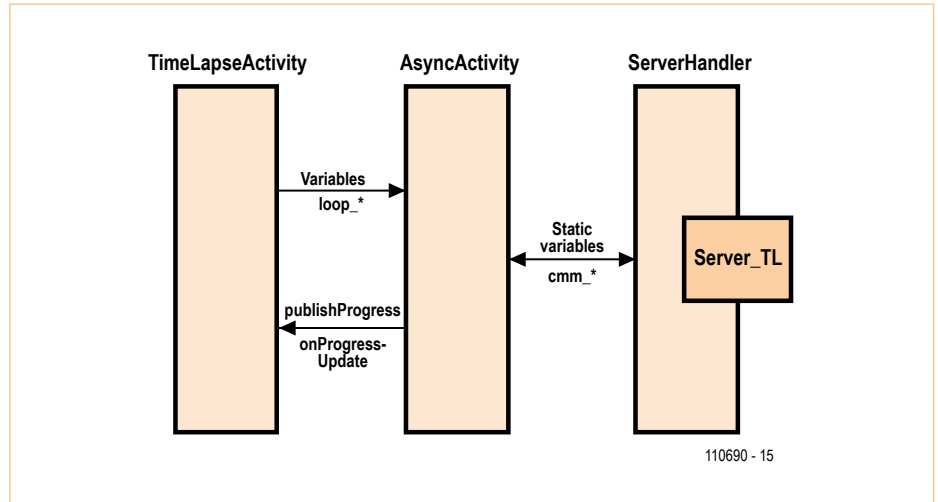
Figure 6. Interclass communication.

this is done with `mtoggleButton1.setOnClickListener(this)`. Only the Start button is used actively in this program. The time-lapse portion of the application is a procedure that runs for a very long time. This is not possible in the main class (`TimeLapseActivity`) because this class is responsible for handling the GUI. If this class is busy, the GUI will not be refreshed, with the result that the GUI will freeze solid. Furthermore, Android terminates a class of this sort if it does not respond within 5 seconds because it assumes that the program has crashed.

This is why Android provides the class `AsyncTask`. It runs in a separate thread that is allowed to be constantly busy. This thread may be launched only once, for which reason it is not launched every time a time-lapse sequence is started but instead runs continuously, even if a time-lapse sequence is not being executed.

Writing directly from `AsyncTask` to the GUI elements is not allowed (it causes a crash). The functions `publishProgress` and `onProgressUpdate` are provided for this purpose. `publishProgress` is used to call `onProgressUpdate` from `AsyncTask`. Writing to the GUI is only allowed from the latter function. As only this one function is available for driving all of the buttons, a specific action is defined in `onProgressUpdate` according to the value of the first passed-in parameter (see the code in `onProgressUpdate` and the `pp_*` variables).

The variables (`loop_state`, `loop_count` and `loop_delay`) are used for communication to start or stop a time-lapse sequence (communication in the opposite direction of `publishProgress`). **Figure 6** illustrates the communication between the various classes.

The pulse waveform for the servo is stored in an array, which is read out by the function `audioTrack`.

Web server

This project also uses the WiFi Internet link. The status and settings can be viewed in a web page, and the application can be operated from this page. There are essentially three options for making all of this possible.

The first option is to use a separate web server app, for which there are many free downloadable versions available. This allows files to be viewed in the file system. This is a simple solution, but it has the drawback that it is limited to viewing the system; it does not support operating the system. It also requires frequent updating of the HTML file, which in time will wear out the flash memory.

At the other end of the spectrum there is the option of using an extensive package such as i-jetty [3]. This allows you to implement a versatile web server, but such an extensive range of functions (and the associated complexity) is not always necessary. The author chose to use the open-source software 'android webserver' [4] for this purpose, by including the code in the application and modifying it where necessary. The most significant modification is that the web server still reads files from the file system, which allows images such as `logo.jpg` (as used here) or CSS files to be used in the directory, with the exception of a file with a user-defined name (in this case `index.html`) that the program uses to generate the web page on the fly. Because the web server runs in a separate thread and therefore does not have direct access to the main routine `TimeLapseActivity`, the settings and status data are transferred using *static* variables (see the `cmm_*` variables). A static variable can be regarded as a fixed memory location. Unlike normal variables, which require a new variable to be created for each object of a class, a static variable is the same for all objects of a class, and it can also be read and written from another class. For this reason, static variables are called by their class name instead of their object name.

The application is checked by analyzing the link. In the same manner as for HTML forms (in GET mode), the variables are placed after a question mark in the URL (URI). In this case the URI `index.html?act=start` starts the TimeLapse process. The same approach can be used to set the number of pictures and the delay, for example with `index.html?del=5`. This is generated automatically by the HTML FORM order. Static variables are also used to communicate these settings and orders. These variables are monitored in `AsyncTask` loop. Another way to do this would be to use a handler for this communication, for which there are two options: a general handler or a handler that is specified when the object is created. However, that was not necessary in this case.

Other functions

Here we can make a few general comments regarding programming for Android.

It is user friendly to save the settings (pulse widths, phase, number of pictures and delay) in the program so that they do not have to be entered again the next time the program is launched. Android has specific functions for this, comparable to the registry of MS Windows. These functions are elements of the `SharedPreferences` class. The parameter values are read when the program starts up and are saved when the program is stopped (by OnStop). These settings are also retained when a new version of the program is installed.

Current versions of Android do not include an application manager, so it is not possible to see which applications are running (regardless of whether they are running in the foreground or the background). This means that it's easy to lose sight of your application. If you then launch the application again, you will end up with two instances of the application running in parallel. This is especially problematical with the application described in this article because the web servers of both instances will listen to the same port, making the results unpredictable. Three provisions have been incorporated in the program in this connection. Firstly, the icon is displayed on the status bar to make it easy to switch back to the application from the status bar. Functions for this purpose are provided in the `NotificationManager` class. Secondly, the setting `android:launchMode = "singleInstance"` is included in the `TimeLapse manifest.xml` file to indicate that a second instance of this program is not allowed to run in parallel. Additionally, the tablet should not be allowed to enter screensaver mode, since that makes the application unstable for some unknown reason. This is prevented by the setting `android:keepScreenOn = "true"`. Incidentally, there are rumors that this problem will be mitigated in Android version 4 and later.

When an application is installed in Android, the user must give permission to grant the application specific privileges for communicating with various hardware and software components of the tablet. For this program this consists of access to the WiFi interface, access to the flash memory card (for the web server files), and access for blocking the screen saver. These permissions must be placed in the `TimeLapsemanifest.xml` file in the form of `uses-permissions` (take care to avoid the common typo 'user-permissions')

Finally, you can dress up the program with your own icons. This requires placing three images in .png graphic format in the project's Res directory under drawable-hdpi, drawable-mdpi and drawable-ldpi with sizes of 72x72, 48x48 and 36x36 pixels respectively.

All files for this project can be downloaded free of charge from www.elektor.com/110690.

(110690)

Internet Links

[1] http://eclipse.org

[2] http://developer.android.com/sdk/eclipse-adt.html

[3] http://code.google.com/p/i-jetty

[4] http://code.google.com/p/android-webserver

The Five Rules...
...when choosing a DSO

By Andreas Grimm (Germany)

The oscilloscope marketplace has not become any clearer over recent years. Their capabilities have been expanded with the addition of many new and innovative features to increase their usefulness. More recently new manufacturers have also appeared. The oscilloscope is the hub of any test and development environment, it will most likely be in daily use for many years to come so it is vital to consider as many factors as possible before you choose a new model.

The majority of Elektor readers will be able to tell you that the most important things to look out for when buying a digital oscilloscope (DSO) is its bandwidth and sample rate. These are indeed two of the most important or key features but there is also a list of other things that need to be considered. An oscilloscope is such an important piece of test gear that it's worth investing some time to make sure you will not be disappointed.

1. The key features
Your choice of bandwidth and maximum sample rate will depend on the fastest signals that you anticipate will need to be observed. Digital signals are more prevalent in circuits these days so the rise time of the input stage is very important.
As a real world example you may be working on a system containing a processor running at 8 MHz. The rise and fall times for the clock will typically be 10 ns. The rise time of the scope's input amplifier must be faster than the input signal otherwise you will just be displaying the characteristics of the scope's input amplifier rather than the observed signal. A practical figure for the rise time is that it should be about 30 % of the signal under observation. In this example observing an edge with a 10 ns rise time indicates that the scope's input amplifier must have a rise time T_r of 3 ns or better. Using the formula

$$B = 0.3 / T_r$$

indicates a 100 MHz oscilloscope is needed. **Figure 1** shows the effect of the input rise time of a 100 MHz DSO on a signal with a 10 ns rise time.

Once the bandwidth has been calculated we can turn our attention to the required sample rate. We can use the formula SR = 8 to 10 x B where B is the scope's analog bandwidth. For a 100 MHz scope this results in a sample rate of 1 GSamples. This ensures that the square wave fundamental and a sufficient number of harmonics can be captured in accordance with signal theory.

Why do we so often get feedback from engineers who have just used these two basic criteria to select a DSO and are unhappy with their scope when they come to use it? The answer is usually because they have overlooked the importance of memory depth.

We may indeed be interested in the step response of a circuit to a single repetitive edge but more often than not in digital circuits we also need to capture a complex sequence of edges or a data stream. This is where the third criterion, the size of the waveform memory, plays an important part. A very simple formula can be

Figure 1. The effect of input rise time of a 100 MHz DSO measuring a signal with a 10 ns rise time (white), the resulting curve is shown in yellow.

used here to give the necessary memory depth, which is equal to the product of the sample rate and the observed time window. The optimal time window length is determined by the types of signals to be observed. For example in a mains powered switched mode power supply we will need a window in the millisecond range to observe the switching control signals but to observe effects over several mains cycles we would require a window of around 100 ms.

Analyzing microprocessors systems will typically require the display of data transfers occurring over a few memory cycles. The observation window will be in the sub-microsecond range, or in the millisecond range to take in several transfers. To sum up, a display window of 1 to 100 ms is a good practical value.

Taking the example given above (8 MHz processor clock, T_r = 10 ns, SR = 1 GSamples/s) and displaying a 1 ms time window gives:

Memory = 1 GSamples/s × 1 ms = 1 MPoint

i.e. one million memory points (see **Figure 2**).
Conversely with a fixed recording time (1 ms) and a given memory depth using the above formula we get the resulting sample rate. This can be seen to decrease dramatically with a smaller memory depth as shown in the table.

Figure 2. Fast signal changes (<10 ns) can be resolved with a one million point acquisition memory and a recording time of 1 ms.

Time window	Memory depth	resulting sample rate
1 ms	2M Points	2 GSamples/s
1 ms	100k Points	0.1 GSamples/s = 100 MSamples/s
1 ms	10k Points	0.01 GSamples/s = 10 MSamples/s
1 ms	2,5k Points	0.004 GSamples/s = 4 MSamples/s

From this we can see that the memory depth is a very important property of a DSO and one which is all too often overlooked.

2. Measuring properties

The usefulness of the scope is largely determined by the properties of the input analog amplifier stages and its triggering properties. The analog signal path should offer high sensitivity and low noise. The best models on the market offer a maximum input sensitivity of 1 mV/DIV but this is by no means the standard value. To make use of this high sensitivity it's important that the input amplifier introduces as little noise as possible to the measured signal: even at its most sensitive setting the noise should be less than one quarter of a scale division. These properties for example allow meaningful measurement of ripple levels (small signals superimposed on much larger signals) to be made. Trigger sensitivity is also important here to enable measurements to be made on the waveform

of interest. The trigger sensitivity should be much better than one scale division.

Particularly during the development of power electronic systems it can be useful to perform mathematical analysis on the channel waveforms. Any high frequency interference can first be eliminated with a low pass filter then the energy value can be calculated by multiplying the voltage channel by the current channel and then integrating the result. The 'chained math function' capability is often an optional feature and is not seen in scopes under 6,000 Euros.

A standard feature of the DSO is the cursor measurement function. It is especially useful if the cursor can follow the waveform whilst displaying in real time the values of time information and voltage level. This is far more convenient than switching back and forth

Figure 3. An FFT feature is only really practical if it uses a sufficient number of points for the calculations (left 2048 Points, right 65536 Points).

between the time and amplitude cursor. In addition it is also useful to be able to configure Automatic parameter measurements allowing say signal pulse width or overshoot to be calculated. The advantage of using parameter instead of cursor measurements is that it only needs to be switched on once and gives reproducible results.

A pass/fail test is useful to continuously monitor the observed signal waveform by means of a configurable mask. The reference

waveform mask plus tolerances is first defined and when a violation occurs the scope can be programmed to stop measurement, output a signal or perform a screen print.

Meanwhile almost all DSOs come with an ability to perform frequency domain analysis in the form of the FFT function. This can be used to identify the source of any in-band interference. In practice however with some budget scopes this feature is poorly implemented using too few points to be of any use. The number of points used in the calculation (together with the time period) determines the FFT resolution. Using just 1,000 points is insufficient and meaningful results can only be achieved using 32,000 or more points (see **Figure 3**).

3. User friendliness
In addition to the hard facts and figures of the scopes specification there are also features which can best be appreciated by using the equipment. The display size and resolution would be in this category. While 6 inch (and above) color TFT screens are the norm today what often is disappointing is the screen resolution. A VGA display (640 x 480 pixels) with almost full horizontal and vertical viewing angles and high contrast should be the minimum requirement. QVGA displays (320 x 240 pixels) are generally disappointing especially on a MSO (Mixed Signal Oscilloscope) where up to 20 channels of information may need to be displayed. A port for connection of an external mon itor or LCD projector may be beneficial.
During the development of complex hardware designs it doesn't take long for the work bench to fill up with test equipment. Equipment which can be stacked or which have a small footprint will therefore be advantageous. Other factors such as high fan noise can be irritating especially as the equipment will typically be running continuously throughout the day.

4. Future proofing
The proliferation of embedded systems seems is relentless and with them comes the need for engineers to display time-synchronous analysis of analog and digital signals. While memory is usually connected to the processor using a parallel bus other peripherals such as FPGAs, sensors or displays are often connected over a serial bus
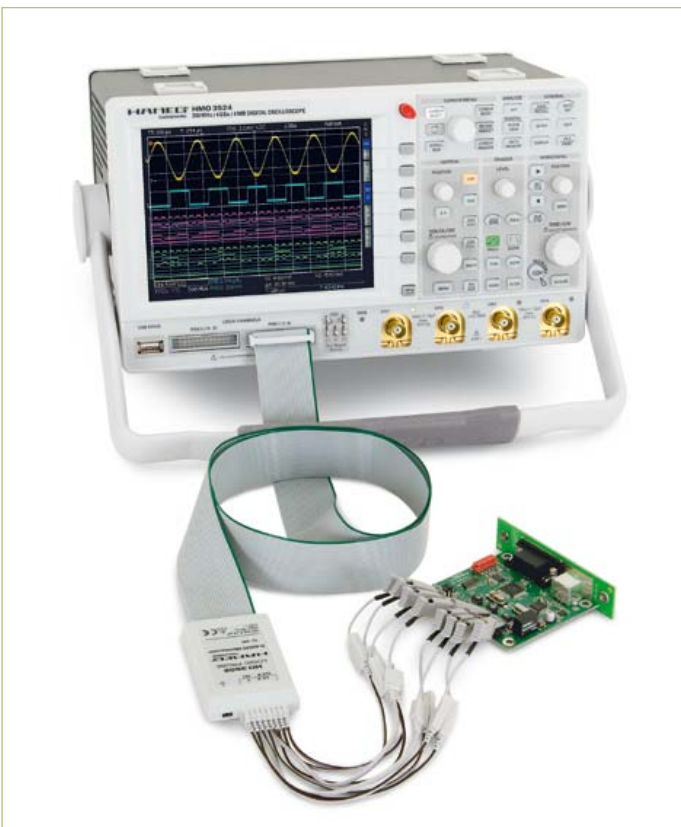
Figure 4. A mixed signal oscilloscope.

such as a UART, I²C or SPI. During development of such designs it is useful if the oscilloscope can display parallel data and also trigger and decode serial data (**Figure 4**).

The oscilloscope will prove far more useful if it offers the flexibility to work in MSO operation or has the capability to decode serial data protocols in common usage or those which may be introduced in the future. With a tight test equipment budget it is worth considering whether the decoding and triggering from serial protocols is necessary on both analog channels and if the external trigger is necessary on the dual channel scope.

Documentation is an important part of project development and it is advantageous to be able to include test results. The DSO should provide at least a USB port for connection of an external PC to transfer data. With the DSO in an automated test environment a GPIB (or more increasingly Ethernet) will be necessary to connect to the test control computer. If not fitted as standard it should at least be available as an optional upgrade. Before purchase it is also worth considering after-sales support, good support will be easy, fast and low-cost to protect your investment for at least five years or more.

5. The price/power trade off

It is clear that your choice of DSO should not just be made on the basis of its most important technical features and cost. There are a number of other factors that also need to be considered. To simplify the process we have collected them together in the form of a checklist:

- Bandwidth (rise time), sensitivity and noise of the input amplifier.
- Sample rate, Memory depth
- Trigger modes and sensitivity
- Display size, resolution and viewing angle, external monitor port
- Functions such as cursor and parameter measuring and 'math' channels
- Pass/Fail waveform test
- Mixed-Signal-Option (or as an optional upgrade)
- Triggering and decoding of serial protocols (or as an optional upgrade)
- Interfaces such as USB, LAN, GPIB (or as optional upgrades)
- Service and support for as long as possible to achieve maximum lifetime from your investment.

The majority of the most important equipment characteristics can be found on the equipment's data sheet or user's manual. Other properties such as the noise level produced by any fan fitted to the equipment or the screen viewing angle can often be answered by calling the appropriate customer services. Best of all is to arrange a hands-on test of the DSO before purchase.

(100896)

Andreas Grimm is head of product management for HAMEG Instruments GmbH (www.hameg.com).

Electronic LED Candle
You can blow it out!

Imitation candles using an LED as the illuminating element are available commercially. But here we're describing a rather different project with a few unusual characteristics — after all, candles are meant to be blown out!

By Antoine Deschamps (France)

Candlelight is by nature variable, so we're going to do a bit of animation to simulate the candle flame. There will be a lighting sequence, a blowing-out sequence, and a few animated sequences intended to reproduce more or less faithfully the flickering of a natural flame.

To produce these animations we need a microcontroller. We went for the PIC16F1827 from Microchip which has a 4 Kword program memory, enough for the code to handle the application and the animation sequence definitions.
This microcontroller also has easy-to-implement functions for handling touch keys. So we decided to include a copper area on the back of our PCB to create a touch pad to let us light the candle.

To achieve our (rather over the top!) objective of being able to blow out the candle, we have found a new use for a temperature sensor in the form of an NTC thermistor. And it works!

Puff detector

As stated above, we're using an NTC thermistor (R2 in **Figure 1**) as an air movement detector. We've chosen a type in a CMS 0603 package with a nominal resistance of 220 Ω @ 25 °C, a beta coefficient of 3,540 K, and a maximum dissipation of 180 mW. A high beta coefficient means the resistance varies to a greater extent with variations in temperature.

The principle of our circuit is as follows. We may consider that a person's breath is appreciably similar to the ambient temperature in the room. So at first sight, blowing on a thermistor that is already at ambient temperature isn't likely to make its resistance change! The trick here is to pass enough current through the thermistor to heat it up slightly. Then a pronounced blow will cause its temperature to drop, which we'll then be able to measure using one of the microcontroller's analog inputs.
Our circuit includes a potential divider with a pad resistor of 100 Ω (R1). At power up

and with an ambient temperature of 25 °C, the current is

$$I_{25} = 5 / (100 + 220) = 16 \text{ mA}.$$

The voltage across the thermistor is $V_{25} = 220 \times I_{25} = 3.5$ V, and the power dissipated by the thermistor is

$$P_{25} = V_{25} \times I_{25} = 3.5 \times 16 \text{ mA} = 56 \text{ mW}.$$

We're a long way from the maximum of 180 mW, but this power is enough to cause the device's temperature to rise, thereby reducing its resistance, increasing the current, and so on. After a few seconds, the whole thing stabilizes, as the oscilloscope trace shows (**Figure 2**). Our software waits for 20 s so that the temperature is properly stabilized before taking the measurements into account. It's tricky to show how this graph changes when you blow on it. After a lot of attempts, we managed to define that a positive variation of the order of 25 mV is enough to detect a human breath.

Technical specifications

- Realistic flame simulation
- Blows out
- Lights via a touch pad
- PIC16F1827 programmed in C, source code provided
- Puff detector based on an NTC thermistor
- Constructional difficulty: average

From a software point of view, it's necessary to suit the frequency of the analog acquisitions to the rate at which the measured signals change. We've used initial filtering to eliminate noise, then averaging over 16 successive values, so as to be able to detect when a filtered value deviates from the average of the 16 values by more than 25 mV. The result is that you have to blow reasonably close to the candle to blow it out. Nonetheless, the whole thing is still quite sensitive to draughts.

You don't need accuracy here, as all you are doing is comparing the temperature differences to the average of the temperatures recorded. So there was no need to fit a voltage reference on the board, nor even to use the microcontroller's internal reference. Hence the analog measurement range is delimited by the 0 V rail on the one hand, and the supply voltage on the other.

Touch pad

The touch pad that enables us to light the candle is connected directly to analog input 0 of the microcontroller IC1 (**Figure 1**). The PIC16F1827 data describes the capacitive sensing module clearly, and programming it should not present any problem. Except for just one detail, as we don't immediately have any idea of the frequency of this oscillator that is formed with the touch pad. Remember that it's the variation of the capacitance of this copper pad when a human finger approaches it that causes a frequency variation, which is what we are going to need to measure or, more accurately, compare against a reference. The oscillator output is connected to Timer 0, which counts freely.

So as to be able to observe these frequencies, we've duplicated the MSB of the Timer 0 counter register TMR0 to the output connected to the test point. This then gives us on the oscilloscope test point a frequency of the order of 130 Hz, which drops to below 90 Hz when a finger touches the contact pad. This frequency corresponds to a frequency on the eighth bit of the register 256 times higher at the input to Timer 0, i.e. around 32 kHz.

As our application is clocked at 1 kHz (see below), all we have to do is examine the contents of the TMR0 register at each 1 ms
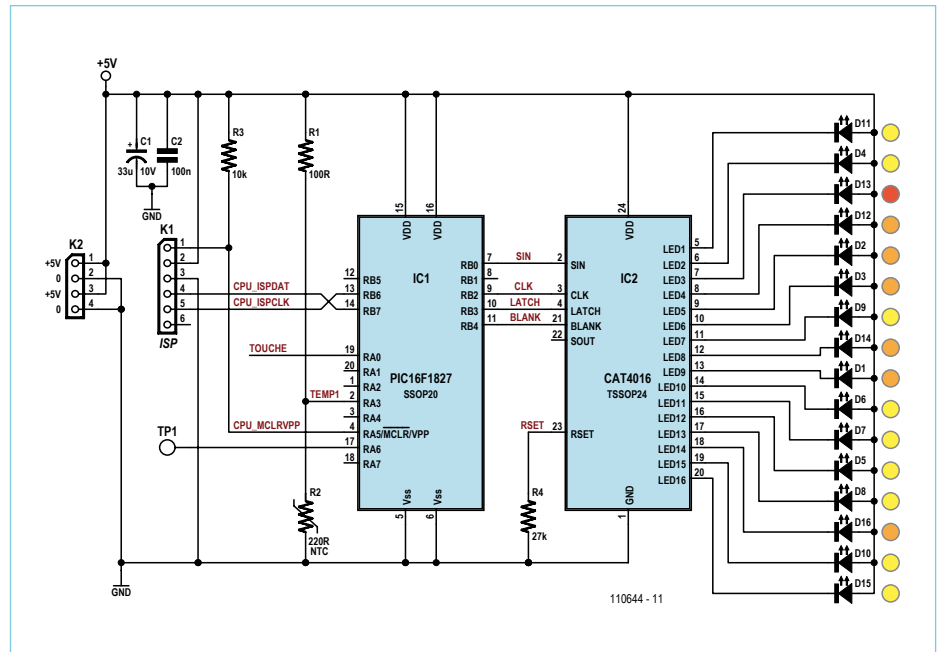
Figure 1. Circuit diagram of the LED candle. The component count has been minimized, so as to fit within the area occupied by the LEDs on the top face of our PCB. The circuit is powered from 5 V, with an electrolytic capacitor (C1) to clean and buffer the voltage. The application of the microcontroller IC1 is reduced to its simplest expression; we shall be using the internal oscillator, since we don't need great accuracy.

cycle. TMR0 has a value of around 32 when the touch pad is free, dropping to around 23 when your finger is touching the pad. Once it has been read, the TMR0 register is forced to 0, so as to start counting again. In our application, we've set the threshold at 26, which is quite low, and so you really do have to put your finger on the pad for it to be triggered. Increasing this value would make the touch function more sensitive.

One remark in passing: when developing a project, it's important to make provision whenever possible for one or more test points. The oscilloscope too is a helpful programming tool!

The flame

This LED application is not intended for lighting, just to be decorative; so we're going to use low-current LEDs, here with a

maximum of about 2 mA. This power range lets us choose SMD LEDs, which will make it possible to fit them onto a relatively small area of PCB.
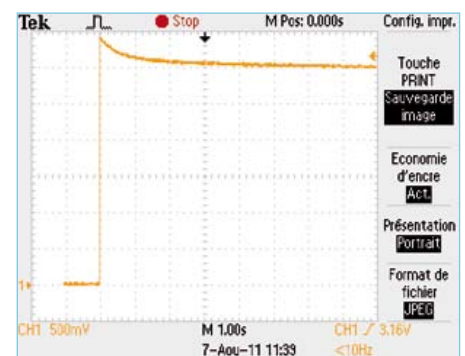
Figure 2. The temperature of the air movement detector stabilizes after a few seconds.

COMPONENT LIST

Resistors (SMD 0603)
R1 = 100Ω
R2 = NTC, 220 Ω, Epcos
    B57311V2221J60 (e.g. Farnell # 129-
    9912 or RS Components # 706-2702)
R3 = 10kΩ
R4 = 27kΩ

Capacitors
C1 = 33μF 10V aluminum electrolytic,
    case C, e.g. Panasonic EEE1AA330SR
C2 = 100nF, SMD 0603

Semiconductors
IC1 = PIC16F1827–I/SS (SSOP20),
    Microchip
IC2 = CAT4016Y-T2 (TSSOP24), ON
    Semiconductor
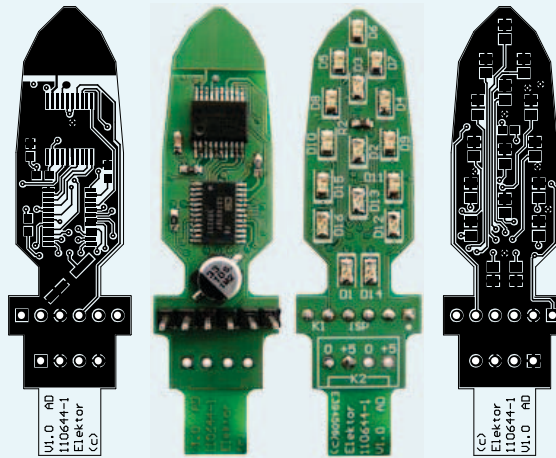
D4 à D11, D15 = LED, yellow, low
current, SMD 0805, e.g. Kingbright
KP-2012SYCK
D1, D2, D3, D12, D14, D16 = LED, or-
ange, low current, SMD 0805, e.g. King-
bright KPHCM-2012SECK
D13 = LED, red, low current, SMD 0805,
e.g. Kingbright KPHCM-2012SURCK

Miscellaneous
K1 = 6-pin pinheader, lead pitch 0.1 inch
(2.54 mm)
K2 = 4-pin pinheader, lead pitch 0.1 inch
(2.54 mm)

Figure 3. The double-sided PCB carries only surface mount (SMD) devices, but it can still be built by hand.

Driving the total of 16 LEDs that are going to make up the flame has been entrusted to a specialized IC, the CAT4016 from ON Semiconductor (IC2). This device incorporates constant current regulation for 16 LEDs, with the current value defined by an external resistor (R4). The current passing in each LED channel is 50 times the current passing through the resistor connected to the R_{set} pin. Given that the voltage on this pin is regulated to 1.2 V, with R4 = 27 kΩ we have a current of 44 μA through the resistor, and hence 2.2 mA in the LEDs. This value offers good visibility of the animations while ensuring acceptable luminosity levels overall. If you use other LEDs you may have to tweak the value of R4.

The CAT4016 is driven via a serial connection. Four signals are used, in the direction from the CPU to the CAT4016:

• CLK: clock signal
• SIN: serial in data signal
• LATCH: store signal
• BLANK: signal to blank all 16 LEDs simultaneously

The device data sheet illustrates the sequencing of the signals for individually driving the 16 LEDs. The data received, clocked by the clock signal, are stored in a shift register whose output is accessible (serial output signal SOUT), which we are not using here, but which would make it possible to connect several devices in series without using up more in/outputs on the microcontroller.

