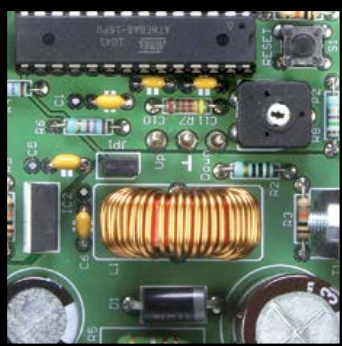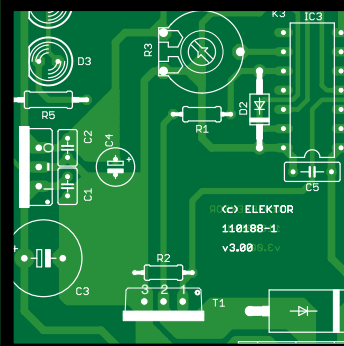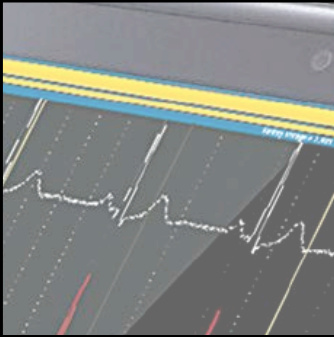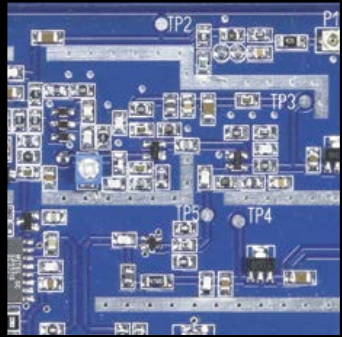# elektor

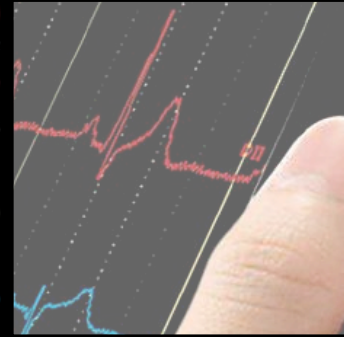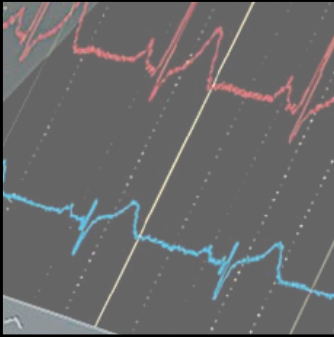## PROJECT GENERATOR EDITION

AMBIENCE LIGHTING CONTROLLER

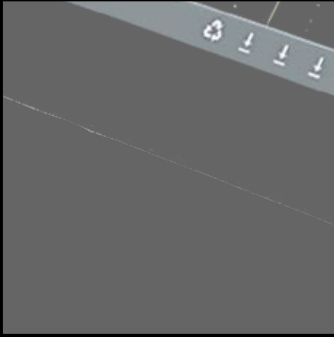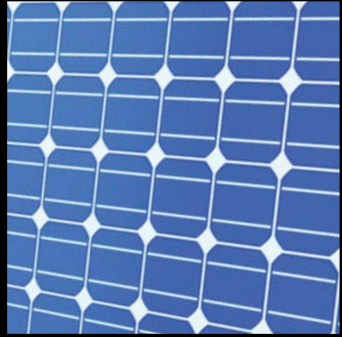**70 CMS WIDEBAND FM EXCITER**

ANDROID ELEKTOR CARDIO-SCOPE

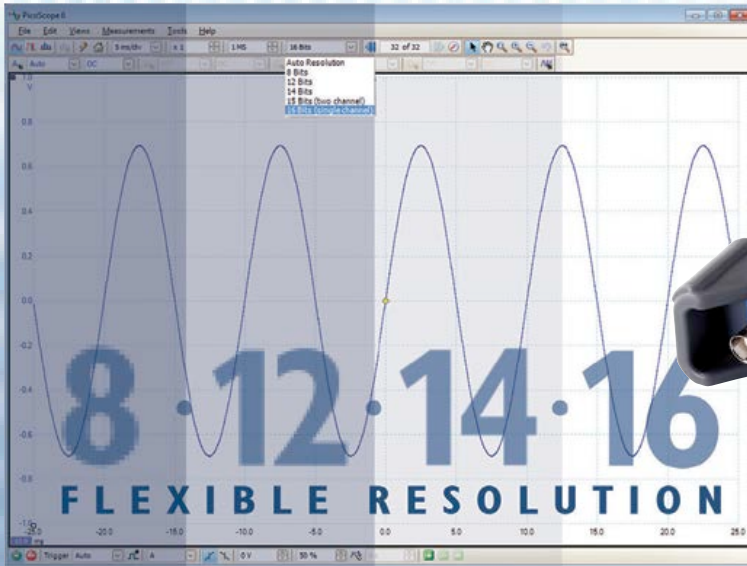**X-TREME INRUSH CURRENT LIMITER**

132 PAGES PACKED WITH NEW DESIGNS

SMARTPHONE A/V REMOTE CONTROL

Contents

● Project Generator

⬤ Labs

⬤ Industry

⬤ Tech the Future

⬤ Magazine

●Community

Project Generator Edition 2013

If you are new to Elektor, it's Hello from me and the Elektor Labs crew, and Welcome to an unusual format of our magazine—the 132-page double edition with its lively blend of small and large projects to keep you busy and 'e'mused over the Summer months. To everyone accustomed to the annual deluge of ideas called Project Generator, it's Hello again and buckle up! In response to your requests I've retained regulars like Hexadoku, Gerard's Columns, Tech the Future, and Retronics cheerfully alongside installments from article series like the 500 ppm LCR Meter, Hexadoku, and Neil Gruending's tips and tricks for DesignSpark.

With the power capacity of the Project Generator 2013 given as 132 pages in spreadsheets, the administrators thought they could accommodate many small projects. Looking back on previous Summer Circuits, the team actually responsible for the contents decided differently. They went for quality and depth rather than quantity, not forgetting to include a few less ambitious circuits covering a page or so, just for the fun of it. All respecting the 132-page payload limit, of course. Surely intentional on part of our Lab team, a good deal of projects this year is related to motors and batteries in some way. The X-treme Inrush Current Limiter and the Big Amps DC Motor Driver are just two examples of circuits designed for use by enthusiasts outside the pure electronics community, like our friends in the R/C model flyer and builder clubs. You're a local hero if you step onto one of their airfields, and offer help and advice on 'electronicizing' their models. In return, you can barter for assistance with your—embryonic—mechanical work.

Speaking of friends it's a pleasure to finally reach out to radio hams again with our 70 cms WBFM Exciter project, and to embedded addicts, with our ElektorBus and several other microcontroller based circuits. Hello also to budding cardiologists— here's a DIY Android-based ECG Meter for you. Heartbeats for the Summer.