The CAT4016 uses on/off digital drive. To make the animations a bit more fluid, we need to be able to adjust the brightness of the LEDs on an individual basis. The solution is to do this in the software.

Our microcontroller is clocked at 16 MHz, and as it takes four clock cycles to execute an instruction, we have an actual speed of 4 MHz. Timer 1 has been set to produce an interrupt every millisecond. This source of main timing is often called a "tick system". At each tick, we refresh the contents of IC2's shift register. On our prototype, the action of transferring the commands for the 16 LEDs takes 160 μs, i.e. 16 % of the processor time.

At this repetition rate, the human eye, or rather the visual perception system, does not perceive the LEDs as flashing on and off. Very far from it, in fact — it would be possible to reduce the refresh frequencies if necessary, for example, to leave more CPU time for the other software routines. So we can create software pulse width modulation (PWM) for each LED. In our case, we've contented ourselves with a 5-state PWM (0 %, 25 %, 50 %, 75 %, and 100 %). This might not seem very much, but the software also has to define the patterns the LEDs are going to make in an attempt to reproduce the flickering of a real flame. Now we're restricted for space in our microcontroller, so unless we give it external memory and a method for programming this storage space, we need to reduce the number of states so as to be able to define enough patterns to make our animation pleasing.

PCB

A 17 × 60 mm board was designed (**Figure 3**), of which the LEDs occupy an area 17 × 39 mm. The lower part of the board is reserved for the microcontroller programming connector and for the power input points.

On the top of the board there are only the LED's in CMS 0805 packages, i.e. a size of 2 × 1.25 mm. We also find on this side the temperature sensor, the NTC thermistor in an 0603 package. This is necessary so we can blow on it!

On the underside, you can spot at once the "tip" of the board, which forms the touch pad. Note too the test point TP1, above the microcontroller (referred to above).

To fit all the other functions on the other side, it was necessary to resort to fine-pitch packages. Our microcontroller comes in an SSOP20 package with a pitch of 0.65 mm and the LED driver was chosen in a TSSOP24 package, also with a 0.65 mm pitch. These fine pitches should not be seen as an obstacle, but rather as an effort to get closer to the industrial techniques that are within the reach of an amateur. An illuminated magnifier, a fine-tipped, temperature-controlled soldering iron, 0.5 mm (AWG24) diameter solder, and above all flux in a syringe! Better not to drink too much strong coffee, though, before you start the soldering…

The technique is well established — you should put a tiny amount of solder on a land

Extract of the code for initializing of Timer 1 for the 1 kHz time base:

```
T1CON = 0b01100101 ; // Timer 1 configuration (Tick Interrupt Source)
// 01     : Timer1 source is Fosc = 16 MHz
//   10   : Prescaler = 4 -> Ftimer = 4 MHz
//     0  : Timer1 oscillator circuit disabled
//      1 : Do not synchronize external clock input
//       0: unimplemented
//        1: TMR1ON=1 -> Timer1 enabled


T1GCON = 0b00000000 ; // Timer 1 gate control
//  0     : TMR1GE=0 -> Timer 1 counts regardless of Gate functions


TMR1H = 0xF0 ; // 0xF05F is the complement of 0x0FA0=4000
TMR1L = 0x5F ; // Counter startup value for 1ms interrupt period (1000 Hz)
PIE1 = 0b00000001 ; // Interrupt sources
//        1: TMR1IE=1 -> Timer1 IT source enabled
```

Extract of the code for initializing the function to handle the capacitive sensor:

```
OPTION_REG = 0b01101000;
// 0       : WPUEN#=0 -> Weak pull-ups enabled by individual WPUx
//  1      : Interrupt on rising edge of INT (not used)
//   1     : TMR0CS=1 -> Timer 0 counts on T0CKI
//    0    : TMR0SE=0: Timer 0 counts of rising edge of T0CKI
//     1   : PSA=1: prescaler is not assigned to Timer0
//      000: Prescaler value, 1:2


// Capacitive sensing configuration
CPSCON0 = 0b10000101 ;
// 1       : Capacitive module is enabled
//  000    : unimplemented
//     01  : oscillator in low-range
//       0 : current direction status (read-only)
//        1: TOXCS=1 -> Timer0 input is Capacitive Oscillator output


CPSCON1 = 0b00000000 ; // Source is CPS0/RA0
```

at one end; then position the device onto its lands and solder the tinned IC pin. Once you are sure it is correctly positioned, you add some flux and solder the other pins. Solder bridges will form, but it's easy enough to remove them using fine desolder braid. The board has been made double-sided and uses through-hole plating. The power rails have been routed in the form of planes; the +5 V plane is on the top surface, since this rail is the common for our LEDs. Hence the 0 V rail is on the underside.

The *in situ* programming connector K1 is in fact a row of six through-plated holes. The pin-out follows the order of the signals given by the ICD3 programming interface from Microchip (**Table 1**).
There are pads for connecting the power (K2) — two for the +5 V rail (pins 1 and 3), and two for the 0 V rail (pins 2 and 4). Here

Table 1. Pin-out for the ICSP connector. Pin 1 is marked with a square dot.	
Contact	**Signal**
1	MCLR/Vpp
2	+5 V rail
3	0 V rail
4	ISPDAT
5	ISPCLK
6	Not used (PGM signal)

too, pad 1 is marked by a square dot. Doubling up the supply pads like this is useful when joining several boards together, to multiply the graphical effect.

Software
As a guide, the current code written in C occupies 84.5 % of the microcontroller's Flash memory, i.e. 3,460 words out of the 4,096 available. A good three-quarters of this space is used for defining the patterns and would require a degree of optimization if the space issue had to be tackled.

The full source code, along with the HEX file, is available free on the project web page. You'll also find there the PCB design and you can order the PCB (bare, or built and tested).

To your soldering irons and... Happy Holidays!

(110644)

Software, PCB, products, etc.

www.elektor.com/110644

USB Data Logger
Store serial data conveniently, safely

By Thomas Fischl (Germany)

This neat design allows you to use a standard USB memory stick to store data in a microcontroller based system. Memory sticks provide a large volume of low cost, portable, non-volatile memory space. For this application the memory stick simply plugs into the data logger system which acts as a host USB controller and has the ability to log or store all serial data sent to it. Both of these tasks are handled with ease by a PIC24FJ64GB002 microcontroller from Microchip.
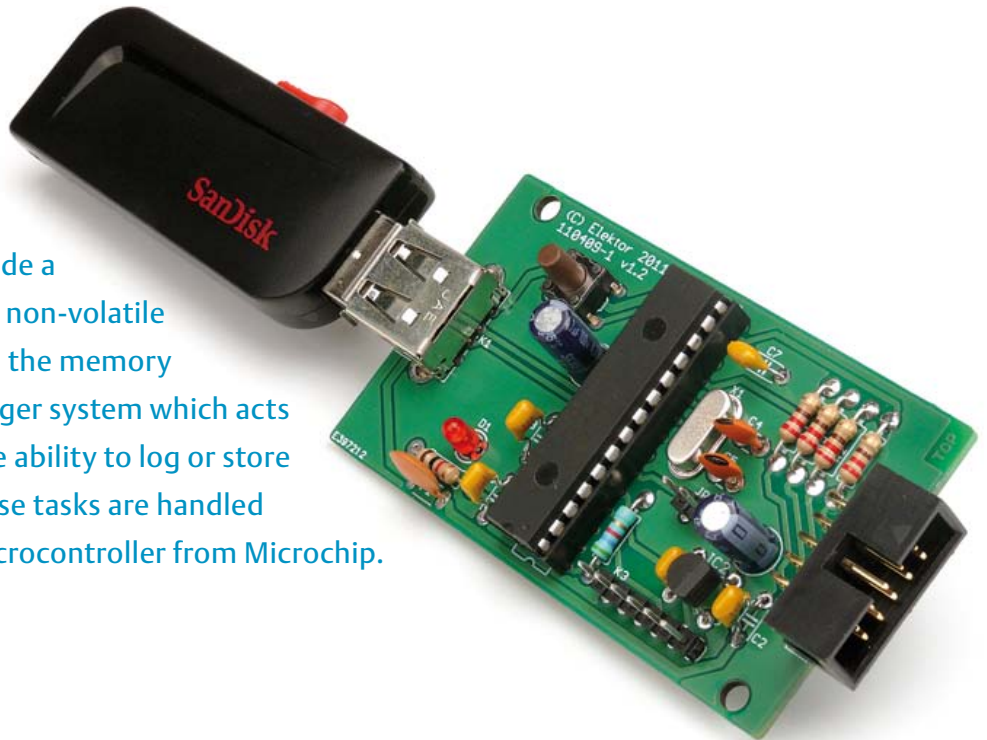
Features

- USB 2.0 compatible
- Supports data rate up to 12 Mbit/s
- Connection to the host:
  +5 V, GND, µC-Tx
- A file stored on the memory stick configures the UART
- Open-Source firmware
- Power consumption:
  +5 V, 50 to 80 mA

These days even the most basic microcontroller will have a built-in serial data interface (UART). This would typically be used to transfer configuration parameters from a control computer or to send information about the microcontroller's operational status. Quite often a microcontroller system may be required to collect data over a long period of time. In this situation it is usually impractical to permanently hook up a PC to continually store the measurement/sensor data; the environment may not be suitable and the relatively high power requirements make this a wasteful solution.

The firmware of the existing microcontroller system could be adapted to perform the data logging function but may require additional memory (both EPROM and RAM) to handle the task.
The USB data logger described here is a low-energy, universal solution to the problem. It takes all the serial data sent from an external microcontroller system and stores it in a file on a USB memory stick which can be analyzed later with a PC.

Hardware
The circuit (**Figure 1**) consists of little more than a PIC24FJ64GB002 microcontroller from Microchip. This particular model implements USB 2.0 OTG functionality. OTG indicates 'On The Go' referring to the amended USB 2.0 specification which allows a USB device to assume not only its traditional role as slave but also to act as a master (host) on the bus. More recent amendments to the protocol enable communication between two OTG devices. For this application we just need the USB host functionality and a standard USB A-type connector to plug in the USB memory stick.

The microcontroller core runs at 3.3 V which is provided on board by voltage regulator IC2. The serial interface is tolerant of input levels up to 5.5 V and is protected by 220 Ω resistors connected in series with the inputs. The 5 V input supply voltage is also used as the bus voltage (VBUS) to power the USB stick. A resettable fuse (F1) provides

Elektor Products & Services

- PCB, order code 110409-1
- Programmed PIC microcontroller: # 110409-41
- PCB artwork file (.pdf): file # 110409-1, free download at [1]
- Firmware: file # 110409-11.zip, free download at [1]

some protection if an external USB device draws too much current.

LED D1 and push button S1 connect directly to microcontroller pins. The LED flashes to indicate data transfer and the push button is used to terminate the data storage process. The 6-way pin header K3 allows the microcontroller to be programmed in-circuit and is compatible with standard programming adapter tools such as those used in PICkit 3 and ICD2/ICD3 from Microchip. Jumper JP1 is not used in the current version of the firmware but may be employed in the future when newer versions of the firmware become available.

The circuit can be built on the PCB (**Figure 2**) which makes assembly very straightforward. All components are fitted to the PCB component side and no SMD outlines are used. An IC socket can be used to fit the microcontroller to the PCB. The fully populated Elektor prototype PCB can be seen in **Figure 3**.

Software

The chip manufacturer Microchip already has a useful number of functions available in their 'Microchip Application Libraries' to support the interface of USB devices. Included here is support for the device class 'Mass Storage Device' which includes USB memory sticks. The FAT file format used by the vast majority of memory sticks is also supported. The firmware was designed using MPLAB, the Microchip development environment with help from the C30 compiler. All necessary programs and software libraries are available free of charge and the firmware source files can be downloaded at no cost from the Elektor project web site [1]. The firmware can be programmed into the data logger microcontroller using a PICkit 3 for example. A pre-programmer microcontroller can also be purchased from the project website [1] which should help speed up construction even more.

Data flow between the UART and the file system is performed by implementing two software ping-pong buffers. The characters received via the serial interface are stored sequentially in one of two buffers. When the receive buffer is full the function of the two buffers is flipped or 'ping-
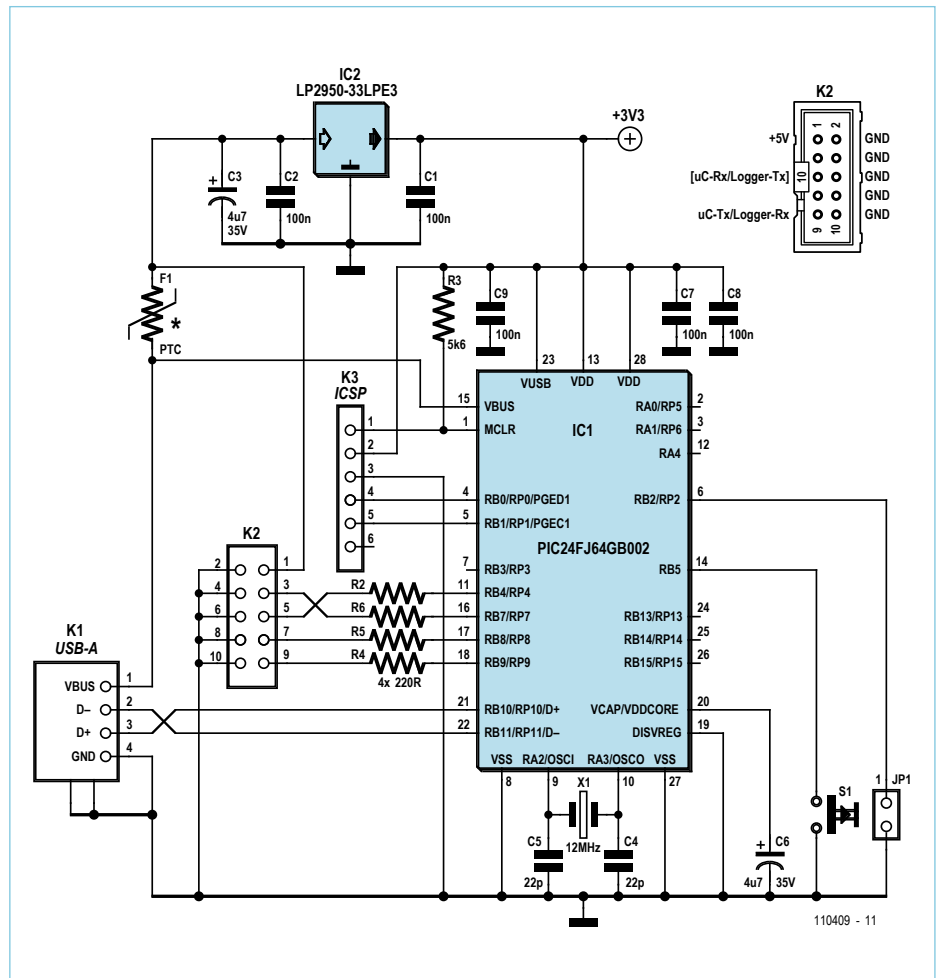
Figure 1. The USB serial data logger circuit basically consists of a microcontroller functioning as a USB host.

ponged' so that the entire contents of the full buffer are streamed to the memory stick while the other (empty) buffer is now used to store received characters. When the receiving buffer fills again the process repeats. Streaming 'chunks' of data to the memory stick in this way improves software efficiency.

System hook up

Communication between the USB data logger and external microcontroller system occurs over the serial interface (UART). The communication signal level from the external system must lie in the range of 3.0 V to 5.5 V; in cases where the external system uses standard RS232 port signal levels it will be necessary to use a RS232/TTL signal converter chip between the RS232 signals and the data logger.

The data logger requires a supply of +5 V which in most cases can be taken from the external microcontroller system. The USB memory stick current consumption varies depending on the manufacturer but usually lies in the range of 50 to 80 mA.

Connections between the external microcontroller system and the USB data logger are all made using pin header K2. The data logger requires just three connections on K2 to the microcontroller system: +5 V (Pin 1), µC Tx/Logger Rx (Pin 9) and ground (Pin

COMPONENT LIST

Resistors
R1 = 1kΩ
R2,R4,R5,R6 = 220Ω
R3 = 5.6kΩ

Capacitors
C1,C2,C7,C8,C9 = 100nF
C3,C6 = 4.7µF 35V radial
C4,C5 = 22pF

Semiconductors
D1 = LED, low-current, 3 mm
IC1 = PIC24FJ64GB002-I/SP (programmed, Elektor # 110409-41)

IC2 = LP2950-33LPE3

Miscellaneous
F1 = resettable fuse, 250 mA hold current, 500 mA trip current (Littlefuse 72R025XPR)
X1 = 12MHz quartz crystal
K1 = USB socket, Type A, PCB mount
K2 = 10-pin (2x5) pinheader, right angled
K3 = 6-pin pinheader
PCB # 110409-1
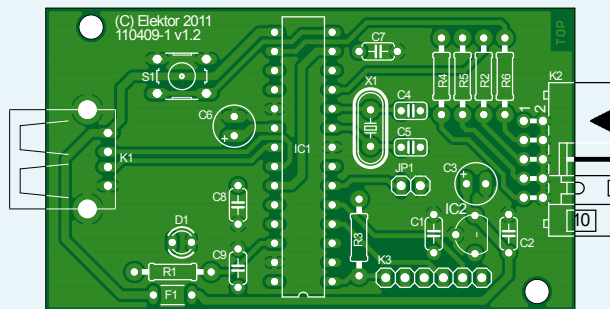
Figure 2. The PCB does not use any SMDs.

10). Pins 2, 4, 6 and 8 are all tied to ground. The second signal connection shown as (µC-Rx/Logger-Tx) is brought out to pin 5 but not used by the current firmware version. Microcontroller port pins 16 and 17 are connected to pins 3 and 7 of K2 via 220 Ω resistors. These additional connections are intended for future use with modified firmware to allow, for example the microcontroller to retrieve stored data.
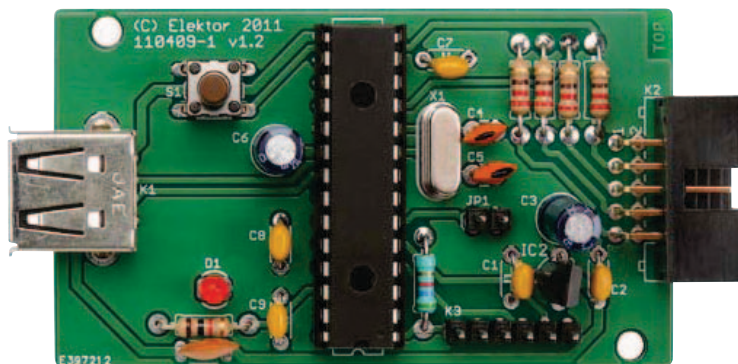
Figure 3. The fully-stuffed Elektor prototype.

Configuration

The serial communication parameters are placed in the simple text file 'config.txt' stored on the memory stick. As soon as the USB stick is recognized (during power-up with the stick already in place or during operation when the stick is plugged in) this file is read and the serial interface configured accordingly. Without this configuration file the default standard communication parameters are: 9600 Baud, 1 start bit, 1 stop bit, no parity.

Operation

The data logger will be in recording mode when it is powered up and a memory stick is fitted. Short flashes from the LED indicate that data is being received over the serial interface. Received data is stored 1:1 in the 'logging.txt' file. Before unplugging the memory stick it is necessary to press the store pushbutton; this ensures that all data held in the receive buffer is stored to the logging file and that the file is closed cleanly. The stored data can then be read on any PC by plugging the memory stick in to a free USB socket and reading the contents of the 'logging.txt' file using a simple text editor program.

Work in progress

Although the data logger design presented here only stores data to a memory stick it is clear that the design can easily be tweaked to make it even more useful. An obvious candidate would be to allow the stored data files to be read back to the external microcontroller system. As a stand-alone logger it could also be programmed to automatically sample the analog and digital inputs at pre-programmed intervals and store the readings to memory stick. Four pins of the 10-pin K2 pinheader are not currently in use but are wired back to the controller I/O pins, these peripheral signals can, for example be used to implement an SPI interface or additional UART.

(110409)

Internet Link

[1] www.elektor.com/110409

Here comes the Bus! (10)
Readings with 22 bits of precision

By Jens Nickel (Elektor Germany Editorial)

This month we up the resolution: an external ADC attached to a node delivers samples with 22 bits of precision. Getting the samples into the microcontroller is easy using its SPI port. Also, we show how to display the results on a PC with a judiciously-modified version of the HTML page from the previous installment in this series.
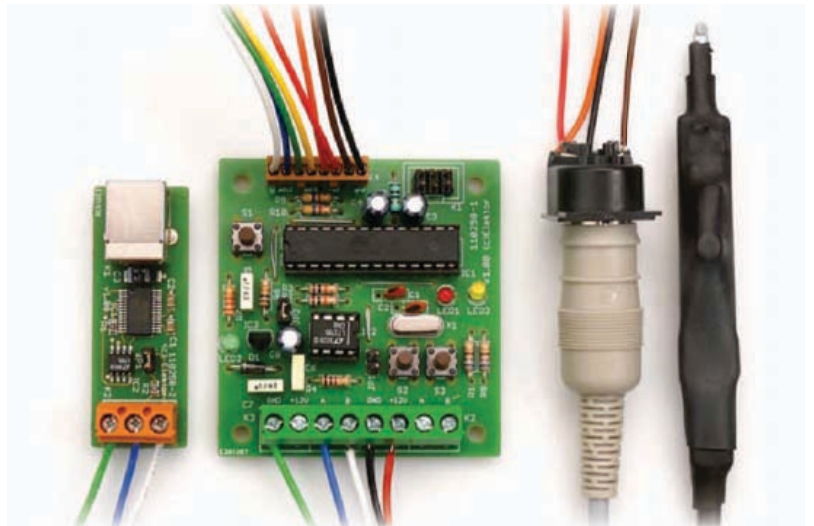
One of the pleasing things about electronics is the way you can get inspiration from other projects. In this case it was the reader project 'Temperature Gradient Meter' from last month's issue [1]. Author Dr Dietmar Schröder selected an external type MCP3551 ADC from Microchip for use in his circuit in order to obtain the high precision required to detect minute temperature changes. This device converts voltages to digital values with 22 bits of precision. **Figure 1** shows the tiny circuit built around
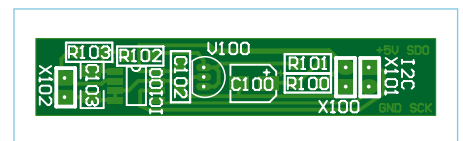
Figure 2. Printed circuit board designed by Dietmar Schröder. X100 and X101 are connected to K4 on the experimental node.
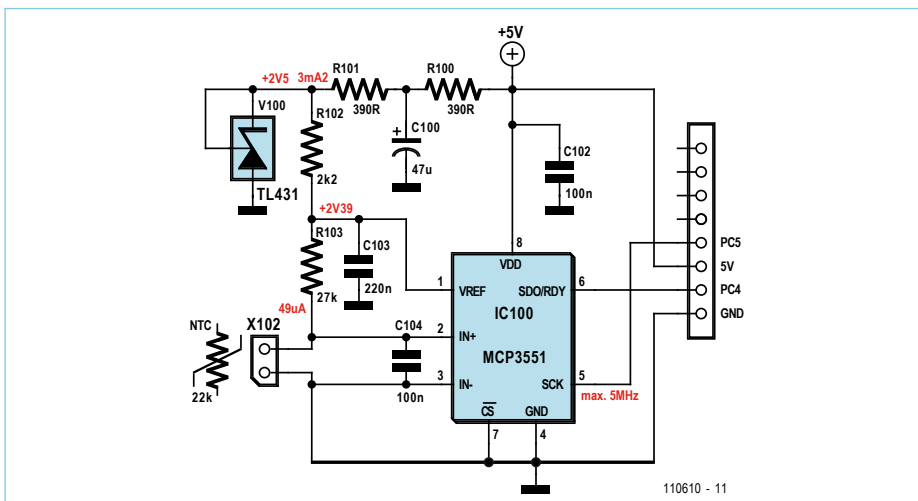
Figure 1. Circuit diagram of the high-resolution temperature sensor, with connection to our experimental node.
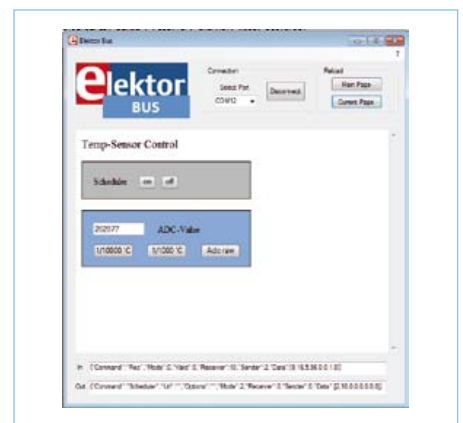
Figure 3. Display of ADC values and temperature readings in the ElektorBus browser.

Elektor Products & Services

- Experimental nodes: printed circuit board 110258-1 or set of three boards 110258-1C3
- USB-to-RS485 converter (ready built and tested): 110258-91
- Free software download (microcontroller firmware plus PC software)
- Products and downloads available via www.elektor.com/110610
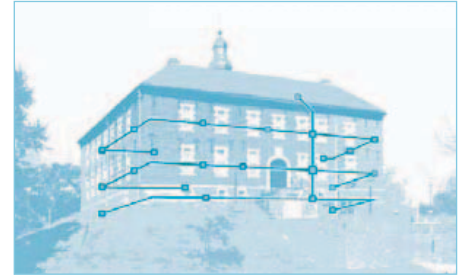
this delta-sigma converter, an NTC thermistor and a couple of support components. **Figure 2** shows the corresponding printed circuit board designed by the author, which can be downloaded at [1]. The ADC makes a new conversion result available at its output (an SPI port) approximately fourteen times a second. The circuit is tailor-made for connection to one of our experimental bus nodes. Two pins of our eight-way connector [2] serve for the digital interface, and two more provide power to the sensor circuit. And because the temperature gradient meter also uses an ATmega microcontroller to process the results, we also have a solid foundation on which to build our code (again, see [1] for download).

Bit banging

The SPI port is driven using 'bit banging', that is, by controlling the interface signals directly rather than using the microcontroller's built-in interface. I swiftly converted the C functions responsible for this into BASCOM (see **Listing 1**). Function `Readexternaladc()` waits for the ADC to pull its SDO pin low. The microcontroller then takes the SCL signal alternately high and low: after each rising edge one bit of the conversion result is made available by the ADC on its SDO pin, with the most-significant bits coming first. The long variable `Dat` stores the result as it is built up: the variable is shifted left and if SDO is high a '1' bit is added in. The datasheet [3] explains that 24 bits must be read in and that an extra clock pulse should be emitted for safety. Note that in this circuit we operate the ADC in 'continuous conversion mode'. All that remains to be done is apply a voltage to the ADC's input and look at the results.

As the original article noted, because of inevitable noise in the circuit only about 19 bits of each conversion result are significant, which corresponds to a still rather impressive temperature resolution of around three ten-thousandths of a degree. There is therefore no reason not to drop the bottom two bits of each result using a right-shift operation. Twenty bits remain, of which the most significant will always be zero for temperatures above —35°C. That fits perfectly with our 'Application Protocol' which allows nineteen-bit integers (plus sign bit) to be transmitted [4]. The BASCOM code, available for download at [5], shows clearly how the nineteen bits are divided among the three data bytes.

Doing the initial tests was easy enough, as using the author's board design saved a lot of time. All I had to do was add a user interface on the PC. I implemented this using the ideas presented in the previous installment in this series, based on HTML pages and the dedicated 'ElektorBus browser'. The HTML page 'Index.htm' from the last installment was modified to receive a 'VALUE4' instead of a 'VALUE2' and display the received value in the text box.

User interface

With the addition of the sensor board we have turned a bus node into a high-resolution sensor device: even just bringing your hand near the unit produces a marked change in the temperature readings. Naturally, we would like to display the readings in Celsius, and so I have implemented the required conversion routine in the sensor node firmware, using the NTC characteristic curve values (in

steps of five degrees) from the temperature gradient meter code with suitable modifications. This demonstration firmware completely avoids the use of floating-point and of division operations. The result can be expressed in units of one ten-thousandth or one thousandth of a degree and transmitted as a four-byte integer. Note that the absolute accuracy of the temperature readings is not especially high unless the sensor is suitably calibrated.