Enjoy this special edition,

Jan Buiting, Editor-in-Chief

The Team

Editor-in-Chief:	Jan Buiting
Associate Editor:	Thijs Beckers
Publisher / President:	Hugo Van haecke
Membership Managers:	Shannon Barraclough (USA / Canada), Raoul Morreau (UK / ROW)
International Editorial Staff:	Harry Baggen, Eduardo Corral, Wisse Hettinga, Denis Meyer, Jens Nickel
Laboratory Staff:	Ton Giesberts, Luc Lemmens, Tim Uiterwijk, Clemens Valens, Jan Visser
Graphic Design & Prepress:	Giel Dols, Jeanine Opreij, Mart Schroijen
Online Manager:	Daniëlle Mertens
Managing Director:	Don Akkermans

Our network

elektor

VOICE COIL

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

hifi.nl
MUZIEK - FILM - LIFESTYLE

audioXPRESS

techthe future

Connects you to

285587 members in 83 countries

Supporting Companies

Not a supporting company yet?

Contact Peter Wostrel (peter@smmarketing.us, Phone 978-281-7708, Fax 978-281-7706)
to reserve your own space for the next edition of our members' magazine

Android Elektorcardi♥scope

(1)

Wireless, button-free: Bluetooth & touch screen

The title says it all—this article describes an electrocardioscope you can build yourself, using an Android tablet or smartphone as a wireless terminal for viewing the electrocardiograms. The project involves skillfully combining a small PIC interface to control an analog input stage with a great deal of software.

By **Marcel Cremmel** (France) in co-operation with Raymond Vermeulen (Elektor Labs)
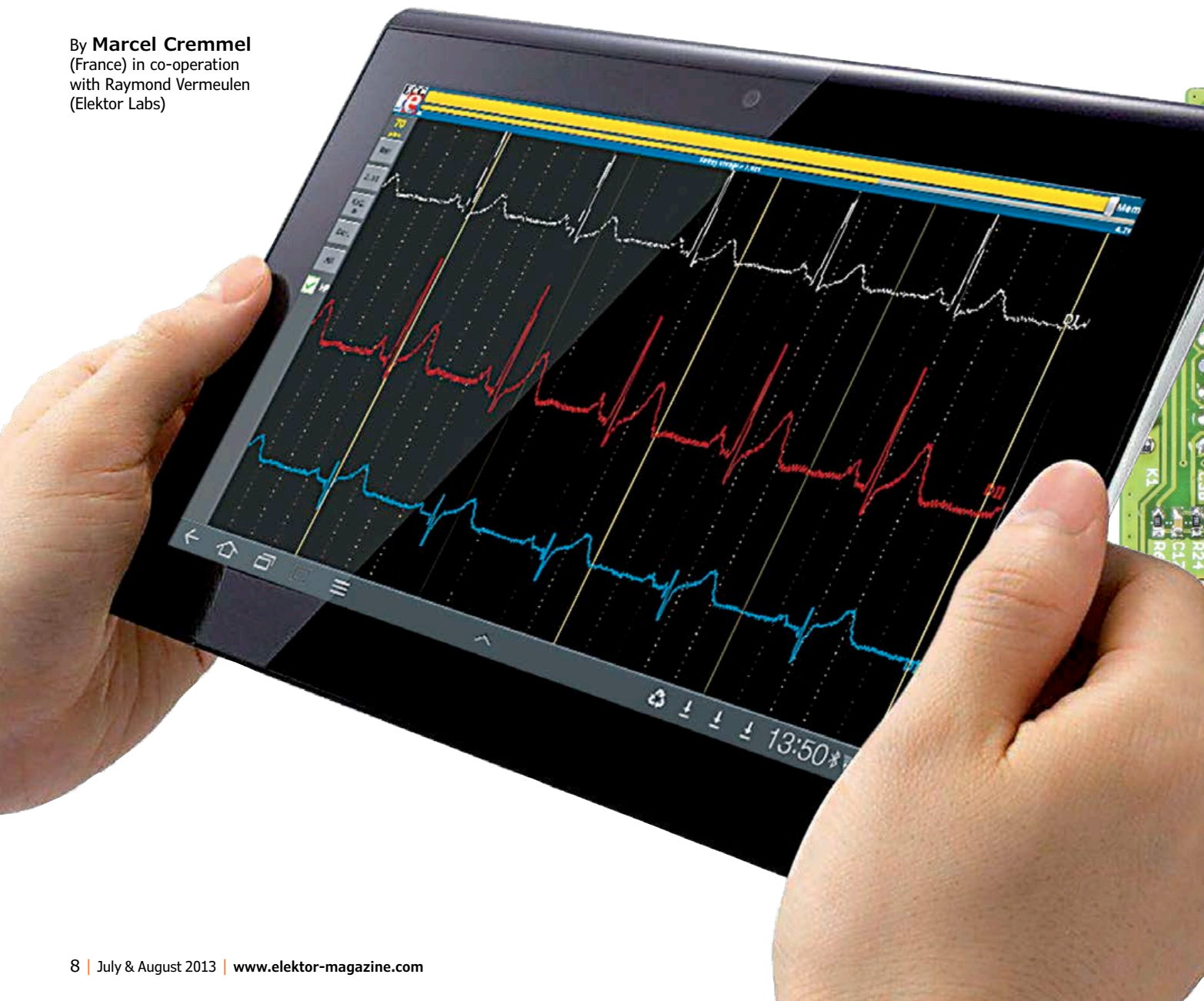
Technical Specifications