The HTML user interface (**Figure 3**) allows switching between raw ADC values and readings expressed in thousandths or ten-thousandths of a degree. This gives a good demonstration of how a sensor's physical quantity, units and scaling settings can be modi-

---

Listing 1: BASCOM code to read a value from the ADC

```
Function Readexternaladc() As Long

    Dat = 0
    Sck = 1
    Notimeout = 100

    While Sdo = 1 And Notimeout > 0
        Notimeout = Notimeout - 1
        Waitms 1
    Wend

    For Ia = 0 To 23
        Sck = 0
        Waitus 80

        Shift Dat , Left , 1

        Sck = 1
        If Sdo = 1 Then
            Dat = Dat + 1
        End If
        Waitus 80

    Next

    Sck = 0
    Waitus 80
    Sck = 1
    Waitus 80

    If Notimeout = 0 Then
        Readexternaladc = 0
    Else
        Readexternaladc = Dat
    End If

End Function
```

Listing 2: Script within the HTML page

```
var SetFlag = false;
var QuantityToSet = 0;
var ScaleToSet = 0;
var DisplayScale = 0;


function ProcessPart(part)
{

  if ((part.Sender == 2) && (part.Parttype == PARTTYPE_VALUE4))
  {
    if (part.Channel == 0) {TextboxSetvalueScaled('ADC', part.Numvalue, DisplayScale);};
  }

  if (SetFlag==true)
  {
    if ((part.Sender == 2) && (part.Parttype == PARTTYPE_SCALE) && (part.Channel == 0))
    {
      if (QuantityToSet==TEMPERATURE) {TextSetvalue('unit','°C');};
      if (QuantityToSet==RAWVALUE) {TextSetvalue('unit','ADC-Value');};
      DisplayScale = ScaleToSet;
      SetFlag = false;
    }

    else
{
      var parts = InitParts();
      parts = SetScale(parts, 10, 2, 0, 0, QuantityToSet, 0, ScaleToSet);
      SendParts(parts, true);
    }
}


function SetSensorScaleIndirect(quantity, scale)
{
  SetFlag = true;
  QuantityToSet = quantity;
  ScaleToSet = scale;
}
```

fied. The command bytes to set a sensor to report values in ten-thousandths of a degree using channel 0 are (in decimal) 40, 193, 33 and –4.

The JSBus Javascript library lets us avoid the need to calculate these bytes by hand. The commands shown below can be included within the HTML page, and they will generate the appropriate 'part' and send it.

```
var parts = InitParts();
parts = SetScale(parts, 10, 2,
0, 0, TEMPERATURE, 0, -4);
SendParts(parts, true);
```

Reliable transmission

I arranged for the above lines of code to be called when the corresponding button on the HTML form is clicked on, exactly as in the previous installment of this series [6].

However, I discovered in testing that clicks did not always have the desired effect. Now the 'OutCommand' text box in the Elektor-Bus browser showed that the message data bytes were being generated correctly, and so it was probably a problem with the node firmware. My hunch was that the message was not being correctly received by the sensor in the case where it was busy dealing with the external ADC. I noticed that

the transfer of commands was reliable if the variable `intFreeBusTime` in the ElektorBus browser was reduced: strictly speaking in this application we do not need any 'FreeBusPhase' at all, although we might implement the ability to control this from within Javascript code at a later date. Unfortunately, in the absence of a bus monitor or firmware debugging facilities I was unable to track down the source of this bug. However, it does give us an opportunity to see how this kind of problem can be worked around.

Listing 2 shows a solution. The buttons for changing the unit setting call a Javascript routine called `SetSensorScaleIndirect()`, which simply sets a flag rather than directly sending the required command. The master repeatedly sends this command as long as the flag is set. The code to implement this can be integrated into the routine `ProcessPart(part)`, which is periodically called by the JSBus library whenever a reading arrives.

The flag is cleared when an acknowledgement is received from the node. (The BASCOM firmware packs the four bytes required along with the reading in the regularly-transmitted message.) Only then is the display of the selected unit updated on the HTML page. To display the decimal point properly in values shown in the text box we call the function `TextboxSetvalueScaled()`, which is implemented in the new version of JSBus [5]. The downloadable HTML file also now includes a little CSS [7] to make the user interface somewhat prettier.

A touch of 'optimization'
Further testing showed that after a few minutes the display would lock up. Firing up the oscilloscope it transpired that bits were still zipping about on the bus, at least until I stopped the scheduler. Next I arranged to display all the bus bytes in a separate text box, which showed that the only messages being sent were 'FreeBusMessages' from the scheduler. It was not easy to track down the reason for this, as the scheduler is based on three processes running in parallel. The first is the scheduler loop itself, which inter-

rogates the nodes one after another. The second is the routine `ShowMessage`, which is executed in parallel when a sixteen-byte message arrives: if this message comes from the interrogated node then the next node in sequence can be processed. And the third adds a timer to the mix: if a bus participant is not heard from for a certain period, the next node is processed anyway.

I finally realized that I had made a schoolboy error. To indicate which node comes next in sequence I had allowed all routines direct access to the global variable `intPolledNodesCursor`, which points to the next node in the scheduling list. If, however, this variable is changed externally while the scheduler loop is at work, the above subtle problem can be triggered. I modified the code so that instead a flag `boolNextNode` was used to indicate when the scheduler should move on to the next node. The flag is inspected only at the start of the scheduler loop and otherwise does not affect the flow of the code. Success! The system was running reliably, and fortunately there was time to upload the corrected files to our website to accompany the previous installment of this series. The improved ElektorBus browser can also be downloaded at [5], in the form of VB.NET source code and as a .EXE file.

Some more ideas
I would have liked to implement further features, such as a display of temperature gradient including a filter with adjustable parameters. That would have made a complete replacement for the processor board, display and potentiometers of Dietmar Schröder's original project. However, time caught up with me, and time will be even more limited in future after a slight problem with my USB-to-RS485 converter (see elsewhere in this edition). Interested readers should be able to implement various extensions to this project without too much difficulty, given a little expertise in C programming, the source code of the temperature gradient meter, and the Bus tools we have previously described.

Incidentally, the ElektorBus is also capable of receiving messages from nodes. In principle it would only be possible to have two

Setting the interval between readings

In principle a single sensor could be remotely controlled over the ElektorBus without using the scheduler. We simply tell the device how often to report a new reading, just as we would in a data logger application.

BIT								BYTE
7	6	5	4	3	2	1	0	
0	0	SET 1	0	1	C2	C1 CHANNEL	C0	ADDRESS
1	1	1	0	0	0	0	0	COMMAND
0	INTERVAL VALUE							FIRST
0	INTERVAL SCALE							SECOND

110610 - 13
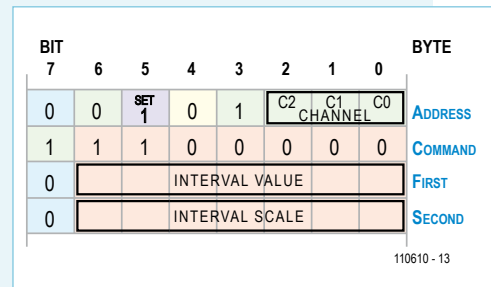
The figure shows the four-byte application protocol command. The interval value is encoded in seven bits. The coding is as follows.

Hex	Dec	Interval
04	4	1 µs
05	5	10 µs
06	6	100 µs
07	7	1 ms
08	8	10 ms
09	9	100 ms
0A	10	1 s
0B	11	10 s
0C	12	100 s
10	16	1 minute
11	17	10 minutes
12	18	100 minutes
18	24	1 hour
19	25	10 hours
20	32	1 day
21	33	10 days
22	34	100 days
28	40	1 month
30	48	1 year
31	49	10 years

Checksums and reliability

When designing a bus system it is almost impossible to pay too much attention to the problem of how to ensure messages are reliably transmitted under all conditions. In the previous installment in this series we described two acknowledge mechanisms: one at the message level, chiefly designed to detect collisions occurring during the uncontrolled 'free bus phase'; and one at the level of the application protocol, designed to cope with interference and other types of problem. The latter mechanism is even used when collisions cannot occur, as described in the main text. In both cases the receiver sends the received bytes back to the sender, with a simple flag bit discriminating between the acknowledge message and the original message. This approach seemed pretty safe to me, and so I postponed the implementation of a message CRC or checksum.

Fortunately in the meantime many others have started to think about how to make use of the bus. *Elektor* reader Werner Koch came to the conclusion that the acknowledge mechanism was inadequate. The mechanism does allow for a message that is lost to be resent by the sender; but it can also happen that interference causes a receiver to see a phantom message which has not actually been sent. As a result an actuator would generate an acknowledgement of this phantom message, from which the master can deduce that something has gone awry. However, the bad news is that the actuator might already have changed the state of a relay, with potentially unfortunate consequences.

One solution to this problem is to have the actuator wait for a confirmation of the command before switching the relay (a three-way handshake). An alternative is to add redundant information to each message: only when the checksum bits are correct will an actuator switch the relay. Using this approach it is much less likely that random interference will generate a valid message than if there had been no checksum.

Next we have the question of whether to use a 16 bit CRC or a simpler sum. My suggestion of banning the value AA_{hex} from appearing in the last two bytes, which are allocated for the checksum, met with vehement opposition from some participants on the mailing list,

although the advantage it brings of allowing simpler synchronization is unarguable. The alternative proposal was for a 16 bit CRC and a more sophisticated approach to synchronization. For example, the CRC itself can be used to determine when a message is complete.

Finally I made the compromise suggestion of allowing both possibilities. The distinction is indicated by a bit in the mode byte, whose layout is now as follows.

Bit	1	0
7	no ID bytes, payload from byte 2	ID bytes from byte 2
6	bytes 2 and 3 are ID bytes	bytes 2 to 5 are ID bytes
5	no CRC/checksum	bytes E and F form a 16 bit CRC/checksum
4	advanced synchronisation	AA_{hex} does not appear from byte 2 onwards
3	last ID byte is fragment number	all ID bytes used for addressing
2	top six bits give bus segment	no segment address
1	acknowledge message	original message
0	acknowledge message expected	not expected

At least one of the checking functions will need to be implemented in the AVR microcontroller C library that is currently in development.

A little later I had the idea of adding redundancy within the data payload bytes. for example, an important two- or four-byte command could be repeated within a single message, respectively four times or twice. With non-periodic interference the probability of a valid message of this type arising is vanishingly small.

nodes talking to one another under such a scheme, but RS485 in combination with our protocols nevertheless form a good basis for the remote control of sensor equipment. With the sensor no longer being regularly interrogated by the scheduler it needs to be possible to instruct it to emit readings at specified intervals. The text box shows how such an 'interval command' can be encoded using the application protocol.
There is plenty more in the pipeline: in the

next installment we will look at the promised connection to an Android smartphone.
(110610)

What do you think?

Feel free to write to us with your opinions, ideas and applications.

[1] www.elektor.com/110151

[2] www.elektor.com/110258

[3] http://ww1.microchip.com/downloads/en/DeviceDoc/21950e.pdf

[4] www.elektor.com/110428

[5] www.elektor.com/110610

[6] www.elektor.com/110517

[7] http://en.wikipedia.org/wiki/Cascading_Style_Sheets
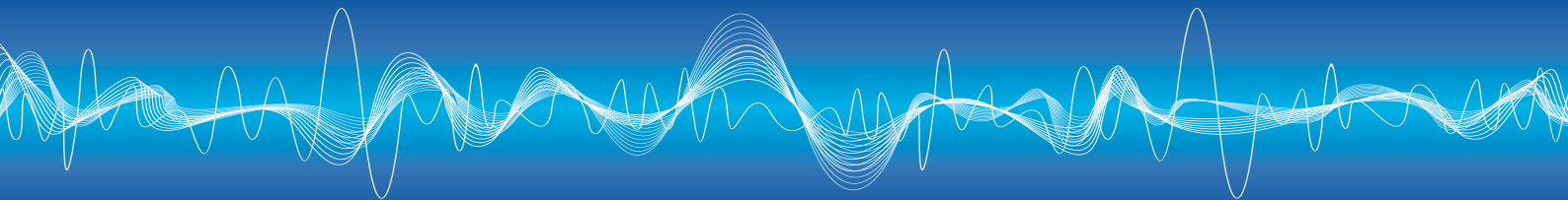
Work in progress

The Elektor Labs generally remain a closed fortress, with access to Elektor House limited to those who happen visit the castle during one of the annual Dutch Open Monument Days. Whatever happens in the lab usually remains behind closed doors. That, of course, does not mean that there is nothing interesting going on. So here's a first, small 'peep show' of two projects that are being worked on in the lab at the moment. First the new LCR Meter. Designer Raymond Vermeulen and colleague Jan Visser are busy with building the prototype...
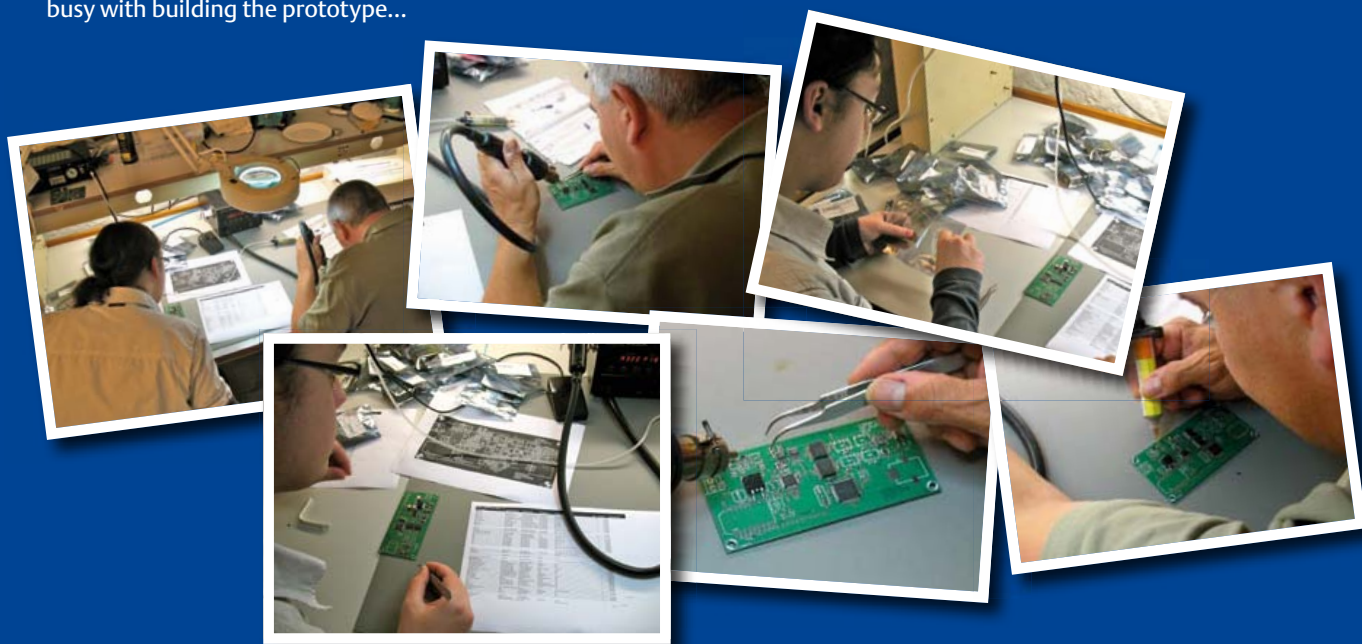
The second photo series shows the efforts of Head of Design Chris Vossen working on his version of a 3D printer. The publication date remains a secret for now, but for more information about this exciting project and the latest state of affairs you can take a look at www.techthefuture.com/3D-printer.
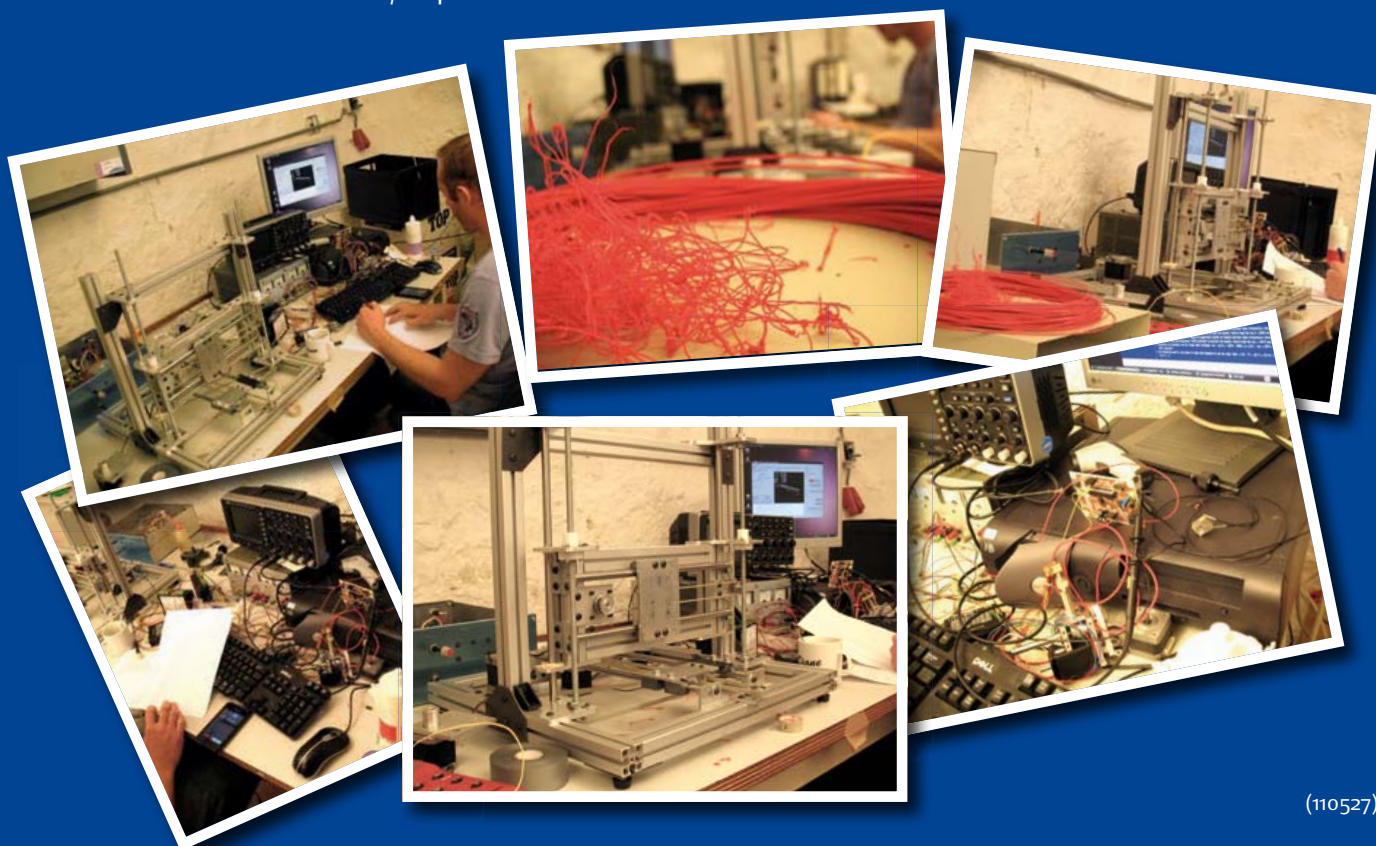
(110527)

E-LABs INSIDE

LED Exorcism

Ghostly goings-on (2)

By Dr. Thomas Scherer (Germany)
& Elektor readers

In the September 2011 issue I recounted how I had witnessed the demise of two different LEDs both of which inexplicably began to flash on and off before they died. In the absence of any reasonable explanation we blamed it on an LED spirit. The ink was hardly dry on the September edition before your replies started coming in, some interesting, some informed, some jesting and some, well, bizarre. Just to recap before we start with your replies: a standard low power LED used in a recharging circuit of an electric corkscrew and a high powered display LED both showed the same behavior when they were about to give up the ghost; they started flashing. In both cases the circuitry around the LED couldn't be much simpler, a series resistor to limit current and an AC power adapter to supply, ermm, current. It's no wonder engineers were rubbing their eyes in disbelief. We couldn't explain the phenomenon; the editors agreed we should draw on the combined wisdom of our readership to help lay the ghost:

Wolfgang Bredow from Lilienthal, Germany wrote:
"When I read your article in the September edition it immediately put me in mind of an experiment I carried out in the late 1970s. At that time I was using a curve tracer to record the operating characteristics of electrical components. On a whim I decided to over-drive an LED to record its characteristics as it failed. The resulting curve can be seen in **Figure 1**.
The curve shows the victim (a green LED) being driven way beyond is recommended safe operating area. With the supply approaching 7 V the LED was drawing around 500 mA when the emitted light changed to dark red. Next it began to flash (aha!!!), the electromechanical curve tracer now started to go a little crazy trying to keep up with the step changes in the LED's characteristics. The dashed line indicates where it failed in this task! "
Mr Bredow assured us that this was the only time ever he had been guilty of deliberately causing the demise of an innocent component...

Craig Hyatt writes from a .com email address:
"The answer Is pretty simple. The LED is made of a variety of materials with different expansion coefficients. When the LED cycles on and off, the materials expand and contract at different rates, and this causes mechanical stress that can cause a bond wire to separate from the substrate. If the bond wire makes contact when the
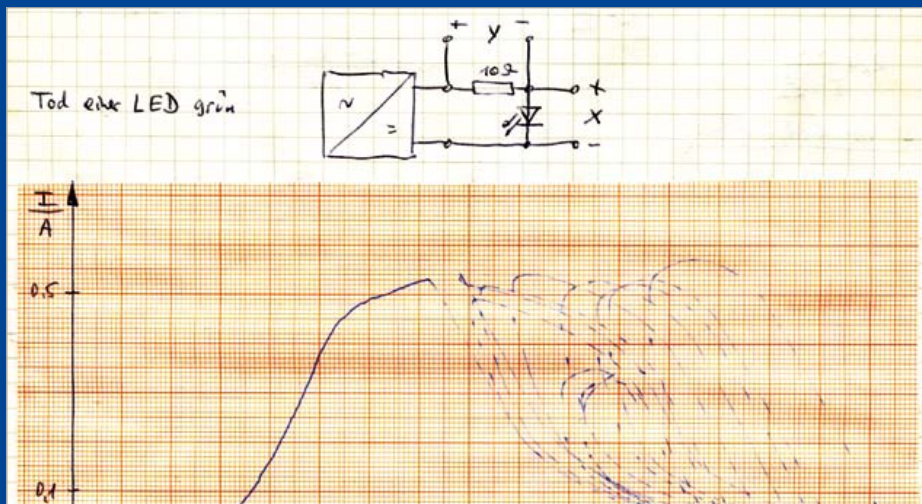
Figure 1. Original curve trace of a failing LED from the 1970s.

LED is cool, but loses contact when the LED heats up, then the LED will blink until the break becomes permanent. See: www.emsnow.com/cnt/files/White Papers/DFRLEDFailures.pdf."
Craig was not the only one to point the finger of suspicion at the wire bond contact point on the LED chip as the cause of the intermittent behavior. One reader even went on to suggest a fix...

Karl-Heinz Ziener from Germany:
*"I would guess that a hairline crack has developed in the bonding wire to the LED chip. When power is first connected the LED is cold and the two parts either side of the crack are touching, current flows and the LED lights up. The LED warms up, expands and breaks the contact. Now with no current flowing the LED cools down, contracts and makes contact again, so the cycle repeats.
A (not entirely sincere) repair bodge suggestion is shown in* **Figure 2**: *Connect a capacitor in parallel across the LED. When the LED goes out the capacitor will charge to full battery voltage. As the LED cools down contact is made again and the capacitor releases its stored charge through the LED, permanently welding the two parts either side of the crack together, job done!"*
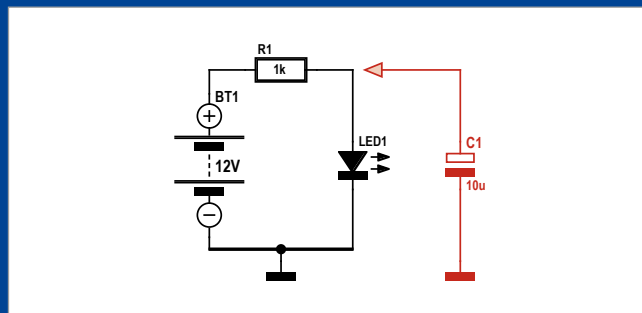
Figure 2. Heal Thyself: Charge stored in the capacitor should be sufficient to weld the wire bond back on the chip.
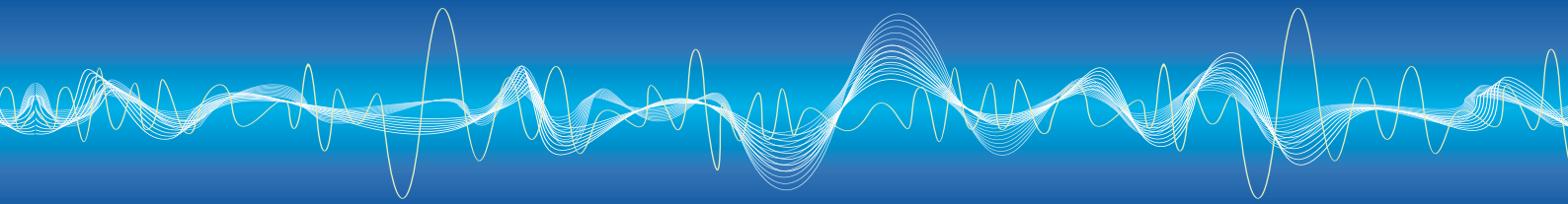
For this to have any chance of working the LED must still be partly working i.e. flashing but not completely dead. ;-)

Gwyn Evans in Munich got in touch via his smartphone:
Obviously the reason for your observations is that the 9 VDC isn't quite DC, it's half rectified AC and what you are observing is a localized breakdown in the space-time continuum, resulting in a 50 times or so reduction in perceive time. This is nothing to worry about, and due to the induced effect resulting from Lorentz Contraction your flat will appear bigger too.
Alternatively the result is probably due to a heating effect. As an LED ages it can become more sensitive to ambient temperature. The LEDs temperature rises causing the LED to fail. The LED then cools allowing it to recover and start working again.
Right, OK, glad we got that one sorted out then… Gwyn went on:
"It is possible to extend the life of an LED by pulsing it; usually 1 kHz is sufficient."
I'm wondering now if I should suggest to the utility companies that they derive the powerline frequency directly from the output of wind turbines…

Hubert Maiwald from Neutraubling, Germany:
"The majority of LEDs consist of a single die cemented to the lead frame with silver-loaded conductive adhesive forming one electrical connection and a wire bond forming the other contact on top of the die. This adhesive is the weak point; it can be damaged by excessive solder temperature during installation or dissipation of too much power in operation. At high temperature the adhesive starts to give off gas, generating voids in the interface between the die and lead frame. This damage degrades the thermal path and reduces the LED's power handling abilities further, leading to more localized adhesive gassing and more voids. As die temperature increases the emitted light wavelength shifts, becoming longer.
Eventually there is very little contact area remaining and the LED mould material becomes locally heated so that expansion of the materials exerts pressure on the die and interrupts the current flow. With no current flowing the die temperature falls and contraction pulls it back to its original position, re-establishing contact. The die now heats up again and the process repeats…
The thermal time constant of this oscillation is in the range 0.1 to 10 s, depending on current and power dissipated. This flashing mechanism is effectively the same as the old mechanical bimetal relays. Incidentally of all the ways an LED can be destroyed this mechanism is not particularly probable. For it to occur it must be assumed that the diode junction is not damaged and also that the bond wire is still intact. More often than not the die bond adhesive just becomes irreversibly damaged and the LED simply goes open-circuit."
Seems logical?

So, dear reader, did any of that sound like a plausible explanation? The consensus suggests a link to thermal effects but that is the fascination with electronics there are always puzzles to solve, sometimes in the most unexpected places.

(110668)

Pins to length

By Thijs Beckers (Elektor Netherlands Editor)

In a previous E-LABs Inside installment we already mentioned placing a piece of experimenter's board between an LCD and a (mother)board, with the objective of making the display easier to remove and without damaging it. This was targeted specifically at the more fragile displays, such as the DOGM-series made by Electronic Assemblies. Now there is even a second trick for this experimenter's board.
The DOGM displays mentioned above 'stand quite tall on their legs' (have quite long connecting pins). This length is necessary when they will be provided with a backlight that — and the name gives it away already — is fitted behind the display. But when this backlight is omitted, it is often much better that the display is mounted closer to the PCB, so that, for example, the entire assembly is more compact when building into an enclosure. When there are sockets on the PCB for connecting the display, then the only option for fitting the display closer to the PCB is to shorten all the pins. It is then desirable that all the pins have the same length. And here is where our experimenter's board comes in handy again: insert the pins of the display as far as they go into the board and cut off the part that sticks through. The length of the pin that remains is then perfect for plugging into a header. And all the pins have exactly the same length, so that the display cannot not rock in its socket.
Do you have any handy tips for us?
Mail them to editor@elektor.com.