- Interface for Android phones or tablets with Bluetooth
- Simultaneous or individual scrolling display of the three standard leads (DI, DII, and DIII) and the three enhanced leads (aVR, aVL, and aVF)
- Adapts automatically to screen resolution
- Measures and displays cardiac rhythm
- Audible heartbeat indication
- Scrolling speeds: 250/125/62.5 and 31.25 pixels per second
- Display gain: ×1 ×1.2 ×1.5 ×2 ×3 and ×10
- Full-scale sensitivity: 3.2 mV. 10-bit conversion
- Sampling frequency: 2 kHz
- Common-mode rejection: > 100 dB
- Max. contact voltage: ± 150 mV
- Auto-adaptive alignment time constant
- Frequency response: 0.005 Hz to 170 Hz
- Dynamic trace memory: 10 minutes
- ECG recording in flash memory (10 min)
- Periodic injection of 1 mV/2 Hz calibration signal
- Powered by 2 no. 1.5 V primary or 1.2 V rechargeable cells
- Constant display of battery voltage
- Current consumption: 50 mA (standby: <4 µA)
- Battery life: 15 hours (1 Ah cells)
- Inexpensive
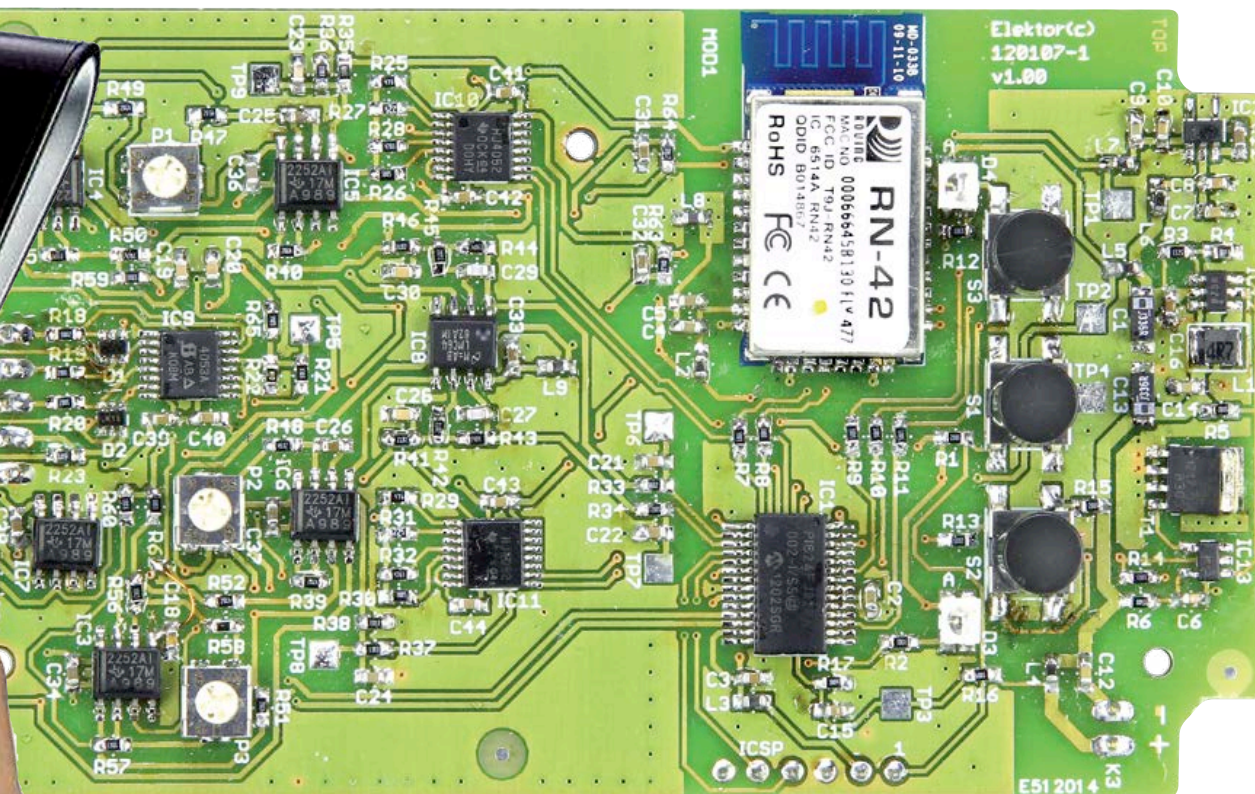
When Elektor published an electrocardioscope based on a GameBoy games console back in 2006, the little Android robot didn't yet exist—or was at an embryonic stage at most. Seven years on, a vast number of applications are using it; at the time of publication, there are over 900 million Android devices in circulation. Each day, new horizons are opening up for the little green fellow. Elektor is happy to be contributing to this saga that's only just beginning with an application that's instructive, fascinating, and potentially useful to everyone: perform your own
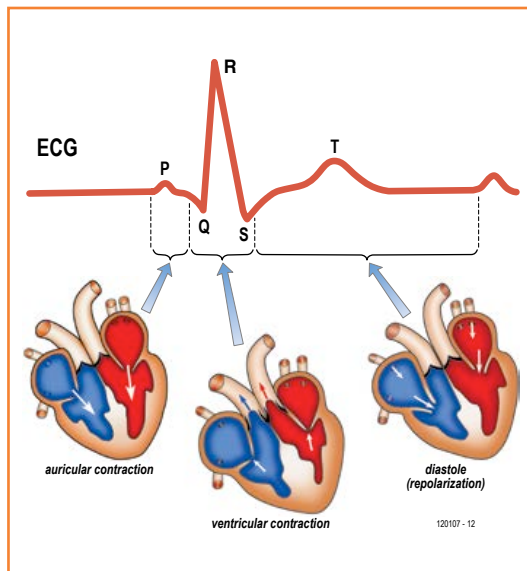
Figure 1.
Correspondence between the electrical activity detected and the phase of a cardiac cycle.

The computing part is quite substantial, so we don't intend going through the source code in detail; but to encourage you to approach it, customize it, or even improve it, we're going to be describing the software functions. To help us do this, you'll find the identifiers used in the diagrams in the source code. The complete MPLAB and Android project files are available on the Elektor website.

Medical imaging and electronics

The Elektor electrocardioscope makes use of the process invented by Willem Einthoven (see Elektor October 2006 [1]), which involves exploring cardiac function using the image of the electrical phenomena produced during the cardiac cycle. The heart is an autonomous muscle—the only one not controlled by the brain. The sinus node, located in the right-hand atrium, triggers the nerve impulses that control the heart muscles. These contract (depolarization) and relax (polarization) to pump the blood. The contraction is caused by a change in the electrical polarity of the cell membranes. During the relaxation

electrocardiograms on your smartphone or tablet! This physiological exploratory accessory consists of a single (very small) board 5.5 × 10 cm (2.2 x 4 inches) carrying the analog and digital sections of the circuit. Our ECG interface is available in

Record and view your own electrocardiograms on your smartphone or tablet!

the form of a ready-to-use module to which you just have to add four electrodes and an Android application for smartphone or tablet; there's no physical connection between this terminal and the interface, as it uses Bluetooth communication! As you'd expect for a modern device, this one has only very few adjustments (three presets in all) or controls (3 buttons). Clearly, it's the software that does all the work.

We're going to be describing it in at least two articles:

- the electronics: our favorite field,
- the program running on the PIC24 microcontroller, written in C under MPLAB (free IDE and compiler from Microchip),
- the Android application, written in Java using Google's SDK,
- building the interface, to be fitted in a small case the size of a smartphone.

phases, the electrical charges are balanced again before a new stimulus comes along. The resulting potentials are transmitted as far as the surface of the skin, where they can be detected using skin electrodes.

With appropriate positioning of these electrodes, a cardiologist is able to deduce the mechanical behavior (and possible dysfunctioning!) of the heart by analyzing its electrical activity [2]. **Figure 1** shows the relationship between the electrical activity detected and the phase of a cardiac cycle:

- *P wave:* contraction of the atria: the blood from the veins is forced into the ventricles.
- *QRS complex:* contraction of the ventricles: the blood from the ventricles is driven into the arteries.
  These two waves are what cause the "lub-dub" sound of the heartbeat.

- *T wave:* repolarization of the ventricles: the ventricle muscles go back to their resting state.

The positions of the four electrodes, one on each wrist and ankle, are shown in **Figure 2**, which also describes the leads displayed by the device. The limb leads are positioned on the inner face of the wrists and ankles. They explore the cardiac field in a frontal plane (**Figures 2a & 2b**). These three leads form Einthoven's equilateral triangle, the polarities being such that DIII=DII−DI. Replotting the DI, DII, and DIII axes to the center of the triangle makes it possible to construct a system of co-ordinates (known as Bailey's hexaxial system), which is useful in calculating the activation vector in the frontal plane (**Figure 2c**). The six leads represent the cardiac activity according to these six axes. Analyzing them makes it possible to establish a diagnosis.

The three active electrodes LA (Left Arm), RA (Right Arm), and LL (Left Leg) detect the DI and DII leads; the fourth, RL, is used to set the quiescent voltage of the other three.

The interface electronics amplify the emfs picked up by leads I and II to produce the DI and II signals needed to produce electrocardiograms. These voltages are converted into digital quantities before being transmitted via Bluetooth to the Android terminal. The application calculates the missing leads (DIII, aVR, aVL, and aVF) from DI and DII (formulæ in **Figure 2b**) and displays these. The screenshots illustrating this article leave us in no doubt whatsoever: the signals are excellent and the noise low, as are residual powerline frequency components.