(110664)
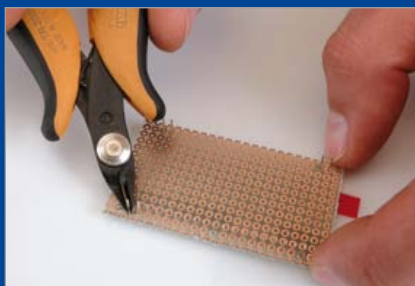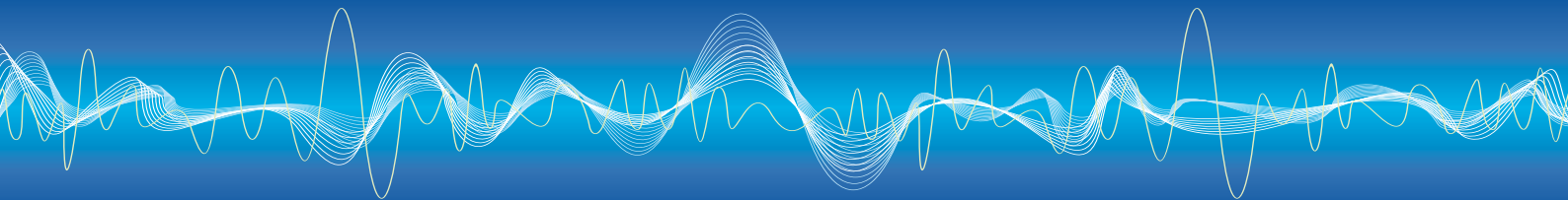
Itsy Bitsy Spider...

By Raymond Vermeulen (Elektor Labs)

After weeks of devotion to a project it has finally reached that stage: you order the circuit board, the components and when everything has arrived you start the assembly. Then it turns out that the footprint for one of your ICs is not right. What now?
In my case the culprit was a DS2003 relay driver. Since 2009 the version in the TSSOP-package is (apparently...) no longer made. Only the SOIC version still runs off the production line. It is nice that we now know that, but too late!
Instead of ordering a new circuit board with the footprint corrected, which takes another few weeks (faster is possible, but gets a bit expensive) and wasting the time and effort that you have invested in that part of the board you have built already, it is time to improvise.
The solution in this case (see photo) is 'quick' but 'very dirty'. Fortunately this IC is only used to switch a few relays. In applications where the signal quality is important, this method is not so suitable.
What I have done: First I soldered a piece of very thin enamelled wire to each of the little legs. Then I cut all the wires to the same length and bent the ends over. The reason that the wires are this long, is not only so that you can get a soldering iron in
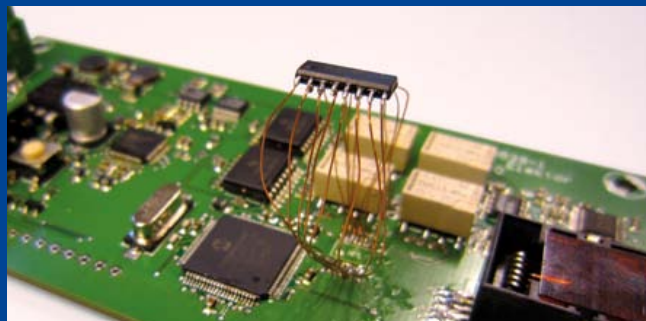
between, but also because otherwise the other end of the wire is likely to unsolder because of heat conduction when you are soldering this end.
If you have a steady hand then the soldering is not that difficult. For people with a slightly less steady hand, such as me, this becomes rather tricky, but I was still successful in the end. It's a good idea to fix the IC in place first and to strip the wires of enamel where they will be soldered. But the best solution of all is just to use the correct footprint on your circuit board...

(110692)

Smelly bus

By Thijs Beckers
(Elektor Netherlands Editor)

While quietly working on the article about the use of cheap tablets in embedded electronics (see elsewhere in this edition) I suddenly noticed that typical smell of burnt electronics. Following my nose (and as a member of the First Aid team I'm of course more or less obliged to act on the smell of fire), I arrived in the adjoining room with the German editorial staff, where colleague Jens Nickel was working on the Elektor Bus project. Was, because it appeared that the tantalum electrolytic capacitor on the USB/RS485 converter, which is part of the Elektor Bus system [1], had given up (see photo). Jens was already back in the lab with the faulty board, but an odor like that of burnt Bakelite was still in the air.
We were of course not at all happy with this. What if this could happen to all the boards that have already been sold? Was this a manufacturing fault or are the tantalum capacitors mounted the wrong way around? Do we have to recall all those boards out there? Is this our fault or that of the assembler? Who will go ballistic? Who will foot the bill?
Checking the stock that was still in the warehouse should give a definitive answer. It was already towards the end of the afternoon, so this would have to wait till tomorrow, because our warehouse is in an industrial area about 6 miles from Elektor House.

In short, the next day we immediately checked whether the remaining stock (and therefore also the boards already supplied) were built correctly or whether we had a big problem.
What emerged: We could breathe a sigh of relief, because the boards we pulled out of stock were built correctly. What we noticed though, is that different components were used on the boards we have in stock. And then it dawned on us... The modules that Jens was using were early prototypes built in the lab, hence the different components compared to the production version. Probably whoever assembled them didn't realize that with tantalum capacitors the + is marked (with electrolytic capacitors the – is usually marked).

So this fortunately ended in a fizzle (well, more like smoker). So here you have one example of all the kinds of things you can come across during the development of a project. It is certainly not likely to be boring! What surprised us quite a bit however, is how long this tantalum capacitor was able to hang on while being reverse polarized. There was nothing unusual to be noticed about the functionality of the circuit.

(110693)

MIAC – the rugged PIC

What does it do?

MIAC is an industrial grade control unit which can be used to control a wide range of different electronic systems. It has a lots of applications in industrial control and automation and is perfect for hobbyist PIC projects that need a little oomph.
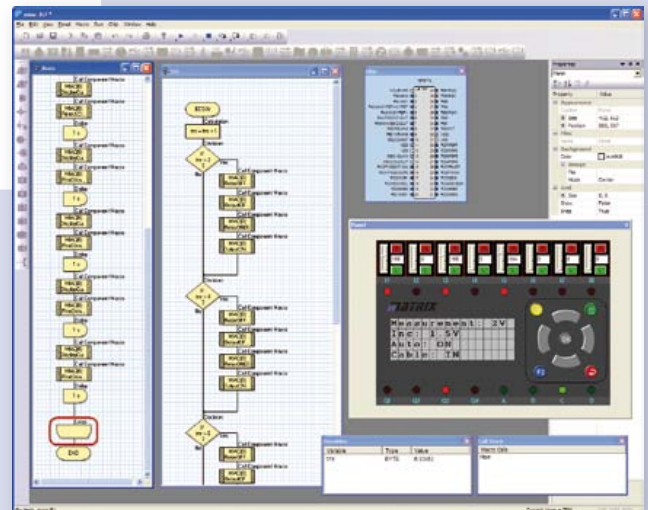
Flowcode – the graphical programming language supplied with MIAC

Key

1. Top hat rail mounting recess
2. 16 character x 4 line LCD display
3. Power LED
4. Input status LEDs
5. 2.1mm power jack
6. Screw terminal inputs
7. Top hat rail retainer clip (upper)
8. Reset / run switch
9. USB socket
10. USB transfer LED
11. Control keys
12. M3 mounting holes
13. Motor status LEDs
14. Motor output screw terminals
15. Top hat rail retainer clip (lower)
16. Relay output screw terminals
17. Relay output status LEDs

Using MIAC with FlowKit to give full IN-Circuit Debug with Flowcode

Benefits
- Flexible and expandable
- Easy to program with flowcharts, C or Assembly code
- Physically and electrically rugged

Features
- Programmable from USB
- Based on PIC18F4455
- Shipped with a free copy of Flowcode (worth $192.00)
- Compatible with third party C compilers
- 8 digital or analogue inputs
- 4 relay outputs @10amps
- 4 motor outputs @500mA with speed control
- 4 line LCD display and control keys

Audio DSP Course (6)
Part 6:
Digital signal generator
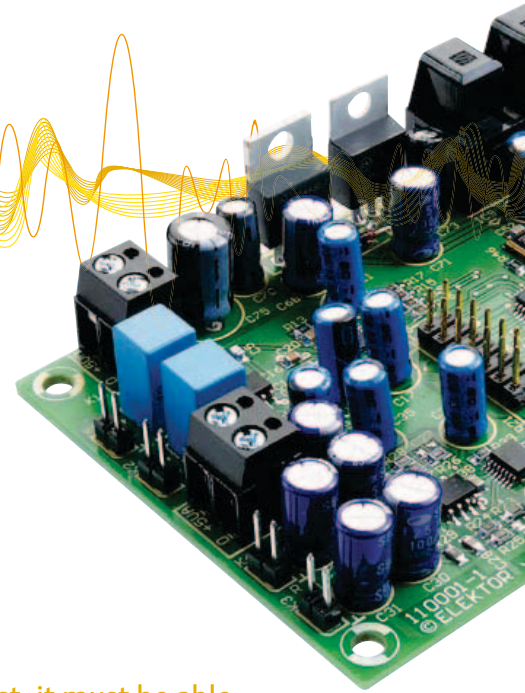
An audio signal generator for lab use needs to have two basic features. First, it must be able
to generate low-distortion sine waves with an adjustable frequency and amplitude to allow
measurement of the frequency response and distortion factors of audio equipment; and second, it
must be able to generate low-pass and band-pass noise signals to help characterise electro-acoustic
transducers. A DSP makes the ideal basis for such a device. The DSP board we have described in this series
of articles is used here as the central component in a lab-grade signal generator, generating high-quality
output in both digital and analog forms.

By Alexander Potchinkov (Germany)

An audio signal generator is a fundamental
item of test equipment in any small audio or
electro-acoustics laboratory. For the utmost
in flexibility the generator should ideally be
able to produce output in analog or digital
form and have two output channels, allow-
ing test signals to be fed simultaneously to
analog and digital devices. It is useful to be
able to carry out measurements both in the
lab, such as on one's home brew loudspeak-
ers, and on devices when installed at their
final place of use, for example to measure
the frequency response of a speaker sys-
tem when installed at a large event. We also
want to be able to measure both linear and
non-linear distortions in equipment.

To this end the audio signal generator
described in this article provides test sig-
nals of sufficiently high quality to ensure
that such measurements can be made accu-
rately. If the top end of the frequency range
required is not unreasonably high (in most
cases for audio measurements the audible
range from 20 Hz to 20 kHz is plenty) then
we can realize the generator using the DSP.
The result is an unbeatable combination of
low cost and high signal quality: for sine

wave signals this means low distortion and
low noise, and for noise signals it means
that the characteristics of the noise filter
are tightly controlled. Analog signal gener-
ators (under comparable conditions) can-
not match the performance of our digital
system. To turn the DSP board into a piece
of test equipment we need to add a user
interface to allow the entry of various set-
tings and, if wanted, an extra analog out-
put stage. In this article we will describe the
signal processing and the DSP program that
implements it, closing with some notes on
how a user interface can be constructed.

Signal processing in the digital audio generator

Figure 1 shows the block diagram of the
signal processing involved in one of the two
identical channels in the system (which we
shall call the 'left' and 'right' channels), fea-
turing two basic signal sources. One is a sine
wave generator with adjustable frequency
and the other is a white noise generator.
Following the noise generator is a bank of
42 spectrum-shaping filters with a switch to
select between them. Then there is a signal
source selection switch and an attenua-
tor. Depending on the switch settings and
the selected filter the system can produce
either a sine wave or filtered noise at its

output. The two channels are independent
of one another, allowing the simultane-
ous generation of a sine wave and (filtered)
noise signals.

The signal processing chain shown in Fig-
ure 1 requires a total of five parameter val-
ues: the phase increment $d\phi$, which sets
the frequency of the sine wave, the index
F_I of the noise filter (regarding this index
as a switch setting), the position of signal
source selection switch S_1, the attenua-
tion factor α and the noise signal amplifica-
tion factor 'Gain', which can be set in steps
of 6 dB (that is, each step corresponding
to a doubling in output amplitude). Using
the parameters α and Gain in combination
allows any desired gain value between the
6 dB steps to be obtained. Because there are
two independent channels the DSP code will
need two separate areas each storing a set
of five parameters.

We will now describe the three main blocks
in the signal processing chain in turn: the
sine wave generator, the noise generator
and the digital noise spectrum filter.

Sine wave generator
There are two main approaches to generat-
ing a digital sine wave of the form
$x(n) = a \sin(2\pi n f_S / f_T)$, $n = 0, 1, \ldots$
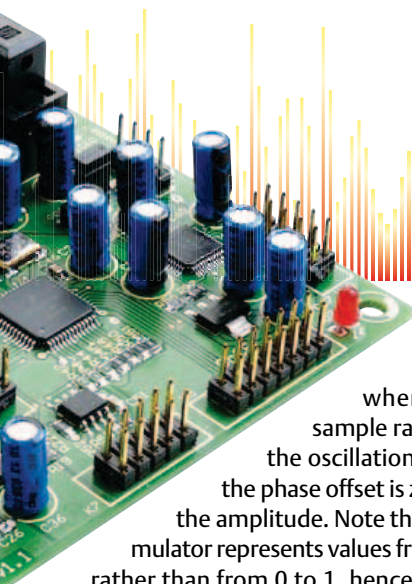
where f_T is the sample rate, $f_S < f_T/2$ is the oscillation frequency, the phase offset is zero and a is the amplitude. Note that the accumulator represents values from –1 to +1 rather than from 0 to 1, hence must work modulo 2.

The first approach is to use an oscillator with feedback, such as the Wien bridge circuit well known in analog electronics. In digital form this is known as a second-order recursive oscillator. In both analog and digital forms if there are stringent requirements on distortion then some form of amplitude control is required.

The second approach is to use a phase accumulator followed by a non-linear function mapper, as used in the 'DDS [Direct Digital Synthesis] RF Signal Generator' project published in *Elektor* in October 2003. In the digital sine wave generator the phase accumulator produces a sawtooth output with a period equal to that of the desired sine wave. The non-linear mapper modifies the sawtooth signal into a sine wave. In analog electronics this can be achieved using a network of diodes and with the phase accumulator outputting a triangle wave, but unfortunately the design requires a lot of adjustments and is sensitive to temperature-dependencies of the components, and does not usually give satisfactory results. The function mapper in the digital version usually takes the form of a 'look-up table' storing one cycle of a sine wave sampled at perhaps 1024 equally-spaced points in time. If the output value from the phase accumulator should not fall exactly on one of the stored sine wave sample points, linear interpolation can be used between the two neighboring values. One disadvantage of this approach is that signal distortion depends on frequency and becomes worse if more points need to be derived using interpolation. An alternative to linear interpolation is to use a polynomial approximation: one one hand this requires rather more computing power to evaluate the approximating polynomial, but on the other hand it is often possible to reduce the required size of the look-up table and guarantee a lower level of distortion; however, the degree of

distortion still rises with frequency. We have chosen to use this last approach as the spectral quality of the sine wave output is our primary goal. In other words, going to the effort of generating a sine wave digitally is only worthwhile if we can achieve a lower distortion figure than can be obtained using affordable analog technology. More-

over, the computation burden can easily be handled by the DSP, and so we have no excuse for using the simpler approaches. Interested readers may like to know that the polynomial coefficients used in the code were obtained using a Chebyshev approximation, and that we use a degree-11 interpolating polynomial evaluated using the Horner scheme.

The phase accumulator is initialized to zero. For each new sample its value is increased by the phase increment $d\phi = 2f_S/f_T$. If the sample rate f_T is 48 kHz and we want to generate a sine wave with a frequency f_S of 3 kHz, then the increment will be $d\phi = 2 \times 3/48 = 0.125$. The two's complement

arithmetic used by the DSP56374 lets us view numeric values as being arranged in a circle and automatically produces the desired sawtooth waveform without further processing. The value in the phase accumulator is held in double-precision format (as a 48-bit quantity) and is moved to and from memory using the processor's 48-bit

move instructions. The frequency stability of our oscillator is determined by that of the master crystal oscillator on the DSP board and should easily be enough for any audio application.

Noise generator

For the noise generator there are again several implementation options. We have gone for the most straightforward, using a shift register with feedback. In this case, however, 'straightforward' does not equate to 'of poor quality'. The shift register approach has been used for a long time in constructing analog noise generators with the help of a couple of digital logic ICs. If you take a look at the DSP code, you

Figure 1. Block diagram of the signal processing chain.

Table 1: Filter selection indices

Filter index	Filter
0	White noise (no spectrum shaping)
1	Pink noise filter
2	One-third-octave filter, center frequency f_m = 25 Hz
…	…
31	One-third-octave filter, center frequency f_m = 20 kHz
32	One-octave filter, center frequency f_m = 31.5 Hz
…	…
41	One-octave filter, center frequency f_m = 16 kHz

Table 2: Gain points of one-third-octave and one-octave filters with center frequency f_m = 1 kHz

Gain	One-third-octave		One octave	
	left	right	left	right
−3 dB	895 Hz	1117 Hz	718 Hz	1393 Hz
−20 dB	790 Hz	1266 Hz	509 Hz	1958 Hz
−40 dB	611 Hz	1636 Hz	291 Hz	3385 Hz
−60 dB	385 Hz	2581 Hz	145 Hz	6507 Hz

Table 3: Program parameters and valid ranges

Parameter	Range*	Data type	Word length	Alignment
SL, SR	[0,1]	Integer	24	right-aligned
FiL, FiR	[0,1,…,41]	Integer	24	right-aligned
DphiL, DphiR	(0,1)	Fractional	48	left-aligned
AlphaL, AlphaR	(0,1)	Fractional	24	left-aligned
GainL, GainR	[0,1,…,6]	Integer	24	right-aligned

* Square brackets indicate that the range interval is 'closed', in other words, that the limiting values are included in the range. Round brackets indicate an 'open' interval, where the limiting values are not included.

Table 4: Default parameter values

Program parameters (left channel)		Program parameters (right channel)	
Parameter name	Default value	Parameter name	Default value
DphiL	0.041666667	DphiR	0.041666667
FiL	18	FiR	18
SL	0	SR	1
AlphaL	0.5	AlphaR	0.5
GainL	0	GainR	1

Table 5: Program files for the audio signal generator

AudioGen.asm	Main program
Kocz_SinCoef.tab	Sine wave polynomial coefficients
ElektorFilter.tab	Digital filter coefficients
src4392.tab	Byte sequence for configuring the SRC
ivt.asm	Interrupt vector table entries for the audio interrupts
esai4r2t.asm	Audio interrupt service routine: four input channels, two output channels
mioequ.asm	Handy names for the DSP I/O addresses

will see that the software realization is very simple indeed. The noise output is actually a periodic signal (or 'pseudo-noise'): in one period of the signal the shift register passes through every allowable state. Only one state (the register containing all zeros) is not allowable: if the shift register does get into this state it can never leave it.

To give a concrete example, if the shift register consists of four flip-flops then of the sixteen possible states $15 = 2^4 - 1$ are allowable, corresponding to the numbers from 1 to 15. We use a shift register comprising 24 flip-flops, giving a total period of around six minutes at a sample rate of 48 kHz. This is long enough for any audio application. The two generators have different feedback combinations which ensures that the two outputs are for practical purposes uncorrelated and statistically independent: this is important when making two-channel measurements. Again we put emphasis on the quality of the signal processing in this project.

Filters

The filters are used to shape the spectrum of the noise. The indices of the digital filters within the bank of 42 are given in **Table 1** and the filters themselves are as follows.

- A dummy filter for white noise.

- A pink noise low-pass filter, which generates pink noise from white noise. Loudspeaker experts among our readers will know that using pink noise is important for protecting sensitive tweeters from overheating, and that when testing loudspeakers using one-third-octave analysis bands a pink noise input should give a uniform output.

- Thirty one-third-octave band-pass filters to generate noise within bands one third of an octave wide with nominal center frequencies at 25 Hz, 31.5 Hz, 40 Hz, 50 Hz, 63 Hz, 80 Hz, 100 Hz, 125 Hz, 160 Hz, 200 Hz, 250 Hz, 315 Hz, 400 Hz, 500 Hz, 630 Hz, 800 Hz, 1000 Hz, 1250 Hz, 1600 Hz, 2000 Hz, 2500 Hz, 3150 Hz, 4000 Hz, 5000 Hz, 6300 Hz, 8000 Hz, 10000 Hz, 12500 Hz, 16000 Hz
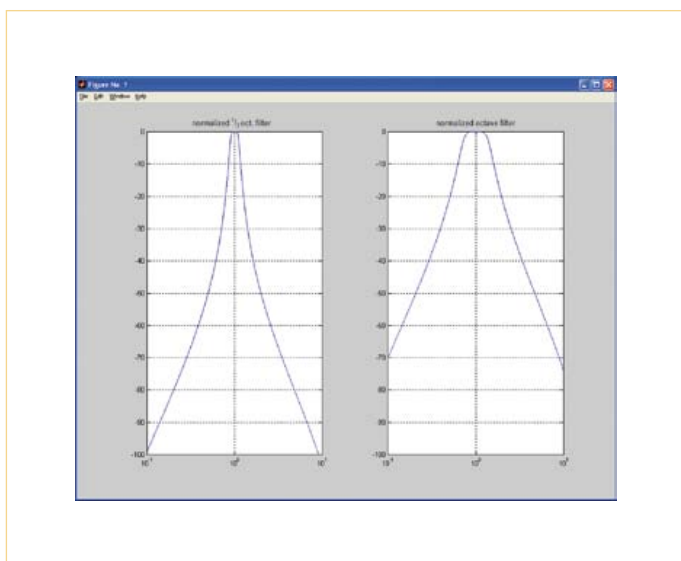
Figure 2. Normalized frequency responses of the one-third-octave and one-octave filters, in decibels.
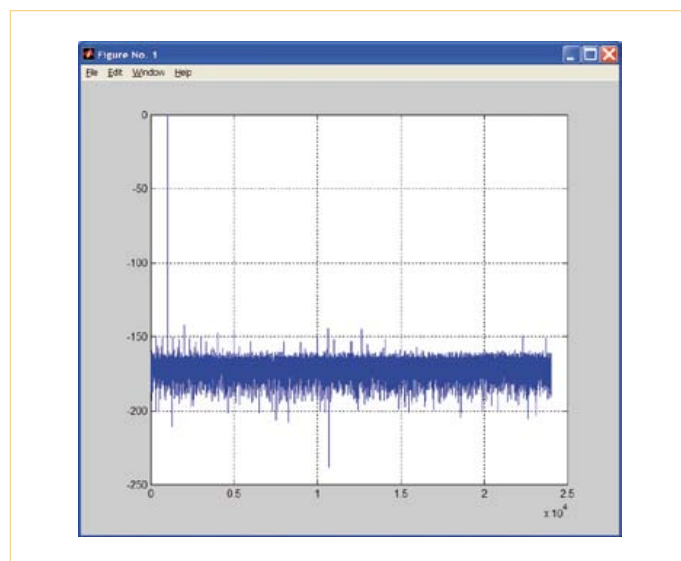
Figure 3. Spectrum of the 1 kHz sine wave oscillator output.

and 20000 Hz: these values are as specified in BS (DIN) EN 61260.

- Ten one-octave band-pass filters to generate noise within octave-wide bands with nominal center frequencies at 31.5 Hz, 63 Hz, 125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz, 8000 Hz and 16000 Hz, again in accordance with BS (DIN) EN 61260.

By way of an example **Figure 2** shows the frequency response of a one-third-octave and a one-octave filter with normalized center frequency. **Table 2** shows various points on the response curve of the two filter types when centered in the middle of the audible range at 1 kHz, giving the frequencies at which certain levels of attenuation occur. The 3 dB points of the filter are highlighted in the table.

Since the filters are implemented digitally and we are using sample rate of only 48 kHz the center frequencies of the filters at higher frequencies tend to deviate from their ideal values. The reason for this is that signals at half the sample rate, or 24 kHz, undergo infinite attenuation: we in effect have an additional null in the frequency response at this frequency. The (not necessarily disadvantageous) result of this is that the left and right sides of the filter's response are not symmetric: the response is steepened on the right-hand side.
All the filters are implemented as sixth-order 'recursive' or 'IIR' filters. Just as in the analog domain, these can be made by cascading

three second-order sections. Our implementation of the sixth-order filter requires eight memory locations to store its state: these memory locations map to capacitors and inductors in the corresponding analog filter realization. The filter also requires a total of fifteen coefficients that determine its characteristics. The complete set of filters thus requires a grand total of $41 \times 15 = 615$ coefficients, which we store in a table. For simplicity in programming we add a forty-second filter which has no function except to pass the white noise signal through unaffected. This filter has index zero.

Selecting a filter is simply a matter of loading the coefficient pointer so that it points to the correct set of coefficients. The value of this pointer is easy to calculate: if the base address of the filter coefficient table is A_B and the filter index is F_I then the pointer address is given by $A = A_B + 15 \times F_I$: this is the start of the relevant coefficient block. Here we see one of the big strengths of digital signal processing: imagine how complicated and costly it would be to create a comparable analog filter bank! The author has a superannuated octave and third-octave filter bank made by Brüel and Kjær in his laboratory: it weighs 15 kg (33 lbs). We could have arranged to have hundreds of sets of filter coefficients stored in the DSP's memory selected just by moving a pointer. For the record, the forty band-pass filters meet the specifications of BS (DIN) EN 61260 class 0. The pink noise filter deviates from the ideal performance by less than 0.1 dB over the frequency range from 10 Hz

to 20 kHz. The band-pass filters are capable of attenuating stop-band signals by more than 100 dB: such performance is difficult to realize in an analog circuit without inordinate amounts of design effort.

The DSP code
The signal processing code itself is embedded as a block in the audio loop. Even though in this case we have no need to read the incoming audio signal, we leave the existing audio loop code as it is and load the input samples as normal. This ensures that synchronization is maintained within the audio loop and means that the framework code is identical for all three of our projects. The code has a total of ten local parameters: **Table 3** lists these along with their valid ranges and other characteristics. The default values of the parameters are chosen so that the left channel emits a sine wave with a frequency of 1 kHz and the right channel a noise signal filtered to one third of an octave around a center frequency of 1 kHz. In both cases the attenuator halves the output sample values: this corresponds to an attenuation of 6 dB. The noise amplifier stage has its gain set to 0 dB (on both channels). The default values of the parameters are listed in **Table 4** and the files that comprise the DSP program are given in **Table 5**.

Subroutines and signals
Two subroutines are called before the audio loop is entered. The subroutine InitState-eVars clears the filter state memories and the two phase accumulators. The shift reg-
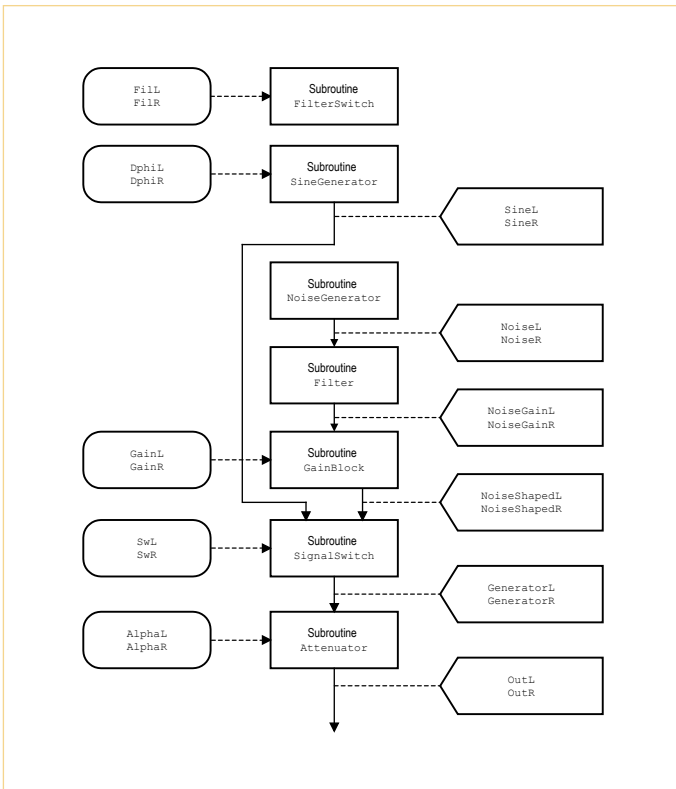
Figure 4. Subroutines and signals in the audio loop.
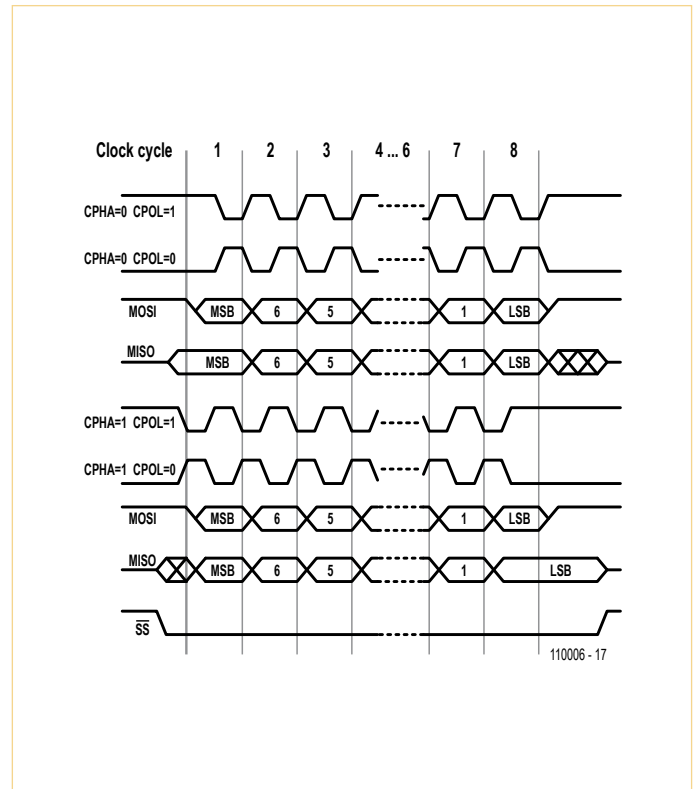
Figure 5. Timing of the SPI port. (Source: Wikipedia)

ister states are set to one of the allowable values. Then the subroutine `SetDefault-Params` sets all the parameters to the values given in Table 4.