Analog electronics

The analog section of our ECG has the task of amplifying the two *very weak* voltages picked up between, on the one hand, active electrodes LA and RA, and LL and RA on the other. These are the DI and DII leads (**Figure 2**). To obtain adequate dynamic range after the 10-bit A/D conversion, we need a gain of 1,000 (i.e. 60 dB). This is combined with two other essential functions: compensation for the electrodes' DC contact voltages (which can exceed 100 mV, i.e. 100 times the amplitude of the wanted signals) and rejecting the powerline frequency (60 Hz or 50 Hz). We'll come back to this later.
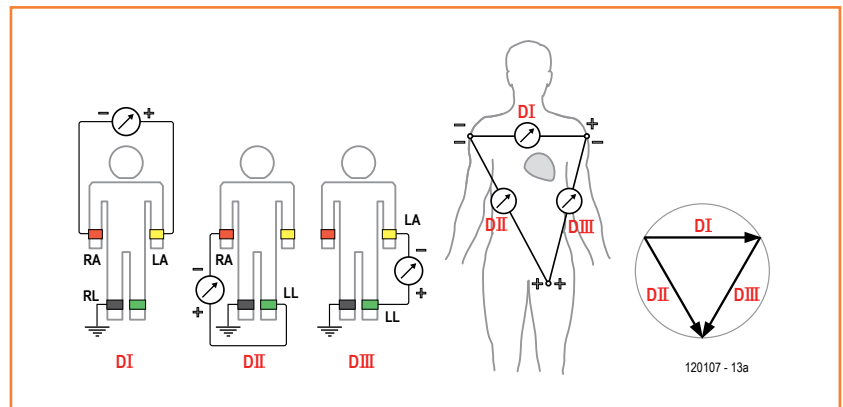
Figure 2a. Standard bipolar leads.

$$aVR = DII + \frac{DIII}{2}$$
$$aVL = DI - \frac{DII}{2}$$
$$aVF = DIII + \frac{DI}{2}$$
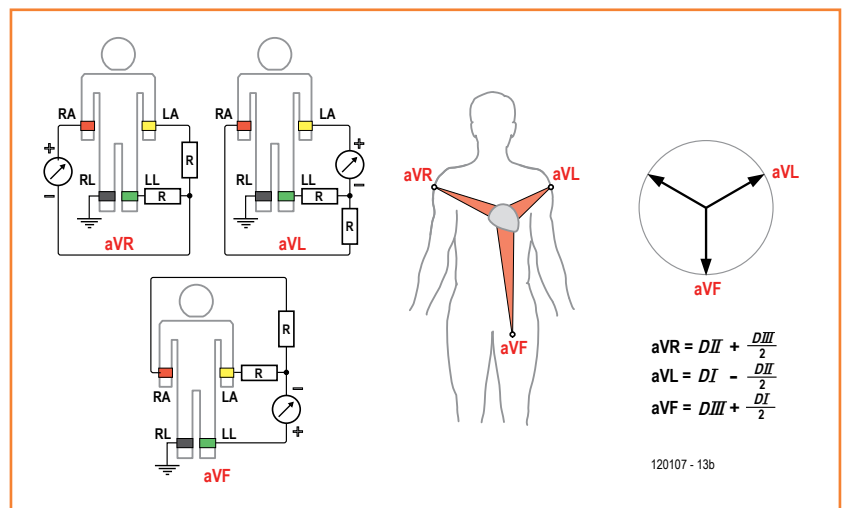
Figure 2b. Standard unipolar leads or enhanced leads.
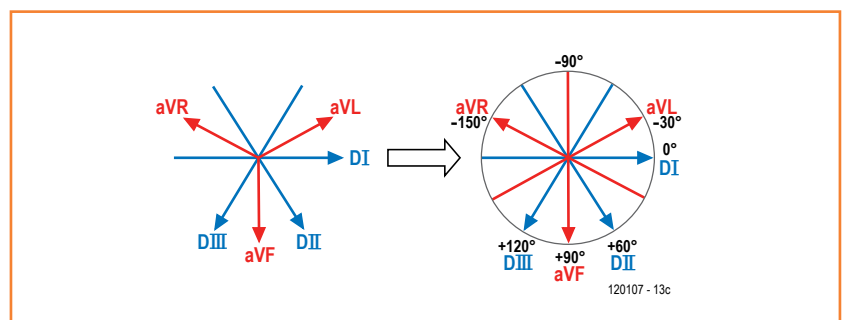
Figure 2c. Co-ordinates system used for calculating the leads.

The human body and the electrode connecting leads are considerably influenced by the strong voltages or voltage differences with respect to ground on the electrical wiring of the surround-

F1: R18, 19, 20, 23, D1, D2
F2: IC9
F3.1: IC4A - IC4B, R53, 54, 55
F3.2: IC3A - IC3B, R57, 58, 56, P3
F4.1: IC5A, R49, 40, 47, 50, P1
F4.2: IC6A, R51, 39, 48, 52, P2

F5.1: IC10, R35, 36 C23, R25, 26, 27, 28, IC5B, C25
F5.2 : IC11, R37, 38 C24, R29, 30, 31, 32, IC6B, C26
F6.1: IC8A, R46, 44, 45, 33 C30, C29, C21
F6.2: IC8B, R41, 42, 43, 34 C28, C27, C22
F7: R62, 59, 60, 61
F8: IC7A, C17, R24

120107 - 14
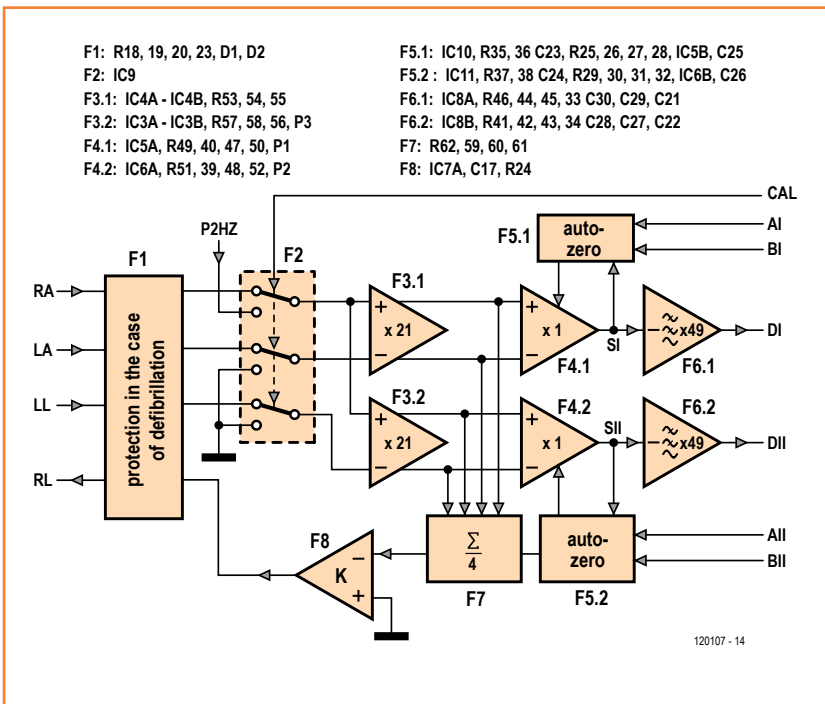
Figure 3.
Function diagram of the analog section.

ing space. Despite the 60 / 50 Hz frequency, mutual capacitance, even though it is very low, causes a relatively high voltage (often over 1 V) to be present on the skin with respect to ground. At first sight, it would seem difficult to isolate the wanted signal, as its amplitude is 1,000 times lower than that of the spurious signal! What's more, the frequency of 60 / 50 Hz falls within the wanted spectrum; so filtering is no use.

Given the wavelength at 50 Hz (6,000 km!), we can accept that, as the skin is conductive, every point on the epidermis is at the same voltage as all the others. Thus as far as the electrodes are concerned, this is a **common-mode** voltage.

In this case, the solution becomes obvious: a **differential instrumentation amplifier** with an adequate common-mode rejection ration (CMRR):

$$ CMRR \geq \left[\frac{S_p}{S_{ECG}} \right]_{dB} + \left[\frac{S}{N} \right]_{dB} $$

where S_P is the amplitude of the spurious signal: 1 V;
S_{ECG} is the ECG amplitude: 1 mV;
S/N is the signal-to-noise ratio: 40 dB required.
i.e. **CMRR ≥ 60+40 = 100 dB**

In addition, the amplifier needs to have very high input impedance (>10 MΩ) and low offset volt-

age. It's very tempting to use a specialized integrated circuit for this (e.g. the ADS1294 from TI). But it is also possible, without any concessions in terms of quality, to build a differential amplifier using standard TLC2252 operational amplifiers (op-amps), a low-power rail-to-rail type (for dynamic range) with low noise. Its cut-off frequency and slew rate are only modest—but more than enough for an ECG signal. The major (but acceptable) concession in this choice is the pre-set resistor to optimize the CMRR, which cannot be dispensed with.

Functions of the analog section

Before looking in detail at the ECG interface circuit, let's examine its structure (**Figure 3**) and functions. The four electrodes are connected to the inputs RA (right arm), LA (left arm), LL (left leg), and RL (right leg). As the electrocardiograph is sometimes use in conjunction with a defibrillator, it needs to be protected against the high voltages produced by this type of device: this is the job of function **F1**. Under normal conditions, this function's output voltages are equal to its input voltages. The protection only comes into operation if the voltages measured exceed the level of the supply rails, i.e. ±3.3 V.

The multiplexing function F2 lets us replace the RA, LA, and LL signals by a P2HZ calibration signal with an amplitude of 1 mV and a frequency of 2 Hz. The multiplexer is activated on request once a minute for 10 s in order to observe the calibration signal on the screen. When establishing a diagnosis, the calibration signal gives an amplitude reference for the ECGs measured. The instrumentation amplifier is formed by functions F3 and F4. Function **F6** is a 2nd order low-pass filter with a characteristic frequency of 170 Hz and damping factor m = 0.71 (Butterworth). This attenuates all the components outside the wanted frequency spectrum and serves as an anti-aliasing filter for the following ADC (**A**nalog/**D**igital **C**onverter).

The overall gain is obtained as follows: AD3 = 21×, AD4 = 1×, and A6 = 49×. The gain reference corresponds to the reference of the functions: AD3 is the differential amplifier of F3 (F3.1 or F3.2), and so on. In line with our objectives, the total gain is 1029×. The gain in the first two stages is relatively low in order to increase the effectiveness of the auto-zero function.

The other functions (F5, F7, and F8) support the instrumentation amplifier to ensure proper opera-

tion. The operational amplifiers are powered at between −3.3 V and +3.3 V. The ideal quiescent voltage for each of their three terminals is 0 V. The RL electrode sets the average voltage (quiescent point) for the active electrodes via functions F7 and F8. It is shown that the F7 output voltage is equal to the average of the RA, LA, and LL voltages. This is compared by F8 with the 0 V set point, and the amplified error voltage produces the voltage of the RL electrode. As no current flows in the electrodes, the RA, LA, and LL voltages are equal to that on RL (to within a few mV). In this way, we create a real servo loop through the skin: the average of the active electrode voltages is maintained equal to the set point of 0 V. That's what we were aiming for: the

input amplifiers' quiescent voltages are correct, without impairing their intrinsic input impedance. In addition, a natural but highly inconvenient phenomenon appears when the electrodes are fitted: a contact emf is generated between the skin and the metal of the electrode. This micro-battery is very weak (a few tens of mV) but is not eliminated by the instrumentation amplifier. Quite the reverse: it is amplified!

Functions F7 and F8 partially reduce this phenomenon, but the offsets between the + and − outputs of F3.1 and F3.2 can still reach several volts. These offsets are compensated by functions F5.1 and F5.2, to avoid saturation in the F4.1, F4.2, F6.1, and F6.2 stages.

F5.1 and F5.2 compare the average values of

Figure 4a.
Analog schematic.