Seven subroutines are called from within the audio loop. Six of these are directly involved with the audio signal processing chain. The seventh, subroutine `Filter-Switch`, reads the two filter index parameters `FiL` and `FiR` and from these values calculates the start addresses of the corresponding filter coefficient blocks, each of which contains fifteen entries. Although this operation is more a translation of parameter values than strictly speaking an audio signal processing operation, it nevertheless must occur within the audio loop as the parameters can be changed dynamically. The other parameters are used directly and do not require further processing within the audio loop.

The subroutine `SineGenerator` contains the two independent sine wave generators for the left and right channels. The two phase increment parameters `DphiL` and `DphiR` determine the frequency of the oscillators. The calculation of the sine function using an approximating polynomial is done within the macro `sine`. The

state variables for the sine generator subroutine are the two phase accumulators `l:PhaseAccuL` and `l:PhaseAccuR`, which contain the current phase values. The six coefficients of the approximating polynomial are stored in Y RAM with base address `SinCoef`. The two sine wave outputs themselves are stored in memory locations `y:SineL` and `y:SineR`.

The subroutine `NoiseGenerator` contains the code for the two statistically independent (for practical purposes at least) noise signal generators for the left and right channels. Again, a macro is used for the signal generation proper. The macro has two arguments: the memory location where the shift register contents are stored and a 24-bit constant representing the feedback pattern. The state variables are the two 24-bit shift registers `y:NoiseL` and `y:NoiseR`, which are also the output signals of the two noise generators.

The subroutine `Filter` implements the two noise shaping filters. Each is a sixth-order recursive, or IIR, filter. We take advantage of the library macro `iir2mac` provided by Freescale (formerly Motorola). The macro can be used to implement IIR filters of any desired order and is a standard

building-block. This subroutine requires no parameters, as the base address of the block of fifteen coefficients has already been computed in the subroutine `Filter-Switch`. Two areas for state memory, each of eight locations, are required. Their base addresses are `FstateL` and `FstateR`. The two filtered noise signals are available at `y:NoiseGainL` and `y:NoiseGainR`. The subroutine `GainBlock` contains the code that applies gain to the filtered noise signals. The gain function is not protected against clipping and should therefore only be used to apply small amounts of gain. More detailed information on this is given elsewhere in this article. The code itself is very straightforward: a gain of 6 dB (a doubling of the signal level) is done using a left shift operation. The DSP features a barrel shifter that can shift by any number of places in constant time: this means that we can apply a gain of any multiple of 6 dB in a single step. The two filtered noise signals, after any gain has been applied, are available in `y:NoiseShapedL` and `y:NoiseShapedR`.

The subroutine `SignalSwitch` implements the two signal source switches that select between the sine wave source and the noise source. The subroutine has two

One-third-octave filters and noise signal amplification

Our signal generator can produce noise in one-third-octave and one-octave bands by filtering white noise appropriately. If the white noise source covering the full bandwidth from 0 Hz to 24 kHz has a signal level of $L_R = 0$ dB, the signal level of the filtered noise will be lower as only a fraction of the noise power is passed through the filter. Let us look at this phenomenon in more detail. The bandwidth B (the difference between the lower and upper cutoff frequencies) of an ideal filter with center frequency f_m is given by

$$B = (2^{1/6} - 2^{-1/6}) \times f_m = 0.2316 \times f_m$$

for a one-third-octave filter, and

$$B = (2^{1/2} - 2^{-1/2}) \times f_m = 0.7071 \times f_m$$

for a one-octave filter. Using these bandwidth values we can calculate the resulting noise power in decibels as

$$L_{1/3} = L_R + 10 \log_{10}(f_m) + 10 \log_{10}(0.2316/24000) = L_R + 10 \log_{10}(f_m) - 50.1547$$

for the one-third-octave filter, and

$$L_1 = L_R + 10 \log_{10}(f_m) + 10 \log_{10}(0.7071/24000) = L_R + 10 \log_{10}(f_m) - 45.3073$$

for the one-octave filter.

We collect these results in a table below.

Amplitude of filtered noise signal		
Center frequency f_m (Hz)	One-third-octave filter output amplitude $L_{1/3}$ (dB)	One-octave filter output amplitude L_1 (dB)
25	-36.18	
31.5	-35.17	-30.32
40	-34.13	
50	-33.17	
63	-32.16	-27.31
80	-31.12	
100	-30.15	
125	-29.19	-24.34
160	-28.11	
200	-27.14	
250	-26.18	-21.33
315	-25.17	
400	-24.13	
500	-23.17	-18.32
630	-22.16	
800	-21.12	
1000	-20.15	-15.31
1250	-19.19	
1600	-18.11	
2000	-17.14	-12.3
2500	-16.18	
3150	-15.17	
4000	-14.13	-9.29
5000	-13.17	
6300	-12.16	
8000	-11.12	-6.28
10000	-10.15	
12500	-9.19	
16000	-8.11	-3.27
20000	-7.14	
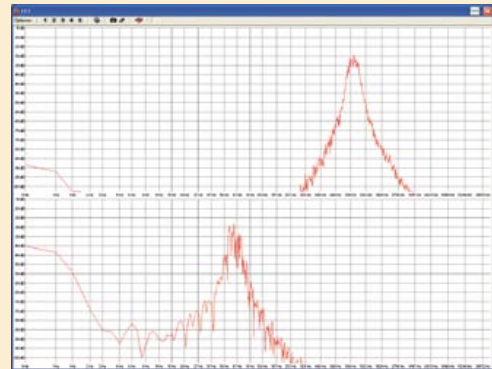
Figure A. GainL = 2, GainR = 3, no clipping.
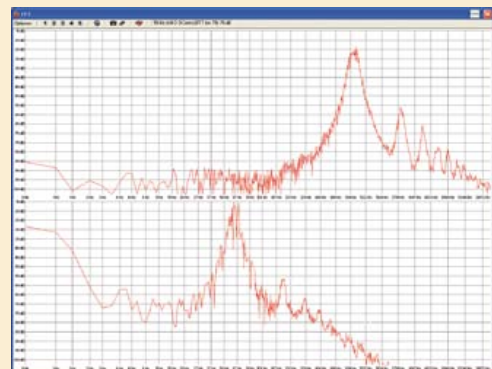
Figure B. Clipping causing non-linear distortion: GainL = 3, GainR = 5.

It is easy to see that the output level rises by 1 dB per one-third of an octave or by 3 dB per octave. If we need higher output amplitudes, particularly at lower center frequencies, we need to amplify the signal. This is done using the parameters `GainL` and `GainR` which operate in 6 dB steps. However, if we apply exactly the gain implied by the above table to the signal, it is possible to cause clipping in the signal generator. This is because the choice of gain must be determined by the peak signal values and not by the average signal power. The values given in the table provide a useful starting-point, however, but it is necessary to check the spectrum of the amplified output. **Figures A** and **B** show the effect of clipping one one-third-octave band-pass filtered noise. The signal generator is set up to generate one-third-octave noise on both channels, the left channel with center frequency $f_m = 1$ kHz and the right channel with center frequency $f_m = 80$ Hz. Figure A, produced using WaveLab, shows the effect of setting the gain on the left channel to 12 dB and that on the right channel to 18 dB. Figure B shows the case with the left gain set to 18 dB and the right gain to 30 dB. In Figure B it is easy to see the distortion components introduced by signal clipping alongside the band-pass filtered noise. When using a high gain it is important to check the output spectrum to verify that clipping is not occurring.

parameters, `SwL` and `SwR`. The routine does not have any state variables or coefficient storage. The two outputs from the switch are available at `y:GeneratorL` and `y:GeneratorR`.

The final subroutine in the audio loop is called `Attenuator`. This implements the two attenuators that appear immediately before the audio output. The attenuation factors are given by the two parameters `AlphaL` and `AlphaR`; again, there are no state variables or coefficients involved. The attenuated signals are available at `y:OutL` and `y:OutR` and are passed through to the audio outputs.

Parameter settings

In order to modify the behavior of the audio signal generator while it is in operation it is necessary to alter the values in the two sets of five parameters, one set for each channel. As the code stands this involves editing the subroutine `SetDefaultParams` to reflect the desired new values, re-assembling the program and reloading it into the DSP using the debugger. It would be much neater if we could provide user controls and a display to allow the parameter values to be changed dynamically and sent to the DSP over its SPI port.

It would also be possible to connect a keypad or digital potentiometer, plus an ordinary alphanumeric LCD, to the DSP itself. However, processing user input at the same time as generating the audio signals makes the code more tricky: it is simpler to separate the two processes from one another and use a microcontroller to manage the user interface. A wide range of suitable devices is available, and there is no great reason to recommend a particular one. The author's preference for such applications is the 68HC11, whose architecture is a particularly good fit with our DSP, although readers embarking on a new project might prefer a device that is rather less long in

the tooth. The microcontroller can be programmed in assembler or (more conveniently) in a high-level language.

The SPI port of the DSP is operated in master mode, and the microcontroller is set up for operation as an SPI slave. The DSP is configured using the registers HCSR and HCKR:

```
bclr    #HEN,x:HCSR          ; SHI disable, SPI reset
movep   #$002048,x:HCKR      ; Cpol=0, Cpha=0, narrow spike
filter,
                             ; f=Fosc / 2 / 8 / 10 = 0.9216 MHz
movep   #$000040,x:HCSR      ; 8 bits, master mode, FIFO off
bset    #HEN,x:HCSR          ; SHI enable

...

move    #Buffer,r0
do #N,RW_MuC
    jclr    #HTDE,x:HCSR,*  ; transmit register empty?
    movep   x:(r0),x:HTX
    jclr    #HRNE,x:HCSR,*  ; receive register full?
    movep   x:HRX,y:(r0)+
RW_MuC
```

these set the clock frequency, clock polarity and phase, as well as the word length, which will normally be eight bits. **Figure 5** shows the various timing options. We need to write a parameter-setting subroutine, which can for example be called as the service routine for an interrupt generated by the microcontroller. On our DSP board the IRQC signal can be used for this purpose.

An alternative approach that avoids external interrupts, and which is simpler if less elegant, is to have the DSP periodically fetch the parameters from the microcontroller and update its internal variables if anything has changed. For simplicity, the polling interval can be set to an integer multiple of the audio sample period. We used a similar technique in the sine wave generator test program described in the fourth installment in this series, where the audio sample clock was divided by 192. The code segment pre-

sented below illustrates from the point of view of the DSP bidirectional SPI communication with polling of the port status flags. It should be possible at least to see from the code the basics of what is required, although some adjustments would be needed in practice.

The code segment writes and reads two buffers of length N, a transmit buffer stored in X RAM and a receive buffer in Y RAM, occupying the same address range. The first four lines of code reset the interface, configure it and then enable it. The rest of the code shows how to write to and read from the buffer.

We end with a small suggestion for a programming exercise. The phase increment $d\phi = 2f_S/f_T$ that is the parameter to the sine wave generator is stored as a left-aligned fractional value. It would be more elegant to store the signal frequency f_S as a right-aligned integer value instead and have the DSP carry out the necessary conversion calculation. This choice of parameter representation also reduces the complexity of the code in the user interface microcontroller.

(110006)

Turn your Oscilloscope into a Reflectometer

By Christian Tavernier (France)

If you're not familiar with transmission line theory, a reflectometer will probably seem like some sort of magical device to you. But using one, you can find out automatically how far a fault is from one end of a cable, without having to physically access the spot. It is used as shown in the figure.

A generator is connected to both the oscilloscope and the cable under test (CUT), which it drives with very fast rise-time pulses. Now transmission line theory tells us that if the cable is terminated in its characteristic impedance, and hence, if it is in good condition, no pulse will be reflected back from the far end of it. And so the oscilloscope displays only the outgoing pulse.

However, if the cable is mismatched, whether by a short-circuit or an open-circuit (cable cut), the transmitted pulse is subject to spurious reflections, and the oscilloscope will then display two pulses: the outgoing one, and the reflected one. We can tell how far the fault is from the near end of the cable simply by measuring the time delay between them.

Obviously, such a device is incredibly useful to professionals, who can avoid having to change great lengths of cable, since they can very easily find the exact location of the fault. But it can also be useful for amateurs — for example, when installing network cabling, or finding out whereabouts your TV aerial down lead is broken.

Sadly, a reflectometer is usually beyond the reach of amateurs because of its high price tag, since, in order to produce a stand-alone instrument, it usually includes the pulse generator with the oscilloscope, as well as a computer section that takes care of calculating the distance to the fault. But as long as you already have an oscilloscope, and you are prepared to do a simple 'rule-of-three' calculation with your calculator, you can build the reflectometer we're suggesting here for around £/€ 20. But don't imagine from the price that this is going to be a cheapskate instrument! It will enable you to make the same measurements as its professional counterparts, as shown for example in the oscilloscope traces illustrating this article.
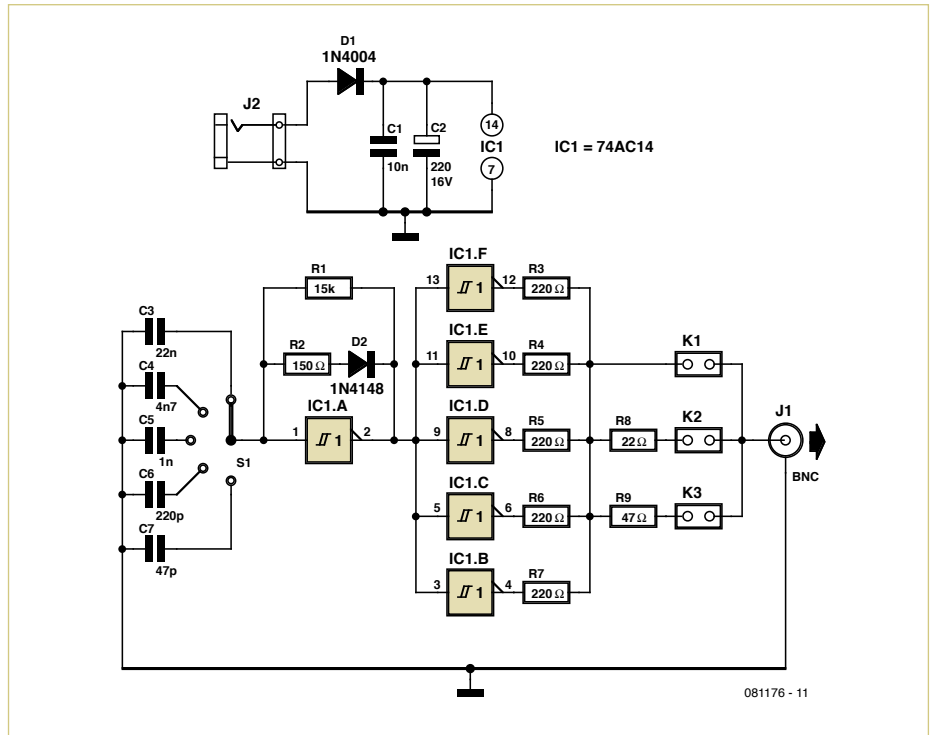
Our reflectometer contains just a single IC, an AC (Advanced CMOS) hex inverter. IC1a is wired as an astable oscillator with a very short mark/space ratio, thanks to diode D2. In this way, it generates very narrow pulses at a relatively low rate. The width of these pulses can be adjusted to various fixed values via S1. For the shortest cables, you need very short pulses, otherwise the reflected pulse arrives before the outgoing pulse has ended, and the oscilloscope trace is unusable. For longer cables, on the other hand, there isn't enough energy in those very short pulses for the reflected pulse to be properly visible, and so we need to use wider pulses that contain more energy.

In order to be able to drive the cable under the correct conditions and at an impedance as close as possible to its characteristic impedance, IC1's other five inverters are wired in parallel; hence the circuit output impedance is determined mainly by R3–R6, to which series resistors (R8 or R9) can be added, depending on whether K1, K2, or K3 is fitted. So we have three output impedances available: 50 Ω with K1, 75 Ω with K2, and 100 Ω with K3, so that the circuit can be matched to the commonest cables.

081176 - 11

081176 - 12
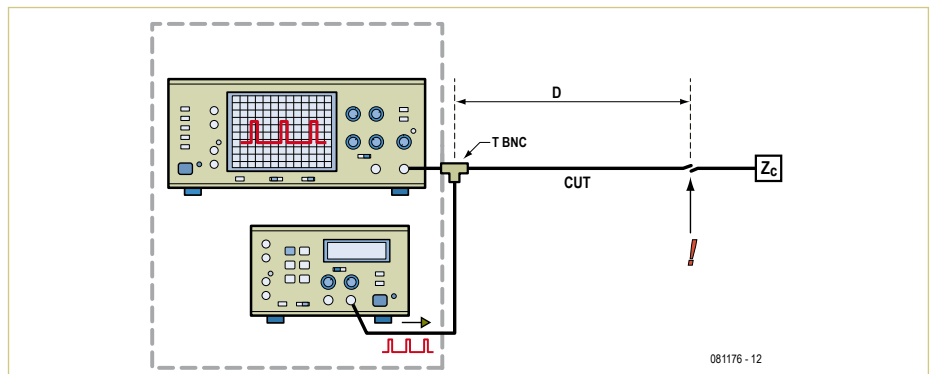
The 6 V power rail should be provided by either a bench supply, if you're in the workshop, or a set of four 1.5 V batteries for portable use. Diode D1 protects the circuit against reversed polarity.

In order for our reflectometer to operate in a stable fashion — the only way to guarantee accurate measurements — we recommend building it on the PCB we've designed for it [1]. This has the advantage of minimizing wiring, as J1 and S1 mount directly onto it. S1 is a vertical PCB-mounting type from Lorlin, with the part no. PT6422/BMH.

Even though the output uses a BNC socket, the circuit isn't only designed for coaxial cables, since the output impedance can be adjusted using K1, K2, and K3. So for other types of cables you'll need to use a suitable BNC adaptor. The instrument is very easy to use, like this:

Fit K1 (50 Ω), K2 (75 Ω), or K3 (100 Ω) according to the impedance of the cable being tested. Connect the circuit up to the input of an oscilloscope and to the cable under test using a suitable T-piece, as indicated in the block diagram. Then turn the circuit on, with S1 in position 1, for example, and adjust the oscilloscope so as to view the outgoing pulse.

If the cable is in good condition and correctly terminated, you will see only single pulses, as shown in **Trace 1**. If the cable is broken, i.e. open-circuited, you'll see a reflected pulse of the same polarity as the outgoing pulse, as shown in **Trace 2**. If the cable is short-circuited, you'll see a reflected pulse of inverted polarity compared to the outgoing pulse, as shown in **Trace 3**.

In either of the last two situations, all you have to do is measure the time between the rising edges of the two pulses (191 ns in our example) in order to determine how far the fault is from the measuring end of the cable. All you need to know is that signals travel at approx 200 m/μs in a coaxial cable, and that the pulse has had to make a return trip to the fault and back. So the distance is given by the equation $D = (V \times T) / 2$, where D is the distance in meters, V is the speed in the cable in m/μs, and T is the time between the two rising edges, expressed in μs. In our example, the time measured was 191 ns, so the fault was at 19.1 m from the near end of the cable.
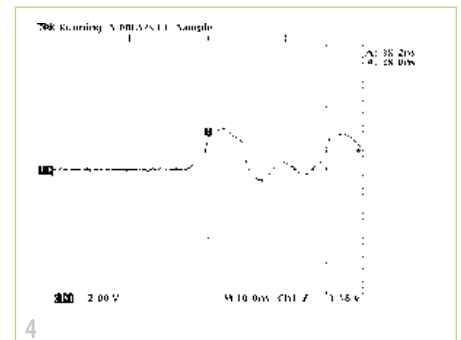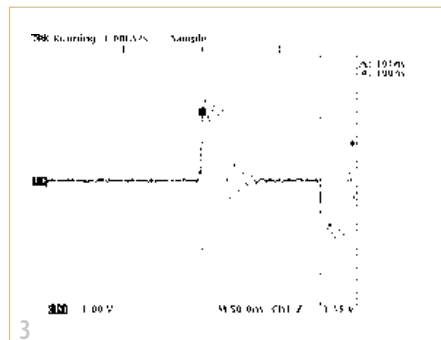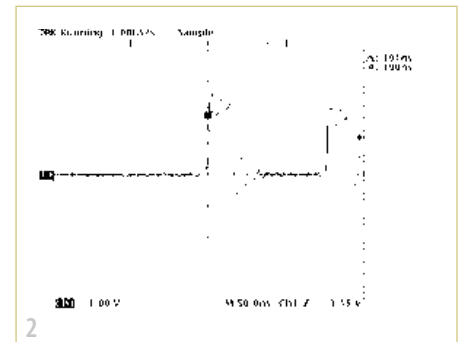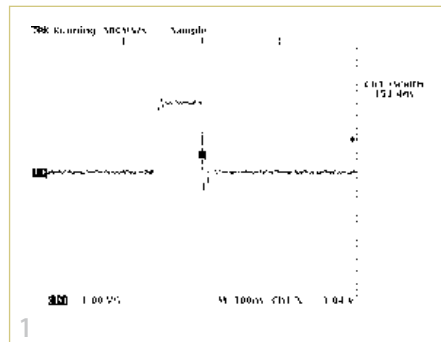
Do note that if the fault is not clear-cut, or for the shortest pulses generated by the circuit, the signals may be quite badly distorted, as shown in **Trace 4**. However, it's still perfectly possible to measure the time between the two rising edges, as shown by this example, where a fault was located 3.8 m from the end of the cable.

Finally, if you want to make accurate measurements, you can use the 'true' value of the signal propagation speed in your cable, in place of the average value given above. All you have to do is look it up in the cable data sheet, where it ought normally to be given.

(081176)

Internet Link
[1] www.elektor.com/081176

Download
081176-1: PCB layout (.pdf), from [1]

COMPONENT LIST

Resistors
R1 = 15kΩ
R2 = 150Ω
R3-R7 = 220Ω
R8 = 22Ω
R9 = 47Ω

Capacitors
C1 = 10nF
C2 = 220µF 16V
C3 = 22nF
C4 = 4nF7
C5 = 1nF
C6 = 220pF
C7 = 47pF

Semiconductors
D1 = 1N4004
D2 = 1N4148
IC1 = 74AC14 (Farnell # 1014022)

Miscellaneous
J1 = BNC connector
J2 = DC supply adaptor socket
K1,K2,K3 = 2-way pinheader
S1 = 5-way rotary switch, Lorlin type PT6422/BMH (Farnell # 1123675)

Pick-proof Code Lock
Using 128-bit AES encryption

By Elbert Jan van Veldhuizen (The Netherlands)

How secure is the remote control of your car or other valuable vehicle? This project shows you how to use a couple of ordinary microcontrollers together with a transmitter and a receiver to implement an IR remote control system that uses a secure code with 128-bit AES encryption and bidirectional IR data communication.

We're all familiar with remotely controlled door lock systems in cars. The remote control transmits a code, and if the receiver of the lock system recognizes the right code, it unlocks the doors. A drawback of systems of this sort is that people with malicious intentions who eavesdrop on the code transmission can effectively pick the lock, since they can transmit the code at any desired time to unlock the car.

A much more secure method is the challenge-response handshake authentication protocol, which is used for many forms of Internet banking. With this approach the lock transmits a specific code and the remote control must perform a defined computation using this code. The result is then send back to the lock. The lock remains locked unless the right computation has been performed.

Eavesdropping on the communication between the lock and the remote control is useless in this situation, because the next time the lock will send a different code for the computation. As long as anyone with malicious intentions does not know the computation, the lock cannot be picked. Here it's important to choose a good computation method. Encryption is very suitable for this. Encryption uses a key to convert data into new data, which is exactly the type of computation that we need for the remote control system.

Encryption algorithm
What makes an encryption algorithm good? Encryption is a process in which data to be encrypted (*in*) is converted into encrypted data (*out*) with the aid of a *key*, which is something like a password.

- The encryption algorithm implements the function $out = f(in, key)$. The inverse function $in = f_{inv}(out, key)$ also exists, but the function $key = f_{key}(in, out)$ does not exist.
- For each value of *in* there is a unique value of *out*. In other words, there are not multiple values of *in* that generate the same value of *out*.
- This also applies to the *key*: two different keys produce two unique encrypted *out* values.

The first condition ensures that if persons with malicious intentions learn the values of both *in* and *out* by eavesdropping, they will not be able to derive the key by using a function f_{key}. The only way to determine the key is to use what is called a brute force attack, which consists of trying all possible keys in the encryption function f. This requires spending so much time searching for the key that trying all possible values takes too long. Powerful computers can try all possible values of a 64-bit key in approximately one day. With a 128-bit key, this would take more the lifetime of the universe, which is long enough to be secure. We chose the AES protocol for the encryption algorithm. This encryption algorithm

Figure 1. Schematic diagram of the base station.

is used in devices such as WiFi routers to prevent eavesdropping on data traffic and breaking into the network. This algorithm has not been cracked up to now. In other words, nobody has found a function f_{key} or some other way to determine the right key with fewer attempts.

The AES algorithm needs a lot of resources compared to what is customary in the microcontroller realm. Over 240 bytes of RAM are necessary for the computations, the code consists of approximately 1500 instructions, and execution of the computation takes approximately 30,000 instruction cycles. Furthermore, tables and arrays are used extensively.

The new PIC16F1827 enhanced microcontroller from Microchip is a device that fulfils these requirements. It has 4096 words of code memory and 396 bytes of RAM. It can also run at up to 32 MHz using its internal clock, and the microcontroller has a new instruction set called "enhanced mid range" that makes working with arrays a good deal easier. Although the RAM is split into individual blocks of 80 bytes each, the enhanced instruction set allows these individual blocks to be viewed as a single large block (linear mode), which facilitates access to tables.

The circuit

Aside from the microcontrollers, only a few components are needed to implement practical circuits (transmitter and receiver) that utilise this encryption method. **Figure 1** shows the circuit diagram of the base station (the lock), while **Figure 2** shows the circuit diagram of the remote control. They communicate using infrared LEDs, in the same was as remote controls for television sets. LED D1 is used for transmission, while IC2 (a standard IR module from Sharp with an operating frequency of 36 kHZ) is used for reception. The switches for operating the devices and configuring the parameters are connected directly to the I/O ports. The 'weak pull-up' capability of the microcontroller make resistors unnecessary here. A keypad for entering a PIN code can also be connected to the remote control. A matrix keypad should be used for this purpose.

The remote control operates directly from two AAA batteries, but a lithium button cell can also be used. The base station can be powered from a mains adapter. The usable supply voltage range is 1.8 to 5 V. Note that the maximum rated voltage of the LF version of the microcontroller is 3.3 V.