120107 - 11
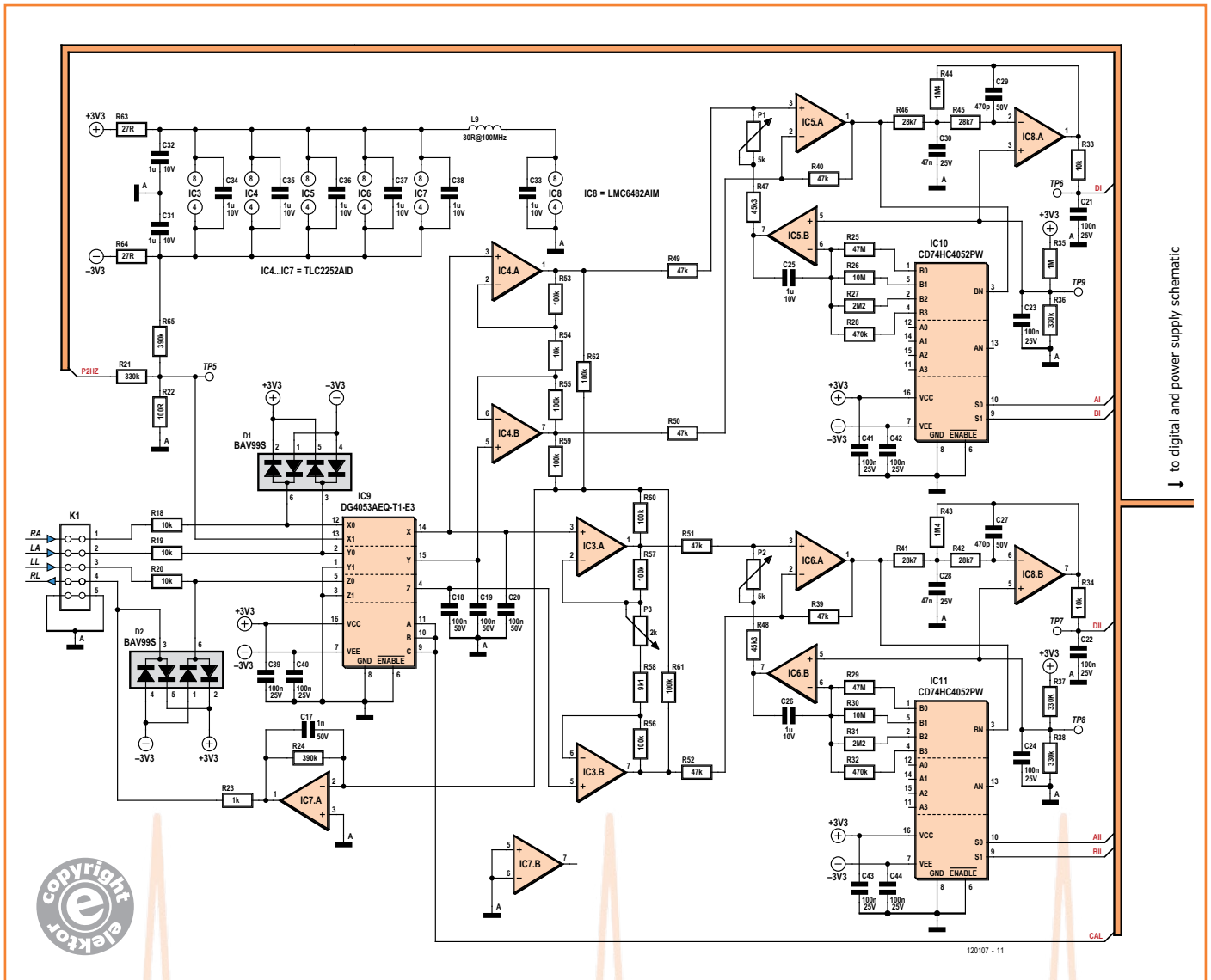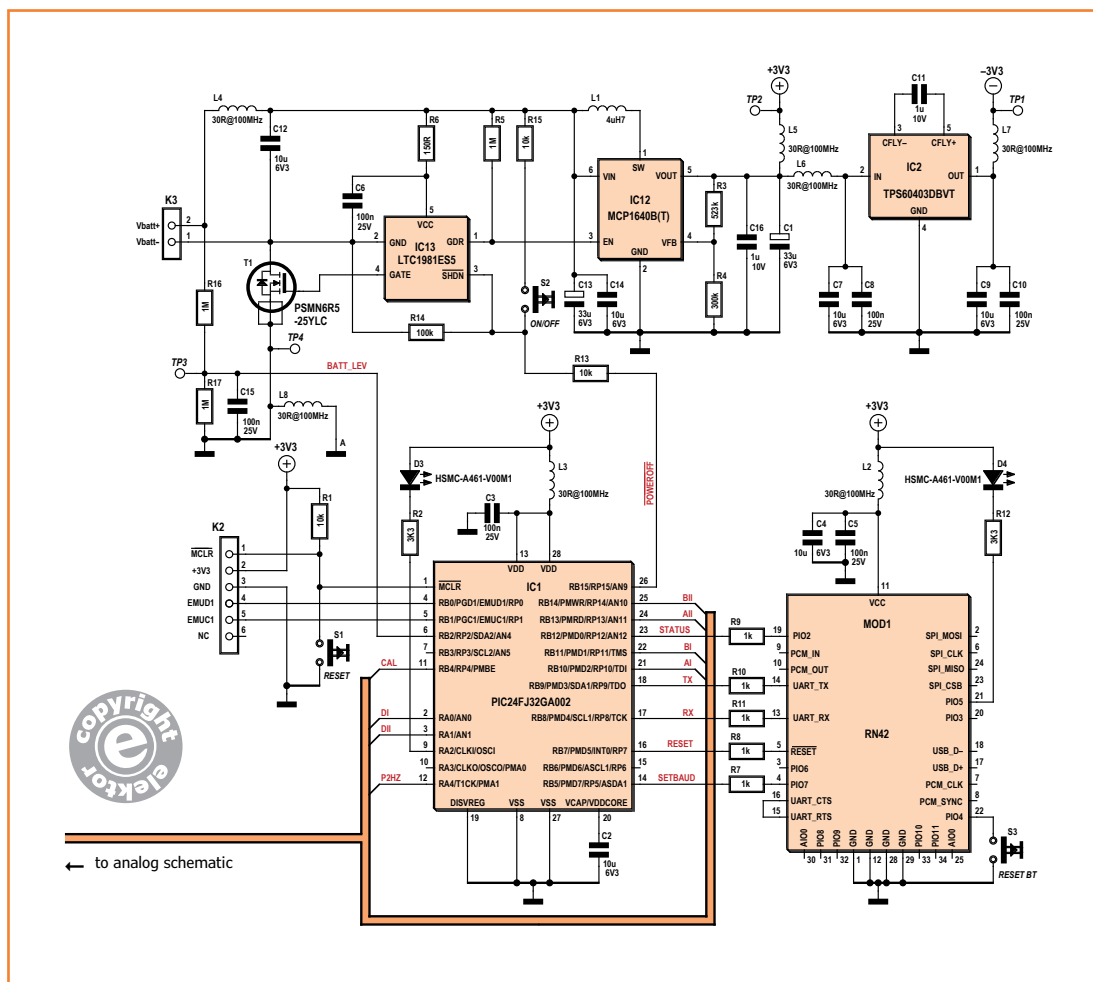
Figure 4b.
Digital and power supply
schematic.

the SI and SII signals with a fixed reference. The error voltage is integrated using a time constant selected using AI and BI (AII and BII respectively for the II channel) to derive the DC offset voltage applied to F4.1 and F4.2. This offset is found on SI and SII. The servo loop thus formed constantly maintains the average values of SI and SII at their respective set points.

Heart of the circuit
You'll easily be able to find all these function on the diagram for the analog circuit (**Figure 4a**). The four electrodes are connected to K1. Resistors R18, R19, R20, R23, and the integrated diodes in D1 and D2 form the F1 protection function. The multiplexing function F2 is performed by the analog multiplexer IC9 (4053).
The differential amplification for each channel (F3 and F4) is achieved using the classic instrumentation amplifier configuration: F3.1 = IC4A + IC4B and F3.2 = IC3A + IC3B, along with F4.1

= IC5A and F4.2 = IC6A.
The DII channel gain is adjustable by P3 to compensate for the difference from the other channel. Even a small difference in gain would have a noticeable effect on the calculations for the other leads.
Presets P1 and P2 let us optimize the common-mode rejection for each amplifier.
The F4 stages include a negative feedback loop via (in the DI channel, for example) multiplexer IC10, one of the resistors R25–R28 and op-amp IC5B wired as an integrator.
This structure performs the F5.1 function and acts on the average voltage of DI via IC5A in such a way as to make it equal to the voltage set by the divider R35/R36 (and R37/R38 respectively for the DII channel).
The time-constant for this loop is selected by a software function driving the multiplexer IC10 (and IC11 in the other channel) via the AI and BI signals:

- T1 = R28·C25 = 0.47 s
- T2 = R27·C25 = 2.2 s
- T3 = R26·C25 = 10 s
- T4= R25·C25 = 47 s

When the electrodes are applied, the shortest time-constant is selected so as to reach the quiescent point quickly. The time-constant is then increased as the anticipated quiescent point is approached, ending up at the value of 47 s, which will not interfere with the ECG signals.

The quiescent voltages are different between the DI and DII channels in order to be suitable for the forms expected for each lead, thereby avoiding saturation in the following ADC while maintaining adequate dynamic range.

The final stage in each channel (IC8a and b) forms a 2nd-order low-pass filter. This has a cut-off frequency of 170 Hz and in-band gain of 34 dB. This is an anti-aliasing filter for the ADC, which uses a 2 kHz sampling frequency. This is followed by a 1st-order low-pass filter formed by R33 and C21 with a cut-off frequency of 160 Hz. The attenuation of these two filters at the Nyquist frequency (1 kHz) is around 15 dB.

Did you notice IC8 is powered between +3V3 and 0 V and not between +3V3 and −3V3 like the other op-amps? This is not a mistake: this arrangement protects the PIC24's analog input, which can't take negative voltages.

Because of its high sensitivity, the analog section is very sensitive to interference of all kinds, in particular that produced by the digital section and the switch-mode power supply. The circuit layout has been carefully designed to keep these three sections well away from each other. In addition to this, the networks R63/C32 and R64/C31 attenuate the residual ripple on the supply rails to the analog section.

For the digital part of the circuit, the diagram (**Figure 4b**) is clear enough, we'll dispense with a block diagram.

PIC24FJ32GA002

Out of Microchip's PIC24 variants, we've chosen the PIC24FJ32GA002, an entry-level type in a 28-pin SOIC package (**Figure 4b**). Its 8 MHz internal clock (no external components required) gives a power of 4 MIPS and 2 % accuracy, adequate for this application. At this frequency, the consumption of around 5.4 mA is reasonable for battery powering.
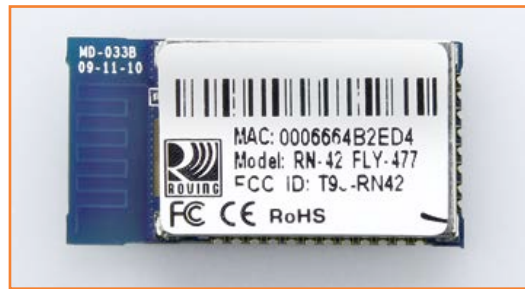
Figure 4c.
Photo of the Roving Networks (now Microchip) Bluetooth module used.

RN-42 Bluetooth module

The Roving Networks RN-42 (**Figure 4b**) is a compact, low-power, pre-qualified class 2 hybrid OEM Bluetooth module. (Roving Networks were recently acquired by Microchip, *Ed*.) The built-in antenna allows a range of about 70 feet (20 m). It supports, among others, the RFCOMM-SPP protocol used in this application, at data rates up to 240 Kb/s. Here, we only use 16 Kb/s.

The Serial Port Profile (SPP) makes it easy to establish a sort of wireless duplex asynchronous serial link. The RX and TX signals from the microcontroller's UART coupler are simply connected to their equivalents on the Bluetooth module. In this way, the sequences produced on TX in the asynchronous serial format are transferred transparently to the connected terminal. Conversely, messages sent from the Android terminal appear in the same format on RX.

A few additional signals are used to control the module:
the STATUS signal indicates the status of the Bluetooth link: Connected (1) or Broken (0)
the RESET input allows the module to be initialized if necessary
the PIO7 input (SETBAUD) lets us select the transmission data rate:
- 1 → 9600 baud,
- 0 → 115.2 Kbaud

The 1 kΩ resistors in series with these signals are in accordance with the manufacturer's I/O protection recommendations.
LED D4 flashes to indicates the module status:
10 Hz: configuration in progress
2 Hz: initialization phase
1 Hz: the module can be identified
steady: connection established.

Powering

For such a simple circuit, the structure of the power supply (**Figure 4b**) comprising IC13, IC12,

A project to your heart's content

and IC2 is fairly complex, but it does guarantee a stabilized symmetrical voltage (±3.3 V) right down to the last drop of power available in the two 1.5 V AA cells or the pair of 1.2 V rechargeables.

It is out of the question to power this sort of device from the powerline!

MOS transistor T1, associated with step-up device IC13, protects the electronics in the event of reversed battery polarity: R6 limits the current through IC13, T1 is turned off along with its internal diode. The battery −ve is not connected to

GND and the ICs are protected, in particular IC12. When the polarity is correct, IC13 is powered correctly. It is controlled by the microcontroller's /PowerOff signal. If the interface is powered down, the PIC24 is not powered and cannot apply a H to /PowerOff. Hence the /SHDN input is low, particularly because of R14. IC13's GATE output is then low and T1 is turned off. However, its flyback diode is conducting (the current consumed flows to GND via the battery −ve) and IC12 is powered. Under these conditions, IC13's GRD output is at 0, inhibiting the switching converter IC12, which as a result does not generate the 3V3 supply voltage. Hence inverter IC2 that supplies the −3V3 downstream doesn't operate. The

interface is well and truly powered down. The only consumption is that of IC13 and IC12 in standby, i.e. less than 4 µA.

Pressing S2 takes IC13's /SHDN input high. Its GATE output turns T1 on with a voltage that is three times its supply voltage, thanks to an internal tripler. This makes it possible to drastically reduce T1's R_{ON}. The battery −ve is connected to GND.

IC13's GDR output also goes high and enables IC12. This then produces the +3V3 voltage from the battery voltage. Inverter IC2 is now powered and in turn supplies the −3V3 rail. The interface is powered.

The microcontroller then starts processing and must quickly follow the user's pressing S2 by setting the /PowerOff signal to High (1).

Powering down is controlled from the Android terminal when the user exits the application: the /PowerOff signal is set to L. IC13 then goes into shutdown and sets its GDR output to 0 to inhibit IC12 and hence cut the power.

The switching converters used make it possible to obtain an overall efficiency of nearly 90 % and will operate (but not for long) with low batteries down to 0.8 V.

In the next issue of Elektor, we'll be presenting the software, the Android application, and the board, together with how to implement the device. You'll see it's amazingly easy.

120107

Internet Links

[1] GBECG article in Elektor October 2006, www.elektor.com/050280

[2] Course in cardiology: http://goo.gl/mSr20

[3] www.elektor.com/120107

[4] Android Apps, programming step by step, Stephan Schwark, www.elektor.com/android

[5] Le Site du Zéro
www.siteduzero.com/informatique/tutoriels/apprenez-a-programmer-en-java
or http://goo.gl/OVZQY (in French)

[6] author's own website: http://electronique.marcel.free.fr/ (in French)

Step Exercises

Stepper motor driver for the ElektorBus

By **Ronald Stärz, Mathias Gfall** and **Jens Nickel**
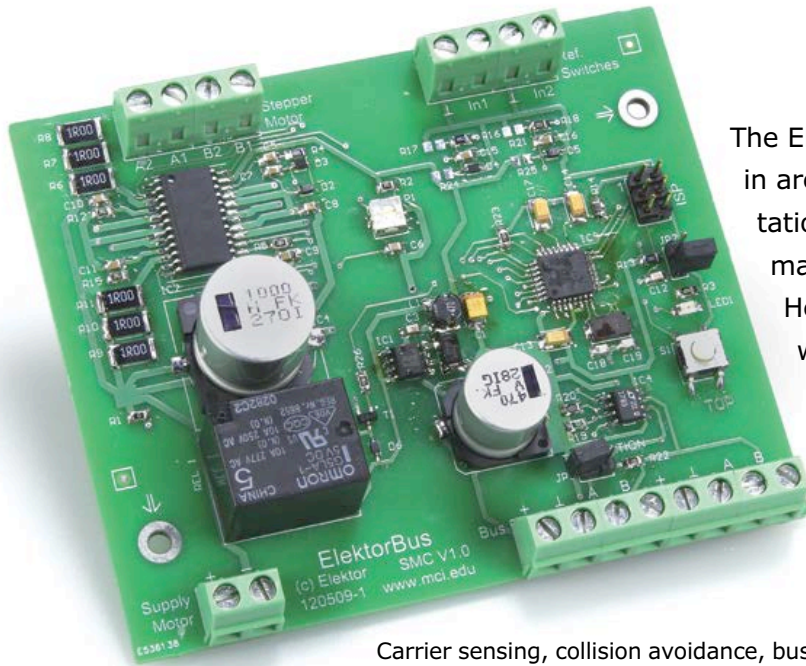
The ElektorBus opens up a wealth of possibilities in areas from home automation to instrumentation. The modular hardware and software makes it easy to build your own applications. Here we extend the ElektorBus 'toolbox' with a board to control a stepper motor.