Operation

A communication session starts when the remote control sends the code 'A6h'. The base station then generates a random 128-bit number. A random number is better than a predictable number because the lock can potentially be picked by 'code phishing' if a predictable number is used. The encryption algorithm is also an excellent random number generator (see inset), using an input value derived from a counter. The encryption algorithm converts the input value into a random number (using a sepa-
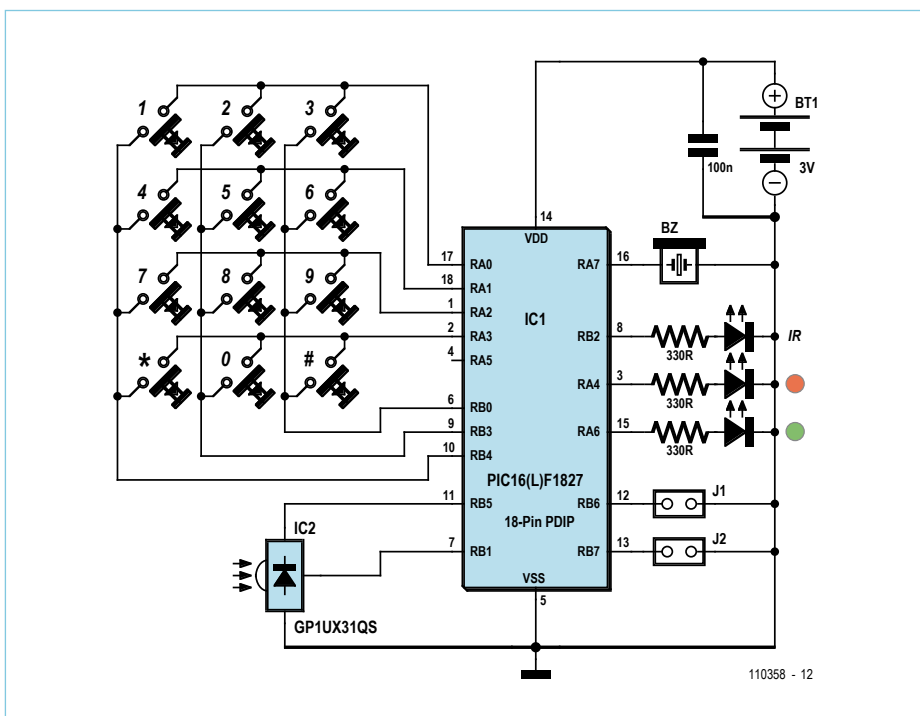

Figure 2. Schematic diagram of the remote control.

Using encryption for random number generation

Linear feedback shift registers are commonly used to generate random numbers. Their output bit streams have the statistical characteristics of randomness, but the bit streams are predictable. As the algorithm is known, the state of the shift register can be reproduced after a specific data set has been read in. This allows the values to be predicted.

A good encryption algorithm also has the statistical characteristics of randomness. Due to the unique mapping from input to output, the ratio of ones and zeros will be exactly 50%. However, the bit stream is totally unpredictable because the key is not known. The pattern repeats itself (or the key can be determined by calculation) only after the entire bit stream has been generated (in this case 2^{131} bits). If this bit stream is transmitted at a rate of 1 Gbit/s, it will take a trillion times as long as the lifetime of the universe to transmit the entire bit stream.
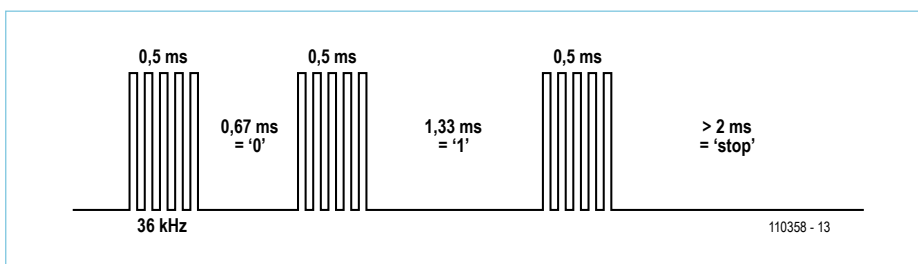

Figure 3. A variant of the Sharp protocol is used to transmit data using IR pulses.

rate key). The counter value is saved in flash memory so that unique numbers can still be generated after a power interruption. As the flash memory has a maximum rated life of 100,000 write operations, the value is saved to memory only once every 65,536 times, and a different memory location is used each time. In the unlikely event that the maximum number of write operations is reached (after 13 million power interruptions or 900 million transactions), an emergency procedure is invoked to ensure that the user is not left standing in front of a locked door. This procedure requires the user to press the remote control button 16 times in a row. After this the random number is derived from the timing of the code transmission by the remote control.

IR communication
The remote control first reads the 128-bit number. Standard modules can only handle a maximum duty cycle of 30% with such long transmissions. The commonly used Manchester coding method (used in the RC5 protocol, for example) has a duty cycle of 50%. For this reason, a variant of the Sharp protocol is used here. The '1' and '0' values are defined by the length of the break between two pulses. A break of 0.67 ms is a '0', while a break of 1.33 ms is a '1'. The pulse width is 0.5 ms, and the end of

the pulse train is indicated by a break lasting longer than 2 ms (see **Figure 3**). The timing tolerances are loose and the algorithm is self-synchronizing, so the accuracy of the clock oscillator does not need to be especially high. This protocol can also be used to transmit 8-bit words (or words of any desired length) as easily as 128-bit words, thanks to the use of a stop bit.

Both the remote control and the base station apply encryption to the 128-bit random number, using the same key. The remote control sends the encrypted 128-bit number back to the base station. The base station compares the received number to the one it computed itself. If they match, the base station opens the lock. Depending on the setting of jumper 1, the base station may return a code indicating a match (0xAB) or no match (0xB5). In theory, returning a result code makes it possible to pick the lock using an automated method, although this is rather unlikely in practice. Nevertheless, if you consider the risk too great you can fit jumper 1 to prevent transmission of result codes.
Jumper 1 on the remote control board has a similar function: the remote control emits a low beep tone if the code is wrong or the base station does not send a response. Fitting the jumper disables this beep.

Generating the key
Jumper 2 enables key programming. This requires switching the base station off and on again. If you now press S1 32 times in a row, two keys will be generated. The red LED goes dark briefly when this has been completed. The timing of the button presses yields totally random numbers due to the speed of the counter. These numbers are stored in the EEPROM. After this the remote control must be programmed with the same key. For this purpose, jumper 2 on the remote control board must also be fitted. After this the remote control sends the code '0xAD' (you may have to press the # button or enter the PIN code first). The base station then sends the key twice. The remote control checks that the two transmitted numbers are the same and then saves the key in the EEPROM (the green LED lights up and a beep sounds). This can be repeated with each remote control unit. Remove the jumpers and switch the units off to restore the base station and the remote control to normal operation.

For security reasons, the key can only be sent to the remote control immediately after it has been generated in the base station. This prevents the 'clandestine' programming of another remote control at a later time. This can only be done by generating a new key, with the result that the original remote control will no longer work, so the action will always be detected.
In addition, data protection of the EEPROM and the program memory is enabled in both the base station and the remote control. This means that the key can never be read out. Furthermore, the key is not known when it is generated because the user simply presses the button, without knowing the value of the key that is generated in this manner. The key is thus stored securely in the microcontroller.

However, there is a risk: when the base station and the remote control are both brand new (not yet programmed), the system is already operational because both EEPROMs are filled with '0xFF', so both microcontrollers have the same key. A user might think that there is no need to program the key, and a person with malicious inten-

tions could use the key 'FF...FF' to try to open locks of this sort. To prevent this, the remote control unit (but not the base station) always increments the key read from the EEPROM, so that the keys are not the same. When the key is programmed the value is decremented, with the ultimate result that the right key is used.

PIN code

The remote control is equipped with a keypad for entering a PIN code. The PIN code is disabled by default. If you do not wish to use a PIN code, simply connect a pushbutton between RA3 and RB0 (this corresponds to the # key on the keypad). The PIN code can be set by fitting jumper 2 and pressing the * key (or entering the current PIN code if it has already been set) and then entering the new PIN code twice in a row. To disable the PIN code, a new PIN code must be set with a value of #### or ****.

If a PIN code has been set, it must be entered when the remote control is activated. If the wrong PIN code is entered three times, the remote control is blocked by erasing the key. After this the remote control must be resynchronized with the base station by generating and programming a new key.

On/off

You may have noticed that the remote control does not have a power switch. The 'problem' here is that the power consumption of the latest PIC microcontrollers is so low that the circuit does not switch off immediately, due to energy storage in the decoupling capacitor. For this reason, we chose a different solution. After five seconds, the remote control enters sleep mode, and in this mode it consumes virtually zero current (much less than the self-discharge rate of the batteries). The # key generates an interrupt and is therefore effectively the 'On' switch.

The software

The author converted existing open source C++ code for the AES routines (the source is stated in the code) into assembly language because the code was not compiled properly by the C compilers for PIC microcontrollers. The program code for the base
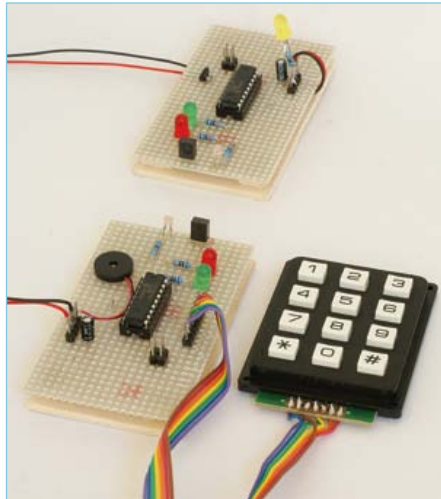

Figure 4. Both circuits can easily be built on pieces of prototyping board.

station and the remote control is located in a single file because many routines are the same for both devices. The correct hex file can be generated by placing '#define remote' or '#define homestation' at the start of the code. Naturally, the code can also be modified.

The IR LED and the IR sensor are connected to the serial port (TX/RX). Other devices, such as a GSM modem, could also be connected to this port. This would allow a lock (or other device) to be actuated securely anywhere in the world by sending text messages.

The user interface

Using the devices is simple after the key has been programmed as described above.
• Press the # key on the remote control to activate it. If the LEDs blink rapidly, the PIN code must be entered. If an incorrect PIN code is entered three times, the red LED blinks constantly and the remote control must be resynchronized.
• Three different beep/visual results are possible at this point:
• High beep tone / green LED – low beep tone / red LED: base station not responding (may be too far away or not switched on)
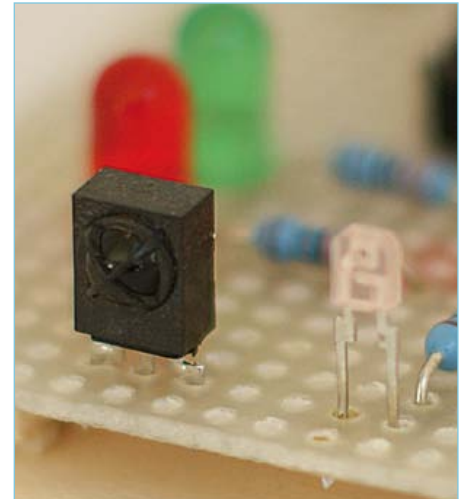

Figure 5. The IR LED and the IR receiver are located next to each other, facing in the same direction.

• High beep tone / green LED – low beep tone / green LED: wrong key; lock remains locked (or the remote control may be blocked)
• High beep tone / green LED – high beep tone / green LED – high beep tone / green LED: lock opened
• If jumper 2 is fitted, only the sequence 'high beep tone / green LED – high beep tone / green LED' occurs in the latter two situations. In this case opening of the lock (or otherwise) is the only indication of whether the right key was used.
• The # key can be pressed again within 5 seconds to send another unlock request without requiring new entry of the PIN code.
• The remote control switches off automatically after 5 seconds without user activity.

(110358)

Note: If you manage to discover a way to reveal the code of this code lock, please let the editors know: editor@elektor.com

LED Cycle Lamp
High-tech on the trail

By Thomas Finke (Germany)

The perfect bicycle lighting system has yet to be invented. They are either not bright enough or badly designed so that they fall off when you go over a bump or have wires that snag or use dynamos that reduce your speed to a dawdle. This design is one step nearer the ideal. It is self contained, powered by Lithium Ion rechargeable cells and has a dazzling 600 Lumen on tap. On full beam the torch will run for around three hours, much longer in 'dipped' mode.

Although a true home brew design this lamp has a truly professional feel to it. The aluminum body is designed to be watertight and requires some machining. The design uses four Li-Ion cells connected in parallel, each with a capacity of 2.2 Ah. A microcontroller takes care of battery management allowing the lamp to be charged at 5 V from either a mains adapter or car adapter.

Lights, cycle, action!

Flashing LED cycle lights are fine for the urban commute; they are highly visible and the batteries last a long time. If you however get the urge to explore unlit country lanes or off-road trails then the chances are, when the sun goes down you will quickly discover their limitations. For the sake of safety you need something more powerful to warn you in good time of the road conditions up ahead. Equivalent professional cycle lighting systems are available but cost hundreds of pounds. This self-contained lamp operates from a single push button to switch between high beam and dipped beam. Conventional vehicle lamps switch between two filaments to achieve a dipped beam but with this design the main beam is just reduced in brightness to avoid the risk of dazzling oncoming vehicles. If you are planning to take your bike further afield it is worth checking the local cycling regulations before you travel. Some countries do not allow lamps that give out too much light. In the UK the regulations only specify that they should emit more than 4 Candelas. Whatever the regulations this robust, versatile lamp should be more than capable of taking anything that's thrown at it.

Operation

The lamp uses just a single push button and operation is quite intuitive. A short press switches the lamp on then further presses toggle the lamp between 'high beam' and 'dimmed'. A long press turns the lamp off. The brightness of the dimmed beam can be set by turning the lamp on with a long press. Now it enters setup mode where further press of PB1 cause the lamp to cycle through increasing then decreasing levels of brightness until the desired brightness is reached. Before long the torch reverts to normal operational mode.

The push button is fitted with an indicator LED which is used here to give a visual indication of the battery's state of charge. Periodically it will issue a series of flashes (four flashes indicating a fully charged battery). When the voltage falls below a threshold the lamp automatically switches to the lowest brightness setting to prolong burn time.

The circuit

White LEDs have a forward voltage drop of around 3.5 V and the voltage output of a single Li-Ion cell varies from 2.7 to 4.2 V
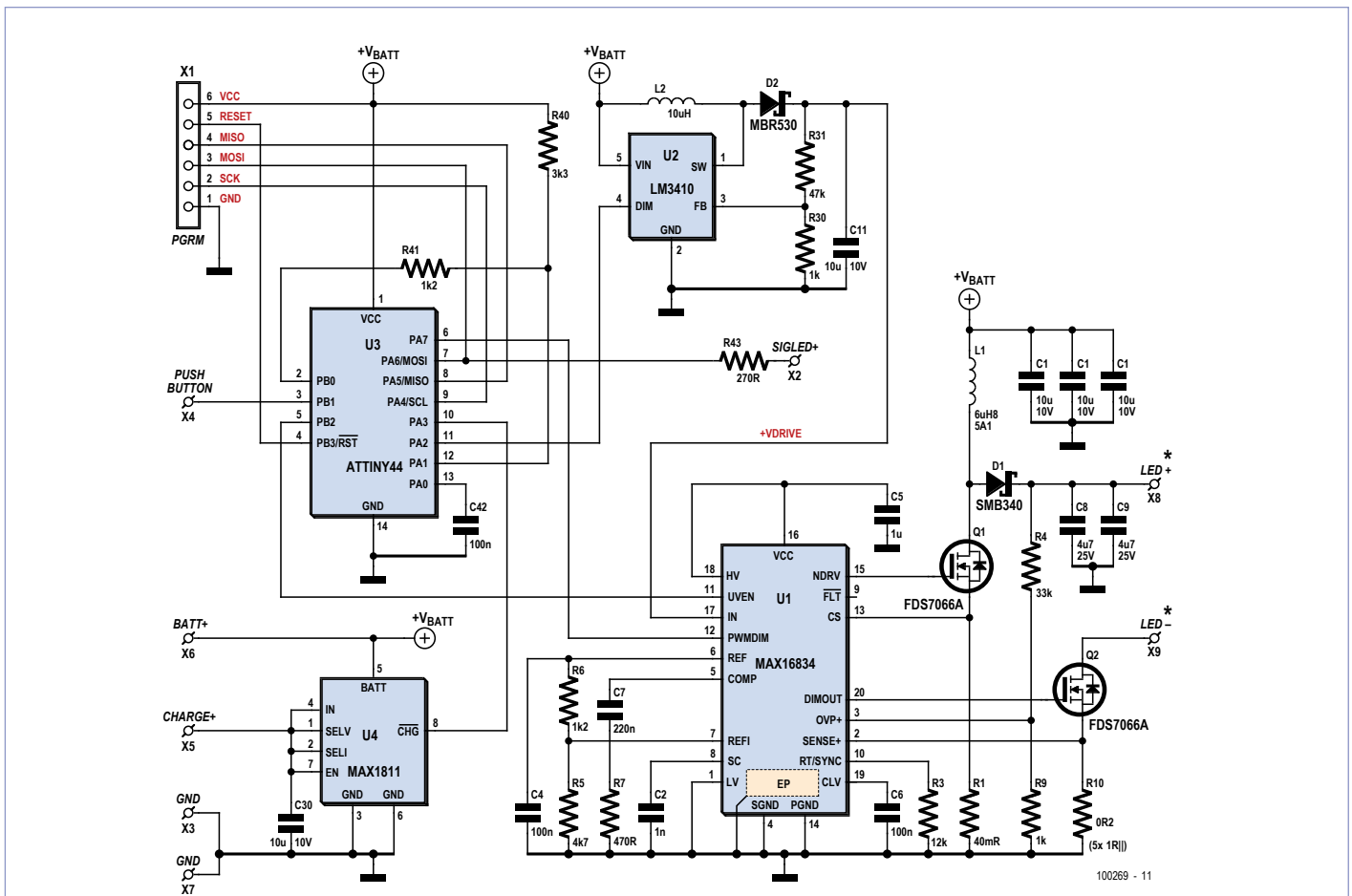
Figure 1. The circuit includes a microcontroller, two switched mode converters and a battery charge regulator.

depending on its state of charge. To operate a single LED from a single cell it would be necessary to use a regulator which provides a voltage initially below the fully charged cell voltage but then above the cell voltage as it discharges. A SEPIC (single ended primary inductance converter) is capable of this sort of operation. The disadvantage of this type of converter (besides the additional circuitry) is that they are relatively inefficient. With this in mind it was decided to use the four LEDs (CREE XR-E) connected in series mounted on a circular PCB and fitted with lens optics made for the CREE LEDs. The regulator required to supply the necessary 14 V is a simpler upwards converter (step-up or boost converter).

The circuit diagram in **Figure 1** shows that a MAX16834 (U1) is used. The IC's configuration in this circuit is taken from an application example in the chip's data sheet, only component values have been changed to make it suitable for this application. Note that with a LED current of 1 A and the battery voltage at its lowest level the coil must

be able to cope with an average current of 5.6 A. The maximum LED current is defined by how many 1 Ω resistors are fitted to the PCB for R10. With all five resistors in place the combined value of R10 will be 0.2 Ω. This will give a maximum LED current of 1 A. The author fitted just three 1 Ω in the prototype to give an LED current of 600 mA. This value produces more than adequate light output and limits power dissipation in the LEDs.

The PWMDIM pin is an input for the PWM signal to dim the LEDs. This is used to reduce the energy dissipated by the LED. Pin 11 (UVEN) would normally be connected to a voltage divider network to sense and shutdown the chip when the supply voltage falls too low. In the circuit here it is connected directly to port pin PB2 of the Atmel microcontroller (3).
For circuit operation and to ensure the MOSFETs fully conduct when they are switched on the MAX16834 (U1) requires a supply voltage of at least 5 V. The rechargeable cells alone have insufficient output so an

additional voltage converter type LM3410 (U2) has been used to provide the stabilized voltage (+VDRIVE). The LM3410 is essentially a constant current regulator for driving LEDs up to 500 mA. At the start of this design the author's original intention was to use several LM3410s to power the LEDs but all that remains is one of these tiny chips configured as a constant voltage boost converter.

The Atmel ATtiny44 microcontroller (U3) has the job of controlling all the other components in the circuit. It generates the PWM drive signal to enable the power LEDs to be dimmed and provides the shutdown signals to U1 and U2 as necessary. The built-in A/D converter measures the battery voltage using the voltage divider network R40 and R41. To reduce current flow through the voltage divider network when the lamp is switched off the ground end of the divider is connected to port pin PB0. It can now be switched into a high impedance state to cut off current flow through the voltage divider network.
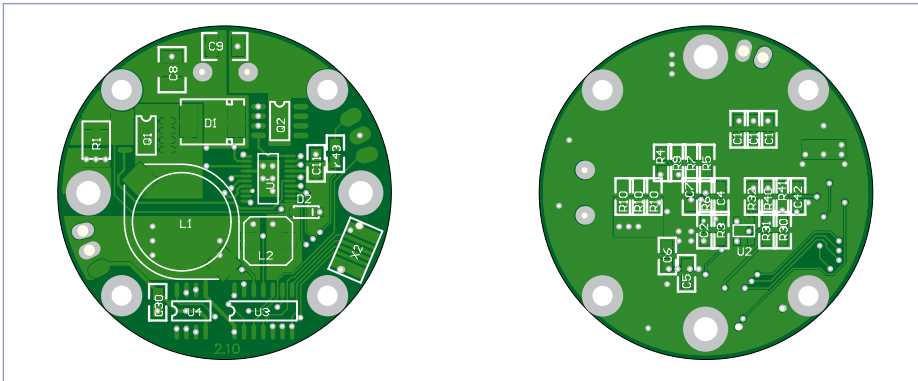
Figure 2. A small circular PCB was designed for this project.

Figure 3. The circular double sided PCB takes SMD components.

Figure 4. A view of the lamp internals.

The lamp is never turned completely off; the microcontroller, boost converter and LED driver are switched to sleep mode where current consumption drops to around 100 µA. This level is insignificant compared to the battery capacity of 8.8 Ah. Push button PB1 produces an interrupt to the controller, bringing the circuit out of sleep mode.

Power supply

The MAX1811 (U4) takes care of battery management. The chip has been specifically designed to charge Li-Ion cells from a USB port. The chip can handle a charging current of approximately 500 mA so a complete charge cycle will take up to 20 hours. it would be problematic to supply a higher current through the type of connector used and from a safety viewpoint it would also then be necessary to monitor the cell temperature to ensure that it does not overheat. The lamp is typically in use for only a few hours a day and then partially recharged overnight. The relatively long recharge time has never been a limitation.

The battery pack has been made up from four cylindrical Li-Ion cells wired in parallel. These were salvaged from a broken Notebook. If these cells were connected in series the higher voltage would mean that the step-up converter could be dispensed with in the circuit but it would then be necessary to have a higher voltage charger and provide a means to balance the cells.
With the LED current limited to 600 mA the prototype operates for almost three and a half hours continuously at maximum brightness.

Construction

A double sided round PCB has been developed for this project (**Figure 2**). The popu-

lated board can be seen **Figure 3**.
Despite improvements in high power LED efficiency over recent years it is still necessary to consider energy losses in the devices and take measures to ensure they do not overheat. The LED module (4 x CREE XR-E fitted to a 34 mm diameter PCB) is mounted on a 5 mm thick aluminum disk which is in good thermal contact with the aluminum body where the heat is dissipated to the surrounding air. The rear face of the disk also serves as a heat sink for the power MOSFETs in the circuit.

The lamp housing has parts turned from round aluminum stock and a length of 50 mm aluminum tubing. The machined body has been black anodized which turned out to be a surprisingly unproblematic procedure. The website [1] for this project contains two sketches detailing the mechanical construction of the body. **Figure 4** shows the mechanical construction of the lamp: The front end with lens, LED carrier plate, 5 mm aluminum heat sink disk, PCB, mounting spacers, plastic board and the battery cells.

The lamp housing must be watertight. The front lens is fixed into position using silicon adhesive. Both front and rear sections slide into the tube body where rubber 'O' rings provide a watertight seal. The waterproof push button with integrated LED is glued to the end of the lamp. The charging socket is a recessed SMB connector. This miniature coax connector is gold plated and is not sensitive to damp conditions. The socket can be fixed into the lamp body with waterproof adhesive.

The lamp construction has undergone strenuous road testing on a daily basis over the last two years. It has proved to be both robust and reliable and on poorly lit lanes, indispensable.

(100269)

Internet Reference

[1]   www.elektor.com/100269

INDEX OF ADVERTISERS

Robusta:
a Satellite Built by Students
Picosatellites to promote space education

By Frederic Giamarchi (France)

France is well known for its significant participation in the European space programme and for its front-line space industry, but perhaps rather less so for the training of its elite in this field. This is why in 2006 the National Centre for Space Studies (CNES) launched Expresso: the first call for projects to higher education. Montpellier 2 University applied with their Robusta project — a cubesat picosatellite carrying a scientific experiment of interest to the space community.

The Robusta (*Radiation On Bipolar University Satellite Test Application*, **Figure 1**) satellite [1] is going to be launched by the new European 'Vega' rocket in late 2011. It will carry a scientific experiment to measure deterioration in electronic components. It will be placed into an elliptical orbit between 340 km and 1,450 km at an inclination of 71°. Throughout the whole flight, it will transmit to the student ground station located on the campus of the University of Montpellier measurement data for the components being tested and the various status parameters. Subjected to the various sources of radiation, solar wind, particles trapped in the radiation belts, and cosmic rays, it will gradually fall back down towards the Earth and after two years will disintegrate on entering the atmosphere.

The Robusta satellite

This satellite has a real scientific mission: to measure the deterioration of electronic components based on bipolar transistors caused by ionizing radiation. The components chosen for testing are LM139 voltage comparators and LM124 voltage amplifiers, frequently-used components on satellites. This deterioration is quantified by measuring currents, voltages, temperature, and dose received (**Figure 2**). This dose corresponds to the absorbed radiation per unit of mass. The results will then be compared with those obtained by a ground test method devised by the researchers at Montpellier University's *IES* laboratory (*IES* stands for *Institut d'Électronique du Sud*, Southern Electronics Institute) [3][4].

The duration of the mission is fixed at two years. The data will be measured at least every 12 hours. They will then be transmitted to the Montpellier campus ground station using an amateur radio protocol and frequencies. Transmission will take place in broadcast mode, i.e. continuous every minute, whether or not the satellite is in a window of visibility for the ground station.

A crucial point for the success of the mission is power manage-

ment. The satellite will be powered by Saft Li-Ion batteries, which will be recharged by special triple-junction space solar cells with 27 % efficiency.

Internal structure

The mechanical structure, the dimensions and positioning of the electronics boards, and the aspects of the launch system via a p-pod (**Figure 3**) are being taken care of by the *GMP* (*Génie Mécanique et Productique*, Mechanical & production Engineering) and *GEII* (*Génie Électrique et Informatique Industrielle*, Electrical & Industrial Computing Engineering) sections of Nîmes Polytechnic. The power supply board and power management for the battery and solar cells sub-system is also being dealt with by Nîmes Polytechnic's *GEII* section. The controller board sub-system, which manages the commands between boards and stores the measurement data, and the microcontroller programming and test receiver elements are being operated by departments within Polytech'Montpellier. The experiment board sub-system, which includes the components under test and the dose and temperature sensors, is being designed by EEA (Electronics, Electrical Engineering, and Automation) degree and Master's students in the Faculty of Science. The radio communication board and ground station sub-systems specifically are being handled by Microwave students.

The components and materials used in this project are commercial components that are not hardened, apart from certain ones like the battery and solar cells. A rigorous radiation quality assurance process has been operated to minimize the risks associated with their exposure to radiation as far as possible. The project will be considered a success if it operates for more than a year.

Mechanical structure

The mechanical structure has been designed and machined out of a solid block so as to form a single piece (**Figure 4**). The structure is made from aluminum 6061, which has stable density in a space environment. The various elements of the satellite, solar cells, PCBs, screws, connectors, wires, etc. have all been designed and dimensioned as the project has progressed. It's taken constant interaction between the different teams for various the parts to keep the various measuring elements up to date with developments in the various corrections validated.
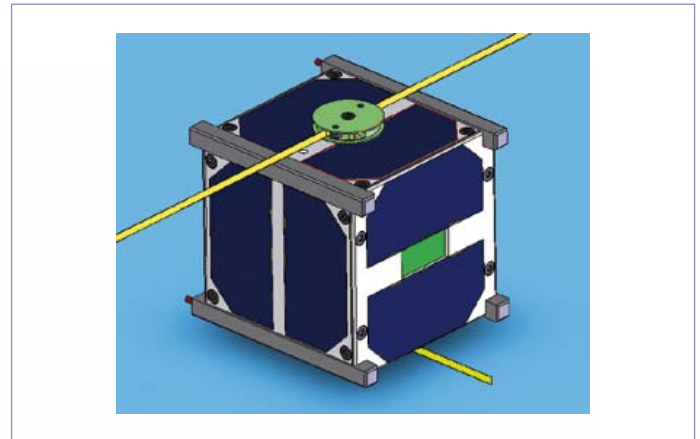

Figure 1. CAD model of the Robusta satellite.
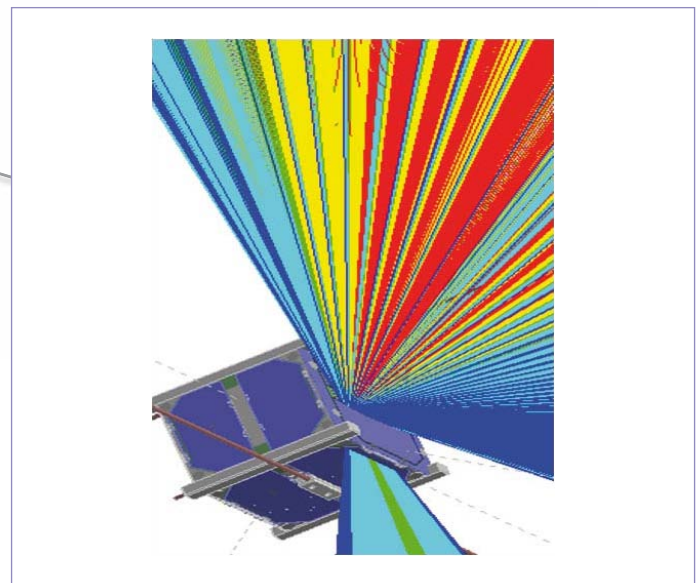(source: RobustaCom)

Figure 2. Example of modelling for calculating the dose absorbed by a component in Robusta using the FASTRAD software.
(source: RobustaCom)

Cubesat

'Cubesat' satellites are part of an educational programme put in place in 2000 by California Polytechnic University (CalPoly) [2], the aim of which is to offer students concrete experience and in-depth knowledge in relation to research and the aerospace industry. A cubesat is a satellite in the form of a 10 cm cube, weighing a maximum of 1 kg and having a maximum power of 1 W. It consists of a useful load referred to as the payload, corresponding to the on-board experiment, and a platform comprising the various electronics boards allowing control of the experiment, communication with the Earth, and power management. The whole thing represents the simply a very small equivalent of a conventional larger satellite, subject to the same stresses, with thermal shocks, extreme vibration at blast-off, radiation, and the vacuum of space.
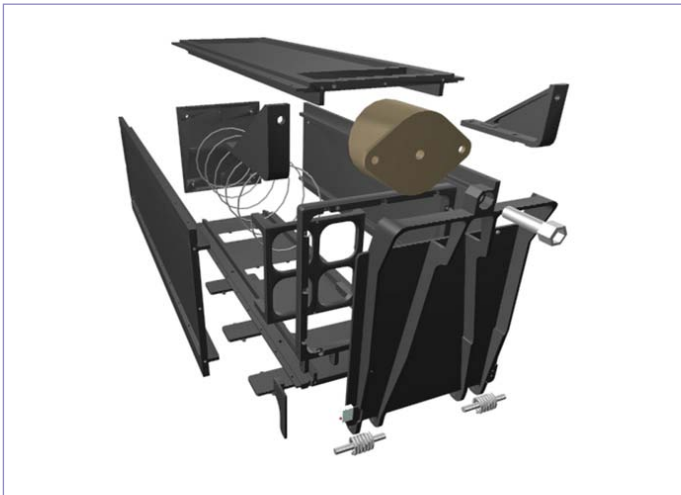
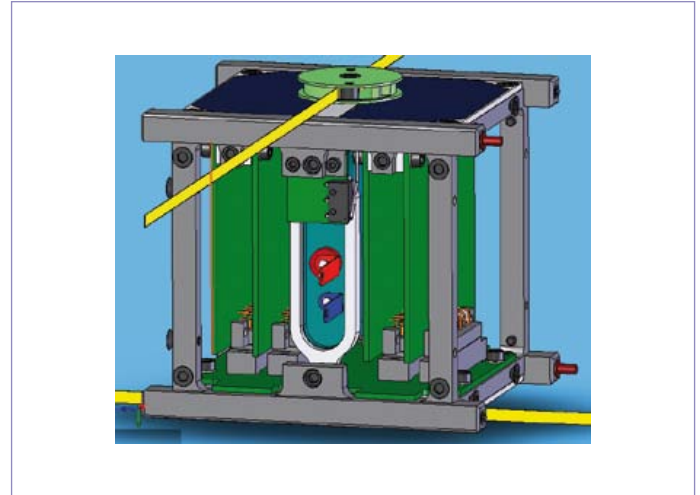Figure 3. 3D exploded model of a p-pod. (source: CubeSat Project)

Figure 4. 3D exploded model of the satellite. (source: RobustaCom)

Power board

The power board recharges the battery and distributes the different voltage rails needed for the various sub-systems. It also includes the system required for triggering deployment of the antennas once in orbit. The component ratings chosen allow a significant margin in terms of the power required. The battery charging system allows for the temperature-dependent variation in the solar cell voltage, as well as their deterioration over time. Three rails are provided: 8 V for the amplifier used to transmit the data back to Earth, 6 V for the logic circuits, and –5 V for the components under test. The six faces carrying solar cells will be subjected to the sun's rays in a random order, depending on the rotation of the satellite. It has been decided to measure the voltage and current from these six faces in order to verify proper charging of the battery and implicitly measure the satellite's rotation. An I²C bus was chosen by the students for dialoguing between the power board and the controller board.

Experiment board

The electronic circuitry for the experiment board had to be very

thoroughly designed and tested. Each of the integrated circuits under test (LM124 and LM139) includes eight elements, for which currents, voltages, temperatures, and doses have to be measured. So it was necessary to find an architecture based on analog switches driven by the microcontroller in order to multiplex the measurements made at the various pins of the devices (**Figure 5**). The students had to choose a bus that would allow managing the large number of addresses allocated to the switches, and in so doing learnt a lot about I²C and SPI buses.

Apart from the power board, the other boards each have a PIC18F4680, an ADC interface and an anti-latchup system (protecting the microcontrollers against short-circuits generated by ionizing particles).

Controller board

The brains of the satellite: its function is to organize the tasks of the other boards. It manages the dialog with the other boards and it is responsible for managing the power available. For example, it inhibits communication with the ground station while an experiment is
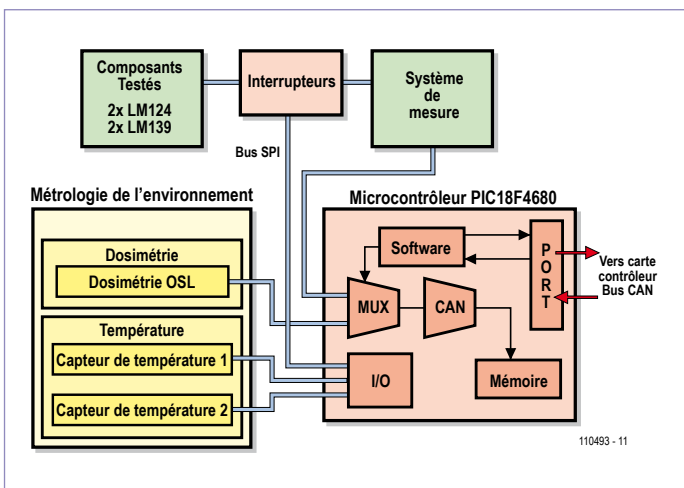

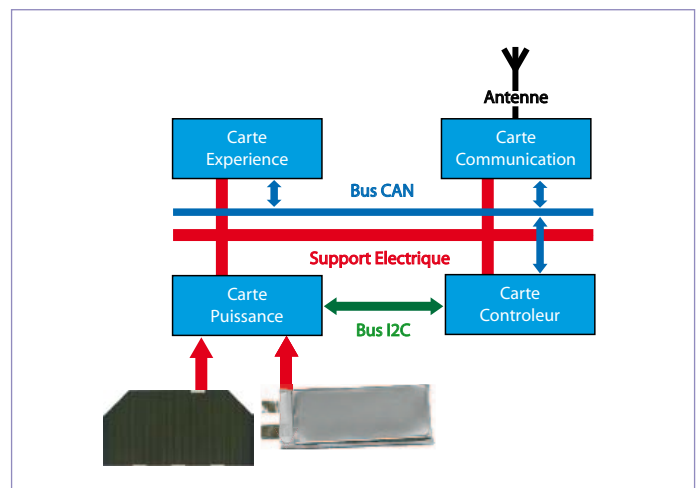Figure 5. Block diagram of the experiment board. (source: RobustaCom)

Figure 6. Interconnection of the four boards using I²C interface and CAN bus.

100 % student

The Robusta satellite and dedicated ground station have been entirely designed and produced by the students from the various courses at the Montpellier 2 University: Nîmes Polytechnic, Polytech'Montpellier engineering college, and the EEA (electronics, electrical engineering, automation) degree and master's courses in the Faculty of Science. This project in partnership with the *CNES* is also being supported by major manufacturers in the electronics sector.

tions and making the project information accessible for the media. The satellite has to be conceived, produced, and tested by the students, guided by educators who are experts in the field concerned, and always under the control of the *CNES*. The students are also responsible for the project management.

This project represents real experience of an industrial nature, but on a scale that remains accessible to students by virtue of its duration, cost, and technical level. Robusta, as a system, makes it possible

Several major educational themes can be identified within this project: the system design and associated project management, the mechanical structure, the environmental tests, and the sub-systems involving several fields within the EEA: power management, payload, controller board, radio communication board, and the associated ground station. And lastly, there is an element involving communica-

for college students from 2nd year degree to PhD level to develop sophisticated engineering prototypes and to improve their sense of communication, while discovering the world of space. They are deeply committed right from defining the mission to exploiting the measurement data, through all the phases of design, component sourcing, production of prototypes, and testing.

in progress, as these two actions are too power-hungry.
In the course of the inter-sub-system meetings, it was decided to use a CAN bus for the communication between the various sub-systems (**Figure 6**). But as part of the prototype design process, the students have been able to develop their own data exchange protocol. As there are such a large number of messages to be sent to the other boards, a Petri net has been used to avoid jams and losing messages, and to correctly handle the imposed restrictions.

direct consultation with the ground station sub-system students, in accordance with the link budget. In addition to learning a great deal about the problems specific to using radio frequencies, they were also called up to implement signal processing processes when choosing the type of modulation and demodulation. And simulation has not been overlooked, particularly for the satellite antennas, which were fully simulated using CST Microwave Studio, a special professional microwave application.

Radio board

For the radio communication sub-system, the students have paid due attention to the selection of the frequency band allocated for an application. After studying various transmission/reception architectures, they opted for a system using two separate frequencies in the radio amateur bands: 435.325 MHz for transmission to the ground station and 145.95 MHz for receiving the remote commands. The choice of components, in particular the amplifiers, was made in

Ground station

The ground station is an integral part of any space mission and is vital for it to function properly. It becomes the sole communication interface possible once the satellite is in orbit. Thus it makes it possible to receive all the experimental data and the flight parameters (telemetry), as well as to send remote commands for modifying the experimental protocol or the behavior of the satellite (for example, manage the power supply, activate or disable certain sections).

Expresso

The Toulouse Space Centre (*CST*) which comes under the *CNES* offers students an opportunity to gain concrete experience in the field of orbital systems. This is also the occasion to test out new technologies and to perform scientific experiments for the space community

at minimal cost. To support the project, the *CNES* is offering financial resources and is making available a project coordinator and experts from the *CST* for the thermal analysis, the solar cells, vibration testing, frequency use permissions, etc.

Figure 7. The Robusta ground station.
(source: RobustaCom)

Figure 8. The Vega rocket.
(source: ESA)

The Robusta ground station (**Figure 7**) is constructed around amateur radio hardware. The central element in its architecture is a transceiver that allows modulation/demodulation of AFSK signals, in the 430 MHz band for the telemetry and in the 144 MHz band for the remote commands. Two motorized antennas establish the link with the satellite. A PC manages the motorization of these antennas, as well as sending the remote commands and the reception of the telemetry. All this ground station software has either been developed internally or comes from the world of 'open source' — starting with the Ubuntu operating system on the ground station PC. This means we can adapt the software to our specific needs and upgrade it over time with no particular restrictions.

The Vega launch vehicle

Following a call for applications, the Robusta satellite was chosen to be put into orbit along with eight others during the Vega rocket's (**Figure 8**) qualification flight. The Vega project should make it possible to put small satellites – between 300 and 2,000 kg — into low or polar orbits. This will be a first for this launch vehicle which will blast off from the Kourou space base in French Guiana in late 2011.

In all, nine cubesats will be released from the launcher at the same time as the main payload, a scientific satellite called LARES System along with the educational mini-satellite ALMASat.

(110493)
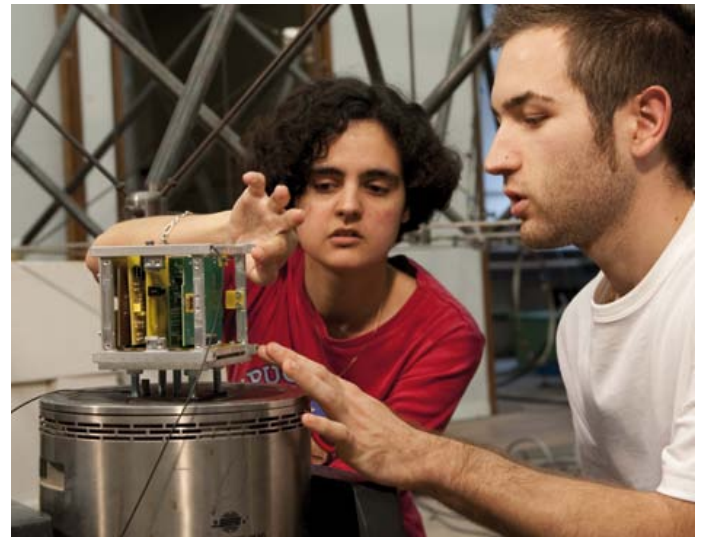
Internet Links & References

[1]   Project Robusta: www.ies.univ-montp2.fr/robusta/

[2]   Cubesat by California Polytechnic State University:
      http://polysat.calpoly.edu/

[3]   J. Boch, "Estimation of Low Dose Rate Degradation on Bipolar Linear Integrated Circuits Using Switching Experiments", IEEE Trans. Nuclear Science, vol. 52, pp. 2626–2621, December 2005.

[4]   J. R. Vaille, F. Ravotti, P. Garcia, M. Glaser, S. Matias, K. Idri, J. Boch, E. Lorfevre, P. J. McNulty, F. Saigne, L. Dusseau, "Online dosimetry based on optically stimulated luminescence materials" IEEE Trans. on Nuclear Science, vol. 52, Issue 6, December 2005 pp. 2578–2582.

## The space adventure continues with Friends and Ristretto

Robusta, Expresso, Friends, Ristretto — no, these aren't terms from some galactic coffee bar, but rather the names of student projects for conquering space. Small student satellites of increasing size — this is what Montpellier 2 University is offering us, in collaboration with the *CNES*, the ESA, and the Baumann University in Moscow.

To meet these new challenges, Montpellier 2 University will shortly be setting up a university space center named SOLARIUM (*Systèmes Orbitaux Liés aux Activités de Recherche Interdisciplinaires de l'Université Montpellier 2*), with support from the Van Allen foundation — the first in France to exploit the potential of small satellites

USB Mouse using R8C/13 Starter Kit

Helmut Posselt (Germany)

The 'Tom Thumb' R8C/13 starter kit described in UK-Elektor in February 2006 has proved popular with our readers and has found its way into many applications, as a glance at the Forum on our website will show. The original idea behind the project presented here was to make a friction-free device for measuring linear movement using the R8C/13 and an optical mouse. The advantage of the optical mouse over its mechanical brother is that it has no moving parts that inevitably get dirty and jam.
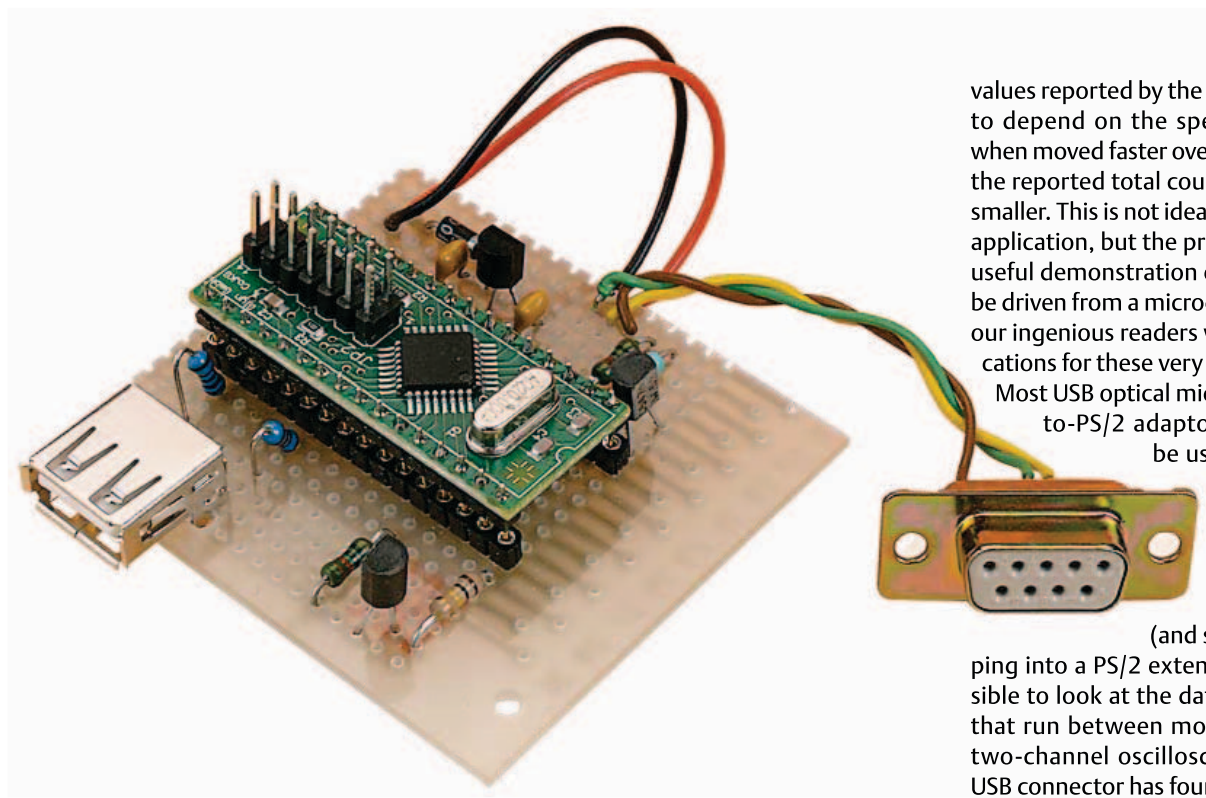
values reported by the optical mouse seem to depend on the speed of the motion: when moved faster over the same distance, the reported total count was considerably smaller. This is not ideal for a measurement application, but the project still provides a useful demonstration of how a mouse can be driven from a microcontroller. No doubt our ingenious readers will find other applications for these very low-cost sensors.

Most USB optical mice come with a USB-to-PS/2 adaptor to allow them to be used on older PCs via the PS/2 connector. The mouse uses four pins on the connector, as shown in **Figure 1** (and see also [1]). By tapping into a PS/2 extension cable it is possible to look at the data and clock signals that run between mouse and PC using a two-channel oscilloscope. Similarly, the USB connector has four pins (**Figure 2**) and again we can tap into an extension cable to monitor the signals. The surprise is that the USB signals are the same as the PS/2 signals!

The system consists of a C program running on the R8C and a short terminal program in Visual Basic to display the results on the PC. Early tests showed that the displacement
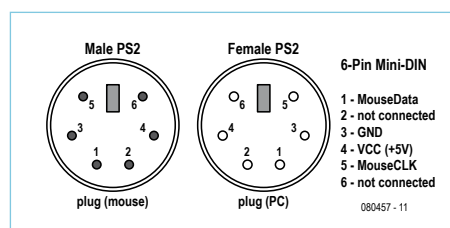
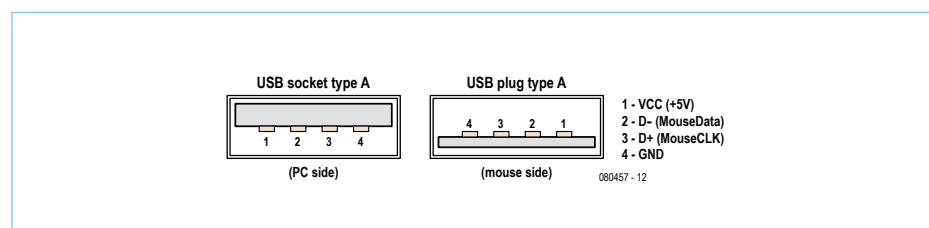Figure 2. Pinout of the PS2-to-mouse connection.

Figure 3. Pinout of a USB mouse with Type-A plug and socket.

The USB-to-PS/2 adaptor is in fact entirely passive and simply connects the USB signal wires through to the PS/2 connector.

For our experiments with mouse and microcontroller we simply need to obtain a suitable PS/2 or USB socket and connect the two control signals to spare port pins, for example via 1 kΩ resistors on the R8C/13. It is also necessary to connect GND on the mouse connector to GND on the R8C/13, and to supply the mouse with +5 V, which can also be done via the microcontroller board (**Figure 3**). For test purposes we connect the R8C/13 board to a PC using its RS-232 interface (RXD1 and TXD1). The short terminal program, written in Visual Basic 5, along with 'port.dll' [4], allows command codes to be sent to the R8C/13 at the click of a button; the R8C/13 in turn carries out the desired command. Where necessary, additional data values required for its execution are appended to the command code (for example, in the case of SetSampleRate).

The returned values are sent back to the PC for validation and display. This makes it easy to check that the mouse and the R8C/13 program are running correctly. **Figure 4** shows an example of received data when the mouse is 'hot plugged', and at the top of the figure is an oscilloscope trace of the data and clock signals.

Communication between mouse and R8C/13 uses a bidirectional synchronous serial protocol [1], one byte at a time. Each byte is sent as a string of 11 bits, consisting of one start bit (always logic 0), eight data bits (sent LSB first), one parity bit (logic 1 if the number of '1' bits is even, logic 0 otherwise), and one stop bit (always logic 1). When transferring data from the host to the mouse there is an additional acknowledge bit.

The clock signal is always generated by the mouse. However, the R8C/13 can hold the clock signal Low in order to interrupt the mouse. The R8C/13 reads data bits from the mouse on the falling edge of the clock signal, and the mouse reads data bits on the rising edge of the clock.
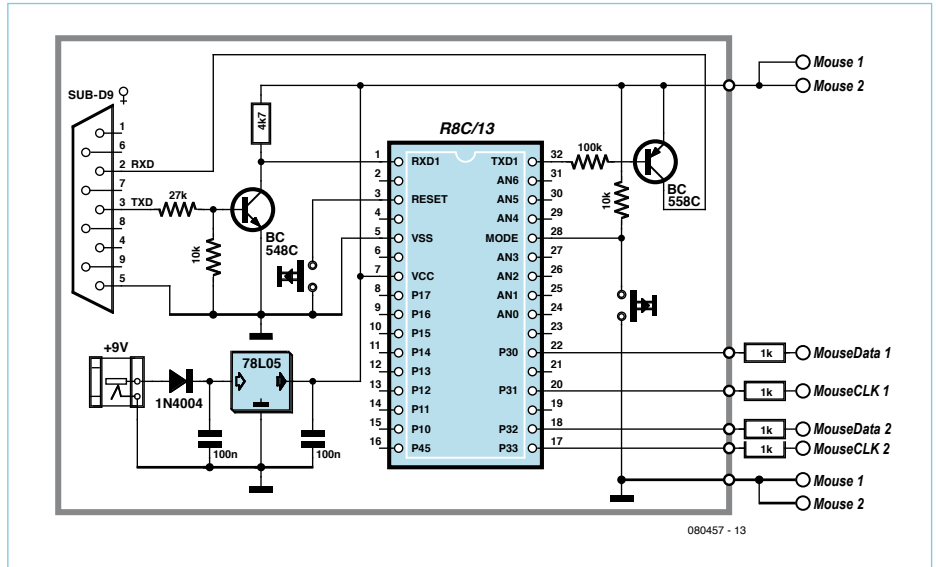

Figure 4. The minimal system with the R8C/13 carrier board complemented with a mouse connection.

At the beginning of a mouse command (see [2] for a more complete description of the commands) the R8C/13 first pulls the clock signal Low for 100 µs, which interrupts any communication from the mouse. The R8C/13 must then take the data signal Low and the clock signal High, forming a 'request to send' instruction to the mouse. The mouse is now permitted to start generating clock pulses.

Figure 5 shows the status information that a Status Request command elicits, and **Figure 6** shows the movement and button information. An example of command execution is shown in **Figure 7**, where the command 'ReadStreamData' has been sent: one of the movement data packets (containing status, X-movement and Y-movement information: compare with Figure 6) is shown in the figure. Further examples can be found in a PDF file at [5], where software files (including source code) for this project are available for download.

If a standard PS/2 mouse is used, movements in the X and Y axes and the state

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	const. 0	Mode	Enable	Scaling	const. 0	centre button	right button	left button
Byte 2	resolution							
Byte 3	data rate							

080457 - 14

Figure 5. Status register structure and content.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Y overflow	X overflow	Y sign bit	X sign bit	const. 1	centre button	right button	left button
Byte 2	X movement							
Byte 3	Y movement							

080457 - 15

Figure 6. Data on movement and switch status.

of the left, right and middle buttons are reported. With certain non-standard PS/2 mice it is also possible to obtain the status of additional buttons and of a scroll wheel. In this project we ignore acknowledge bytes and the overflow bits in the mouse messages, and we do not check parity bits. **Figure 8** shows the partial support for PS/2 keyboards. In the log the first line shows the keyboard being reset followed by a 'GetDeviceID' command being issued. The 'A' key on the keyboard is then pressed and released.

The R8C/13 program
In 'main' the program runs in an infinite loop. On each pass through the loop the serial input buffer is checked for data. If a byte has been received from the PC a mouse command is issued. First 'RequestToSendFlag' is set by a call to 'RequestToSend'. This function holds the clock signal Low for at least 100 µs by disabling interrupts and configuring the port pin as an output. Then, after 100 µs, the data signal is taken Low and the clock signal taken High: interrupts are enabled and the port is configured as an input. Finally 'RequestToSendFlag' is cleared.

Negative-going clock edges are detected via INT2. INT3 is used to detect the level of the data signal from the mouse when receiving data from it and to set the data signal to the appropriate level when transmitting. This behavior is controlled by the subroutine 'mClockLow(i)', which can support two mice or one mouse and one keyboard [3]. The byte counter writes the number of received data bytes into the second byte of the transmit buffer. The first byte in this buffer indicates via a value of 1 or 2 that valid mouse data are available.
If a negative-going edge is detected and no mouse command is being sent, then either a 'hot plug' event has occurred (Figure 6) or a data packet is being received in stream

Figure 7. ReadStreamData from a mouse captured on an oscilloscope (above) and in the VB Terminal program (below).

Figure 8. A PS2 keyboard is also partially supported.

mode (Figure 7).
Flag 'ucModeFlag' is used to control how the transmit buffer is filled with mouse data and with position data accumulated since the last external reset of the X and Y movement integrator registers.

Variable 'timeout[i]' is set to a positive value on each negative-going clock edge, which is taken as an indicator of mouse activity. TimerY decrements this value once per millisecond, and every 50 ms TimerY checks whether the value has reached zero. If 'timeout[i]' has reached zero and the first

byte is the transmit buffer is 1 or 2, then there is a data packet to send; otherwise the byte '0' is sent.

The Visual Basic Terminal Program
This program provides a way to visualize mouse activity. It initializes the mouse and sets its operating mode. The most important part of the program is the form which contains buttons for sending control commands and text fields for displaying data from the mouse.
The program defines the following modes: initialization, test-remote, test-stream, remote and stream. The test modes allow manual retrieval of individual mouse data packets after a mouse state change; otherwise position requests are made periodically using Timer1.
Timer1 checks every 50 ms whether a data packet is available (by examining the first byte in buffers 1 and 2). If a packet is available the second byte in the buffer determines how many bytes remain to be received. The current mode is then used to control the display of data in the text boxes.
The 'Mouse ID' button issues a series of consecutive commands which are used to identify the type of mouse connected, for example whether the device is a three-button scroll mouse with ID code 3.

(080457)

Sources and Internet links

[1] www.computer-engineering.org/ps2protocol

[2] www.computer-engineering.org/ps2mouse

[3] www.computer-engineering.org/ps2keyboard

[4] www.b-kainka.de/port.zip

[5] www.elektor.com/080457

Hexadoku
Puzzle with an electronics touch

If you are into puzzle solving, why not join the crowd of Elektor readers who solve the secret of Hexadoku every month, or at least have a crack at it! We're sure you'll be thrilled. "Simply" enter the right numbers in the puzzle below. Next, send the ones in the gray boxes to us and you automatically enter the prize draw for one of four Elektor Shop vouchers. Have fun!

The instructions for this puzzle are straightforward. Fully geared to electronics fans and programmers, the Hexadoku puzzle employs the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation. Correct entries received enter a draw for a main prize and three lesser prizes. All you need to do is send us the numbers in the gray boxes.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for one Elektor Shop voucher worth $ 140.00* and three Elektor Shop Vouchers worth $ 70.00* each, which should encourage all Elektor readers to participate.

*Subject to exchange rate.

Participate!

Before January 1, 2012, send your solution (the numbers in the gray boxes) by email, fax or post to

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

Fax 860 8751-0411   Email: hexadoku@elektor.com

Prize winners

The solution of the October 2011 Hexadoku is: **D0837**.
The Elektor $140.00 voucher has been awarded to Olavi Parkka (Finland).
The Elektor $70.00 vouchers have been awarded to Susanne Müller-Furrer (Switzerland),
Robert Amandine (France) and Thierry Notot (France).
Congratulations everyone!

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

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

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

Product Development II

Gerard Fonte (USA)

Back in June 2010 I talked about product development from a general and theoretical point of view. This time we'll look at some of the more practical aspects of taking an idea and making it real.

Inside or Outside

The first decision you will have to make is if you plan on fabricating your product by yourself or paying someone else to do it. The first instinct is to do it yourself and save all that money. And while it is true that you will save a lot of money at the start, you may not save money in the long term. Here's why.

Suppose your product has a parts cost of $30 and takes one hour to assemble (which is exceptionally fast). If you do it yourself and sell your product for $100, you've made $70 profit per unit. If you take it to a production house they might charge you $25 for the parts, $40 for the labor and a few hundred dollars for a "set-up" fee. Your profit drops to less than $35/unit.

But, this is because you didn't charge for your own labor or overhead initially. How much is your time worth? If you want to make $50,000/year for your labor (like your day-job) then you have to charge $25/hour, at least. That doesn't cover health insurance, social security payments, holidays, retirement plan, vacation and other fringe benefits that are included in your regular day-job. Do you want to spend all your time ordering parts, assembling, testing and shipping? What about the time needed to market your product, or develop new products, or build your company, or keep up-to-date on engineering advances? Since you probably don't want to spend all your time soldering, you would have to hire someone. You can see that paying a technician $20/hour ($40,000/year) plus all those extras, plus work-space, plus tools, plus training, can rapidly exceed the $40/hour that the production company charges. And they have lots of experience in producing quality products.

So, from the start, you basically have to decide if you want to develop your own production facilities. It should be noted that very successful companies buy their "start-up" production companies with a fair degree of regularity. This brings the production back into the company and provides an additional stream of revenue.

Paper Work

Whether you build your product in-house or hire a contractor, you must have a proper set of blueprints. It cannot be stressed too much that a complete set of technical drawings is the most important aspect of any product. These describe precisely what the product is and how to build it. A sample product is nice and useful, but it is in no way sufficient. A production house requires paperwork and so will you as soon as you hire someone (or your computer crashes, or you find a production bug, etc.).

The documentation should be contained in a single bound notebook of some sort (or integrated computer document). You don't want loose pages falling out. There should be several copies in different locations. One copy should always be with production. Putting one in a safety deposit box is a good idea (Remember, your product is your source of income). If you number them it's easy to be sure that they are ALL updated as needed. There's nothing worse than having two different drawings for the same thing.

The drawing package should include: schematic diagram, technical specifications, operating instructions, parts list (with special parts noted), front panel drawings, assembly drawings, printed circuit layouts (for all layers), assembly procedures, testing procedures, theory of operation, engineering notes, software listings, software notes and anything else needed to produce the product. Any engineer should be able to read the drawing package and build your product without any questions.

There are two options for updating the drawing packages. The first is to keep the old version for traceability. This is useful, but can make the package too big and can sometimes lead to confusion if the old and new drawings are not clearly identified. The other way is to replace the old drawings with the new ones. This has problems if an old product needs to be serviced. Note that there must always be a copy of every drawing revision on file. Discarding "old drawings" is a bad idea. By the way, never sell your prototypes! They're generally of poor construction quality and they are invaluable for troubleshooting, if you have a production problem.

Profit Margin

Pricing your product is a critical decision. If it's too expensive, it won't sell. If it's too cheap, you lose money. In the example above, the production house unit provided a $35 profit. If you want to make $35,000 in profit a year you will need to sell 1000 units. Pretty straightforward. Except that we didn't subtract other costs.

For example there is marketing and advertising (they are not the same). Often this is about 10% of your basic production cost or about $6.50/unit or $6500/year. There is overhead (taxes, heat, rent, insurance, gasoline, travel, etc.). If this is $300/month then this is $3600/year. Just these two items subtract about $10 from your per-unit profit. So instead of selling 1000 units, we have to sell 1400 units/year to get out $35,000 in profit. That's a 40% increase in sales over what we initially expected. Alternatively, we would only see $25,000 profit for 1000 units sold.

We can now see that the original selling price was too low. Our profit is less than $25/unit when we initially thought we were going to get $70 (if we made them ourselves). The rule of thumb is that your selling price has to be at least twice the basic production cost. So if it costs $25 for parts and $40 for labor then your selling price must be at least $130. This increases our profit from $25 to $55/unit or $55,000/year (1000 units). Now, that's not bad.

The bottom line is that your bottom line may not be as simple as you first thought.

(110731)

RCA Cosmac Development System
hELLO wORLD from Embedrock City

By Jan Buiting (Elektor UK & US Editorial)

As far as microprocessors are concerned my roots are in the early 1980s when all sorts of 'hobby systems' were around based on competing devices like the Z80, 6502 and 8085. I've always hated electronics that gets hot while doing nothing useful so I left these NMOS current hungry micros, their fan clubs and BBSs well alone and went for a less popular 8-bit number cruncher called CDP1802 originally designed and produced by RCA, the company behind the famous CD4000 series of logic ICs. The CMOS (hence acutely energy friendly) 1802 and its family of 'Cosmac' peripheral ICs did really well in the USA after a cute DIY system called Cosmac ELF was published in *Popular Electronics* way back in 1976 (it was roughly the time when Jobs & The Woz tinkered with 6502s in their Palo Alto garage). A more sophisticated European variant called 'Cosmicos' was developed about four years later and that was to become my personal 'platform' (see "CDP1802 — the first micro in space", Elektor October 2006).

In 1980, as a student, if you could lay your hands on an RCA Cosmac data book, you were King. I managed to permanently borrow one from a kind soul at Vekano, a former RCA distributor in Holland. Probably to fill the book and keep the marketing people happy, the last 40 or so pages of the book showed some of RCA's software tools, hardware, programmers, compilers (like BASIC and PL/M) and Cosmac development systems. Although my own DIY Cosmicos system was pretty well equipped (48 KByte RAM and all that), 'green' (under 200 mA @ 5 V) and fast too (3.58 MHz), I marvelled and drooled over the specs and (very poor) photograph promoting RCA's top-line product: the monumental Cosmac Development System IV CDP18S008. No price was given — I guess you had to telephone.

Depending on your hobby or interests, it may take just 30 years before you can actually buy what you dreamt of as a youngster or student. Two forces work to your advantage, slowly but surely: (1) you make money instead of wasting it and (2) the price of the 'desideratum/a' drops to the level of techno junk no one wants. I've

always cherished my Cosmicos CDP1802 system and all its cards and peripherals, and a few years ago while browsing the ELF pages at Yahoogroups I could not believe my eyes! A posting from a fellow countryman politely asking a mostly American audience on the forum if anyone would like a complete Cosmac IV system. All respectfully declined because of the colossal cost of getting the kit shipped to the US, where it had come from in 1981. To cut a long story short, I collected the complete system, paid a symbolic price and drove it home. A load of hardware actually developed with the system was also included, as well as documentation in binders and software on 8-inch floppy disks. It was the first time I had to adjust the headlights on my car to prevent dazzling oncoming traffic. I never realized the system I had seen in a book 30 years ago was so bulky and heavy. But Home & Mother, what a find!

The bright blue and off-white unit labeled 'Cosmac IV' is basically a CRT dumb terminal talking internally to a CDP1802 system. It weighs only 17 kgs (38 lbs). Believe it or not but the 'terminal' is itself a CDP1802 video system sitting between a keyboard and a 12-inch CRT. The actual development system is a separate backplane onto which RCA CDP18Sxx 'micromodules' are plugged like CPU (CDP1802), ROM, RAM, I/O, FDISK, etc. The software you want to develop for a custom application is fully written, tested and debugged using the 'card nest', until (you think) hex code is ready to safely burn firmware (E)PROMs like the 2708 and 2716. The (E) PROM programmer is accessible under a hinged panel. The panel opens and closes by pressing it. People find it funny. On the rear of the terminal casing there's connectors for AC power, disk, printer, CRT EIA, SYS, EIA, MOPS EIA, spare #1 and spare #2.

The Cosmac IV terminal unit has a nostalgic bright green CRT display that's just perfect to read although at 24 lines of 80 characters it's claustrophobic compared to today's LCD monitors. Also, the Sperry made 73-key ASCII keyboard is a delight to listen to when typed upon at speed, the keys being of the Hall effect type. Using the full-screen editor (FSE) to write your code (in assembler, of course) you hardly miss the ease and comfort of a mouse, menus, icons or the constant distraction of the Internet!

IV (CDP18S008) (ca. 1978)
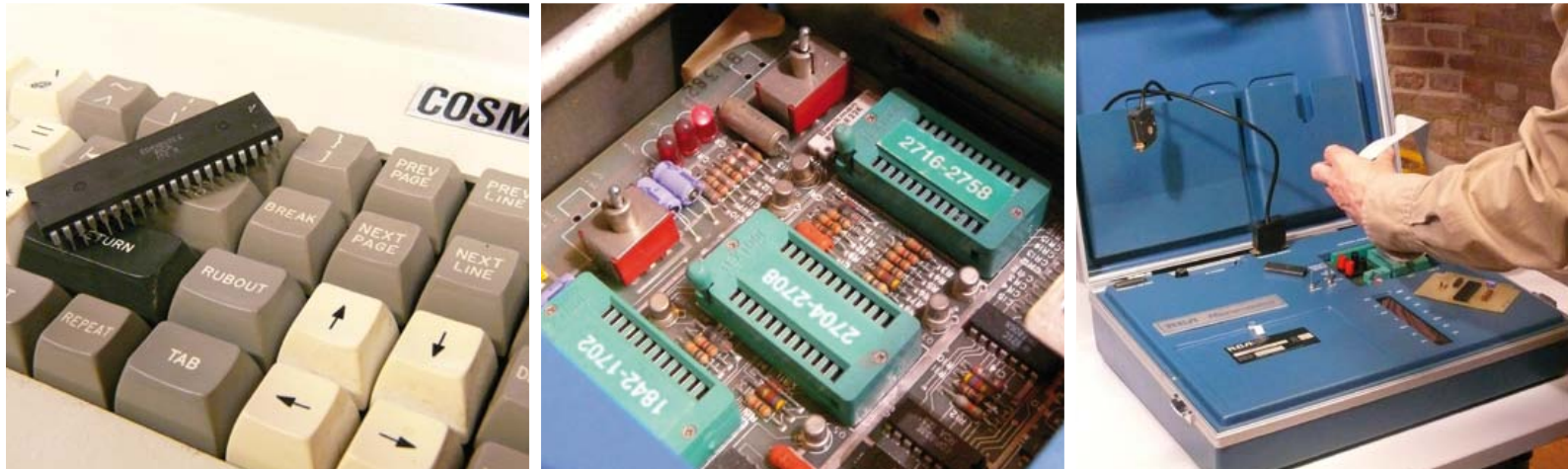
If you thought the Cosmac IV blue & white CRT terminal is "expensive & comprehensive" and you can't wait to start writing code and burn ROMs, wait! There are two more 'components' that go into the CDP18S008 system.

First, there's a dual disk drive for 8-inch floppy ("flexible") disks with a gross capacity of 256 KBytes (0.025 GBytes) each. Made by Pertec (Chatsworth & Irvine, CA) and the case cover sprayed 'RCA blue' according to a leaflet I found, this monster called Model 3712 weighs 34 kgs (75 lbs). The same drive units but in a different color were supplied to Altair for their 8080 based systems, and very likely to other minicomputer manufacturers. The unit produces a horrendous amount of noise due to a huge extractor fan and the two disk drive motors spinning all the time (probably to keep disk access times within limits). A loud 'clack' is heard whenever a floppy disk gets selected or released. The hum of the heads racing across the disk surface would make a nice ringtone. Inside the case, a control board roughly the size of two iMacs is seen, stuffed with 150-odd ICs — mostly plain TTL running at 5 V. Noisy, hot and slow as it may be, the dual disk drive unit turned out to work reliably after 30 years — all 8-inch disks I got from the previous owner could be read without problems and I am now considering starting an 8-inch-to-USB-stick On Demand Conversion Service.

The third component of the system is called 'Micromonitor' (CDP18S030), where 'micro' definitely refers to 'microcontroller', not size. It's an aluminum suitcase, again sprayed RCA blue, containing not much more than LEDs, switches and ZIF sockets. The idea was to migrate the CPU from the customer application to the Micromonitor (via a length of 40way ribbon cable) and then sort of single-step the firmware to watch what the CPU lines were doing! So, the blue suitcase not unfit for a traveling salesman appearing on *The Jetsons* was intended to help debug CDP1802 applications "in-system, in real time, on site"! But just how? I really can't figure out because you soon need to view hex code and possibly enter it in a

comfortable way. A later version (CDP18S030A) had a detachable display/keyboard unit that looked like a 1970s pocket calculator. The only mass-produced CDP1802 application circuits that seem to have survived are a US made traffic lights system occasionally offered on Ebay, and a 1996 (!) Nokia UHF in-car radiotelephone that got hacked and converted to amateur radio use. Oops, I should not forget a home / game computer called COMX35.

Today, a small group of people still enjoy working with the CDP1802 'Cosmac' CPU; they can be found within the ELF communities on the web. Personally, I am using the Cosmac IV system occasionally to tweak the software of my DIY hothouse climate and irrigation control based on a good old CDP1802. I cheerfully use things like PL/M, CDOS, UT5, MOPS, BASIC1, and ASM8. The bulky disk drives and the Micromonitor are no longer used, the former being emulated by two SMD static RAMs with battery backup! For sure I can appreciate 30 years of progress and miniaturization we all achieved in microcontroller systems and components. True, all 70 kgs (155 lbs) worth of CDP18S008 should easily fit in, say, a single Xilinx 'Spartan' FPGA. I know, my Cosmac IV system really belongs in a computer museum. On the Internet, a 1976 price of $70,000 was mentioned somewhere for a full blown CDP18S008 system like the one described here. I have no way of verifying. All I know is that the previous owner never quite managed to recoup the cost of his system through developing, selling and supporting highly specialized applications commissioned by clients including government institutions. It's a sobering thought that today microcontroller development systems are available at give-away prices from the manufacturers.

If any Retronics reader out there has any original RCA Cosmac CDP-18Sxx disks containing software tools or higher level programming languages, please let me know. The same for a CDP18S021 Microterminal running UT5.

(110528)

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

products and services directory

Going Strong

A world of electronics from a single shop!

NEW!

**Complete
your collection now at
www.elektor.com/books**

Creative solutions for all areas of electronics

311 Circuits

311 Circuits is the twelfth volume in Elektor's renowned 30x series. These Summer Circuits compilation books have been bestsellers for many years. This brand new book contains circuits, design ideas, tips and tricks from all areas of electronics: audio & video, computers & microcontrollers, radio, hobby & modelling, home & garden, power supplies & batteries, test & measurement, software, not forgetting a section 'miscellaneous' for everything that doesn't fit in one of the other categories. 311 Circuits offers many complete solutions as well as useful starting points for your own projects. Both categories and anything in between represent a veritable fountain of inspiration for cultivating your own ideas and learning about electronics. This book deserves a place not far from the workbench!

420 pages • ISBN 978-1-907920-08-0 • $47.60
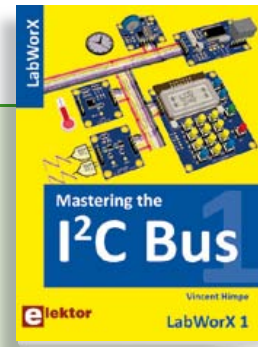
LabWorX: Straight from the Lab to your Brain

Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C Bus and its applications. Besides the Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I²C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9 • $47.60
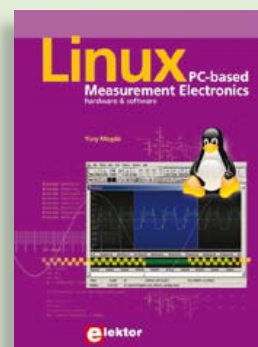
A highly-practical guide

Linux – PC -based Measurement Electronics

If you want to learn how to quickly build Linux-based applications able to collect, process and display data on a PC from various analog and digital sensors, how to control circuitry attached to a computer, then even how to pass data via a network or control your embedded system wirelessly and more – then this is the book for you! The book covers both hardware and software aspects of designing typical embedded systems using schematics, code listings and full descriptions.

264 pages • ISBN 978-1-907920-03-5 • $47.60

CD/DVD-ROMs

More than 25 projects based on the Elektor ATM18 board

CD ATM18 Collection

This CD-ROM contains all articles from the popular ATM18-CC2 series published in Elektor magazine. From RFID Reader and Bluetooth linking right up to a chess computer! Project software and PCB layouts in PDF format are included. What's more, this CD also contains a Bascom AVR programming course and helpful supplementary documentation.

ISBN 978-0-905705-92-7 • $39.60

All articles in Elektor Volume 2010

DVD Elektor 2010

This DVD-ROM contains all editorial articles published in Volume 2010 of the English, American, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy of PCB layouts at printer resolution, adapt PCB layouts using your favourite graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-267-9 • $37.90

New!

Improved Radiation Meter

(November 2011)

All that's required to measure radiation is a simple PIN photodiode and a suitable preamplifier circuit. Elektor presents an optimised preamplifier and a microcontroller-based counter. The microcontroller takes care of measuring time and pulse rate, displaying the result in counts per minute. This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive samples.

Kit of parts incl. display and programmed controller

Art.# 110538-71 • $57.30

Bestseller!

FT232R USB/ Serial Bridge/BOB

(September 2011)

You'll be surprised first and foremost by the size of this USB/serial converter — no larger than the moulded plug on a USB cable! And you're also bound to appreciate that fact that it's practical, quick to implement, reusable, and multi-platform — and yet for all that, not too expensive! Maybe you don't think much of the various commercially-available FT232R-based modules. Too expensive, too bulky, badly designed, … That's why this project got designed in the form of a breakout board (BOB).

PCB, assembled and tested

**Art.# 110553-91 • $20.90**
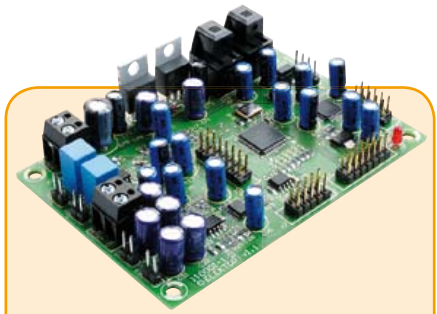
Audio DSP Course

(September 2011)

This DSP board is the platform for the applications described in our Audio DSP Course. It is also intended to enable you to develop your own initial digital audio signal processing applications. The DSP board can be used stand-alone as is, and even though it is an ideal learning platform, with its 24-bit signal processing capability for sampling rates up to 192 kHz and its high-performance interfaces, it is also suitable for applications with very stringent quality requirements for both signal to noise ratio and DSP computing power.

Populated and tested DSP board

Art.# 110001-91 • $186.70
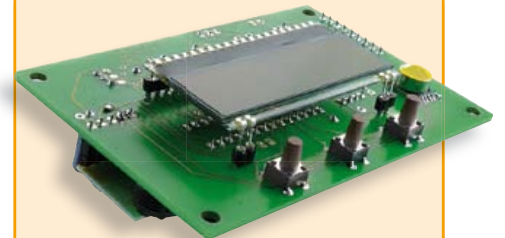
USB Long-Term Weather Logger

(September 2011)

This stand-alone data logger displays pressure, temperature and humidity readings generated by I²C bus sensors on an LCD panel, and can run for six to eight weeks on three AA batteries. The stored readings can be read out over USB and plotted on a PC using gnuplot. Digital sensor modules keep the hardware simple and no calibration is required.

Kit of parts incl. PCB, controller, humidity sensor and air pressure sensor modules

Art.# 100888-73 • $50.20

Kits & Modules

Bestsellers

Books

1. **Controller Area Network Projects**
   ISBN 978-1-907920-04-2........................ $47.60

2. **311 Circuits**
   ISBN 978-1-907920-08-0........................ $47.60

3. Design your own
   PC Voice Control System
   ISBN 978-1-907920-07-3........................ $47.60

4. **Mastering the I²C Bus**
   ISBN 978-0-905705-98-9........................ $47.60

5. **Linux – PC-based measurement electronics**
   ISBN 978-1-907920-03-5........................ $47.60

CD/DVD-ROMs

1. **CD 1001 Circuits**
   ISBN 978-1-907920-06-6 .... $55.70

2. **DVD Elektor 1990 through 1999**
   ISBN 978-0-905705-76-7 .... $111.30

3. **DVD Elektor 2010**
   ISBN 978-90-5381-267-9........................ $37.90

4. **CD Elektor's Components Database 6**
   ISBN 978-90-5381-258-7 .... $40.20

5. **CD ATM18 Collection**
   ISBN 978-0-905705-92-7........................ $39.60

Kits & Modules

1. **FT232R USB/Serial Bridge/BOB**
   Art. # 110553-91.................................... $20.90

2. **USB Long-Term Weather Logger**
   Art. # 100888-73.................................... $50.20

3. **Improved Radiation Meter**
   Art. # 110538-71.................................... $57.30

4. **Audio DSP Board**
   Art. # 110001-91.................................... $186.70

5. **Pico C Meter**
   Art. # 100823-71.................................... $118.40

Order quickly and securely through

www.elektor.com/shop

or use the Order Form near the end

of the magazine!

elektor

AC Powerline Frequency Meter

The frequency of the AC grid voltage is nominally 50 Hz or 60 Hz. The actual value varies a little, depending on energy supply and consumption. With an accurate measurement of the frequency you can determine what's happening on your local power grid. This handy 'frequency magnifier' can detect even the smallest changes due to peak load instants.

Andropod

Android devices are ideally suited to use in conjunction with embedded electronics. For little money they offer access to a display, computing, various interfaces and sensors. The big problem however is to pinpoint the connectivity between the Android device and the hardware connected to it. The Elektor Andropod was developed specifically with that in mind: it's an Android USB interface with host functionality.
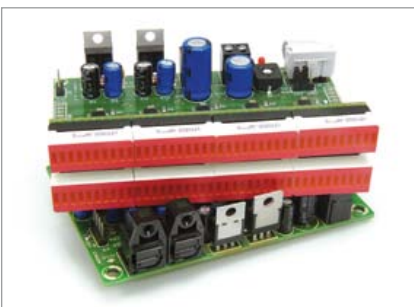
Digital VU Meter

In Part 7 of our DSP course, the DSP board gets combined with an LED VU meter. This setup enables the levels of two digital audio signals to be visualized with great accuracy. The VU meter employs special LED driver ICs from Texas Instruments and has a display made from 2 x 40 LEDs. If you think that's not enough, you can simply connect multiple boards in series.

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

Elektor UK/European January 2012 edition: on sale December 15, 2011.   *Elektor USA January 2012 edition: published December 12, 2011.*

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

Elektor on the web

All magazine articles back to volume 2000 are available individually in pdf format against e-credits. Article summaries and component lists (if applicable) can be instantly viewed to help you positively identify an article. Article related items and resources are also shown, including software downloads, hyperlinks, circuit boards, programmed ICs and corrections and updates if applicable.

In the Elektor Shop you'll find all other products sold by the publishers, like CD-ROMs, DVDs, kits, modules, equipment, tools and books. A powerful search function allows you to search for items and references across the entire website.

Also on the Elektor website:

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

Description	Price each	Qty.	Total	Order Code
311 Circuits NEW	$47.60			
Design your own PC Voice Control System NEW	$47.60			
Controller Area Network Projects	$47.60			
LabWorX – Mastering the I²C Bus	$47.60			
Linux – PC-based Measurement Electronics	$47.60			
CD 1001 Circuits	$55.70			
	Sub-total			
	Shipping & Handling $ *(Airmail from Europe)*		5.00	
	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/Company

Street & No.

City/State/Zip   *Country*

E-mail   *Tel*

EL12   *Date* – –   *Signature*

METHOD OF PAYMENT

Remit in US $ drawn on a US bank. Please see next page for ordering information.

☐ **Check or Money Order**

☐ **Mastercard**

☐ **VISA**

☐ **Discover**

☐ **American Express**

Expiry date: _____

Verification code: _____

Send order form to:

**Elektor US
PO Box 180
Vernon, CT 06066
USA**

Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com
www.elektor.com/shop

elektor

Yes, I want to subscribe to Elektor US for 1 year*

I would like:

☐ **Standard Subscription for $39.95 (11 issues)**

☐ **Plus Subscription for $59.95**
(11 issues + the Elektor Volume 2011 DVD-ROM + exclusive access to www.elektor-plus-usa.com)**

Name/Company

Street & No.

City/State/Zip   *Country*

E-mail   *Tel*

EL12   *Date* – –   *Signature*

METHOD OF PAYMENT

Remit in US $ drawn on a US bank. Please see next page for ordering information.

☐ **Check or Money Order**

☐ **Mastercard**

☐ **VISA**

☐ **Discover**

☐ **American Express**

Expiry date: _____

Verification code: _____

Send order form to:

**Elektor US
PO Box 180
Vernon, CT 06066
USA**

Phone: 860-875-2199
Fax: 860-871-0411
E-mail: subscriptionsus@elektor.com
www.elektor.com/subs-usa

* Offer available in US and Canada only. Canada please add $11.00 per year for postage.
** Please note: For a Plus subscription, it's important that you fill in your email address. This will enable you to have access to Elektor-plus.usa.com where you can browse an online Elektor archive exclusively available to Elektor Plus subscribers!

ORDERING INFORMATION

To order contact customer service:

Phone: 860-875-2199
Fax: 860-871-0411
Mail: Elektor US
          PO Box 180
          Vernon, CT 06066
          USA
E-mail: sales@elektor.com
On-line at www.elektor.com

Customer service hours: 8:00 AM–4:30 PM Monday–Thursday. Voice mail available at other times.
When leaving a message please be sure to leave a daytime telephone number where we can return your call.

PLEASE NOTE: While we strive to provide the best possible information in this issue, pricing and availability are subject to change without notice. To find out about current pricing and stock, please call or email customer service.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in the magazine. If difficulties in obtaining components are suspected, a source will normally be identified in the article. Please note, however, that the source(s) given is (are) not exclusive.

PAYMENT

Orders must be prepaid. We accept checks or money orders (in US $ drawn on a US bank only), VISA, Mastercard, Discover, and American Express credit cards. We do not accept C.O.D. orders.
We also accept wire transfers. Add $20 to cover fees charged for these transfers.

TERMS OF BUSINESS

Shipping Note: All orders will be shipped from Europe. Please allow 3–4 weeks for delivery. Shipping and handling via airmail: US $20.00 per order. **Returns** Damaged or miss-shipped goods may be returned for replacement or refund. All returns must have an RA #. Call or email customer service to receive an RA# before returning the merchandise and be sure to put the RA# on the outside of the package. Please save shipping materials for possible carrier inspection. Requests for RA# must be received 30 days from invoice. **Patents** Patent protection may exist with respect to circuits, devices, components, and items described in our books and magazines. Elektor accepts no responsibility or liability for failing to identify such patent or other protection. **Copyright** All drawing, photographs, articles, printed circuit boards, programmed integrated circuits, diskettes, and software carriers published in our books and magazines (other than in third-party advertisements) are copyrighted and may not be reproduced (or stored in any sort of retrieval system) without written permission from Elektor. Notwithstanding, printed circuit boards may be produced for private and personal use without prior permission. **Limitation of liability** Elektor shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connection with, the supply of goods or services by Elektor other than to supply goods as described or, at the option of Elektor, to refund the purchaser any money paid with respect to the goods.

SUBSCRIPTIONS (US & CANADA ONLY)

Subscription rates (1 Yr.)

Standard Subscription: $39.95
Plus Subscription: $59.95

Canada add $11 per year for postage

All subscriptions begin with the current issue. Expect 3–4 weeks for receipt of the first issue. Subscriptions, renewals, and change of address should be sent to:

Elektor US
PO Box 180
Vernon, CT 06066
USA

E-mail: sales@elektor.com

Order subscriptions on-line at www.elektor.com/subs

Subscriptions may be paid for by check or money order (in US $ drawn on a US bank only). We accept Mastercard, VISA, Discover and American Express credit cards.

For gift subscriptions, please include gift recipient's name and address as well as your own, with remittance. A gift card will be sent on request.
Subscriptions may be cancelled at any time for a refund of all unmailed issues.

Does your subscription expire soon?
Renew it online at www.elektor.com/renew