Carrier sensing, collision avoidance, bus termination – these telecommunications terms are unlikely to set a student or hobbyist's pulse racing with excitement. However, the topic can be made more interesting if students have the opportunity to build their own circuits and projects within a bus-based system. The ElektorBus provides the perfect foundation for this, building on the reference designs. Indeed, the Elektor-Bus is now central to the study of mechatronics at MCI (Management Center Innsbruck) in Austria [1]. The circuit of this stepper motor board is an example of student project work.

The printed circuit board is available from Elektor either as a blank board or as a ready-assembled and tested module. Software that can be used as a basis for your own applications is available, as always, as a free download [2].

A journey of a thousand miles...

The circuit is organized into a number of blocks, as illustrated in **Figure 1**. In the middle are the microcontroller and the stepper motor driver. Around these components are the bus interface, a (switch-mode) power supply and universally compatible position/limit switches.

The stepper motor itself is controlled by an STMicroelectronics L6208 device. **Figure 2** shows what is inside this component [3]. You will see the integrated flyback diodes which are necessary to protect the power transistors from being damaged by the inductive load presented by the motor windings. This particular driver device was chosen because it contains not just the power output stages, but also a state machine to generate the necessary control signals, a current regulator for

Features

- STMicroelectronics L6208 stepper motor driver IC
- Suitable for bipolar stepper motors up to 60 V
- Maximum motor current approx. 2.5 A
- RS-485 interface
- LED and button for test, debug and operation
- Safety relay for motor power (under software control)
- Microcontroller and board files for the Embedded Firmware Library (EFL) included in download
- EFL stepper motor library included in download
- ElektorBus-compatible open source demonstration application included in download

the motor windings, and an over-current detection circuit. It is therefore an easy device to use: we simply need to provide four digital control signals, for rotation direction (CW/CCW), step (CLK), operating mode (HALF/FULL) and strobe (EN). Because of the device's high level of integration the number of external components is kept to a minimum, as can be seen from the circuit diagram in **Figure 3**. The internal current regulator measures the voltage across resistors R6 to R11 on the two sense inputs to determine the motor winding current; the reference value for the regulator is obtained from the V_{ref} inputs, which in turn comes from trimmer potentiometer P1. With the values given in the circuit diagram the winding current is 100 mA when the reference voltage is set to 33.3 mV. The networks comprising R12 and C10, and R15 and C11, determine the off-time of the power MOSFETs in the bridge driver [3]. The L6208 uses a charge pump to generate the gate voltages needed, which requires external components D2, D3, R4, C5 and C7. R5 and C9 condition the voltage at the enable input.

The microcontroller selected is the ATmega328 which, with 32 Kbyte of flash memory, has enough space to hold a modular software library such as the Embedded Firmware Library (see below). Support circuitry for the microcontroller includes the standard components for voltage regulation and clock generation, as well as a manual reset facility using JP2. A button connected to PD5 (S1) and a LED on PD4 (LED1) are very handy both in operation and when debugging new applications. The physical interface to the bus is implemented using an LT1785 RS-485 transceiver. The bus can be terminated by resistor R22 if JP3 is fitted, should the stepper motor board be either the first or the last node on the bus. The screw terminals for the RS-485 signals and for the 12 V supply are, as in the familiar ElektorBus experimental node [4], both duplicated: this makes it more convenient to loop the four wires of the bus through the node.

Power supply

Voltage regulator IC1 (an LM2675M-5) provides the power supply for the logic circuitry. In combination with L1, D1, C1 and C3 it provides an almost lossless conversion from the input voltage, which can be anywhere from 7 V to 24 V, to the 5 V required. The supply for the power electronics is brought out to separate screw terminals,

so that the current required to drive the stepper motor does not have to be provided over the 12 V lines on the ElektorBus. And in this area a small disadvantage of the L6208 stepper motor driver makes itself known: if power is present for the power side of the driver circuit but the 5 V for the logic side of the driver circuit is not (yet) present, the device makes its disapproval apparent in the rather undesirable form of a smoke signal! In the circuit we have therefore taken the power to the L6208 via a relay (REL1) which is under software control of the microcontroller on pin PD6. This also means that there is a possibility of controlling the power to the power section of the driver by means of a signal over the ElektorBus. Since currents of several hundred milliamps can give

Figure 1.
Block diagram of the ElektorBus stepper motor board.

Figure 2.
Block diagram of the STMicroelectronics L6208 stepper motor driver (from [3]).
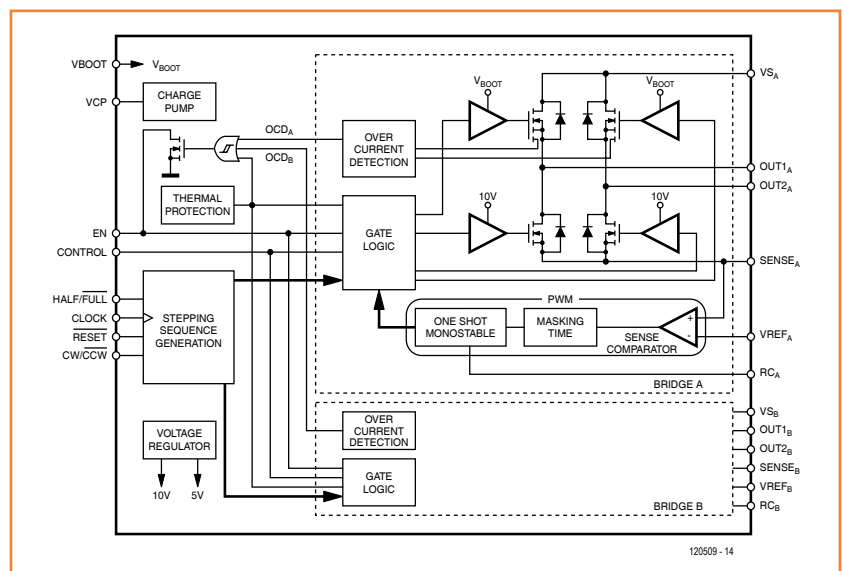
rise to problematic voltage drops in ground wiring, we have provided a ground star point at R1 (a zero-ohm resistor). R1 allows the grounds to be separated between the logic and power parts of the circuit in the event of a fault.

If the stepper motor is to be used in an application where its rotor position is important rather than as a drive where just its (variable) speed of rotation matters, limit switches or position-

detecting sensors will have to be used. Options here include mechanical devices, proximity switches operating on an inductive or capacitive principle, and slotted optical sensors, to name just a few possibilities. In order to keep options as open as possible the board includes two input channel circuits that can be adapted to almost any sort of sensor, formed from R16, R17, R25, C15 and D4 for the first channel and R18, R21, R25, C16 and D5 for the second.
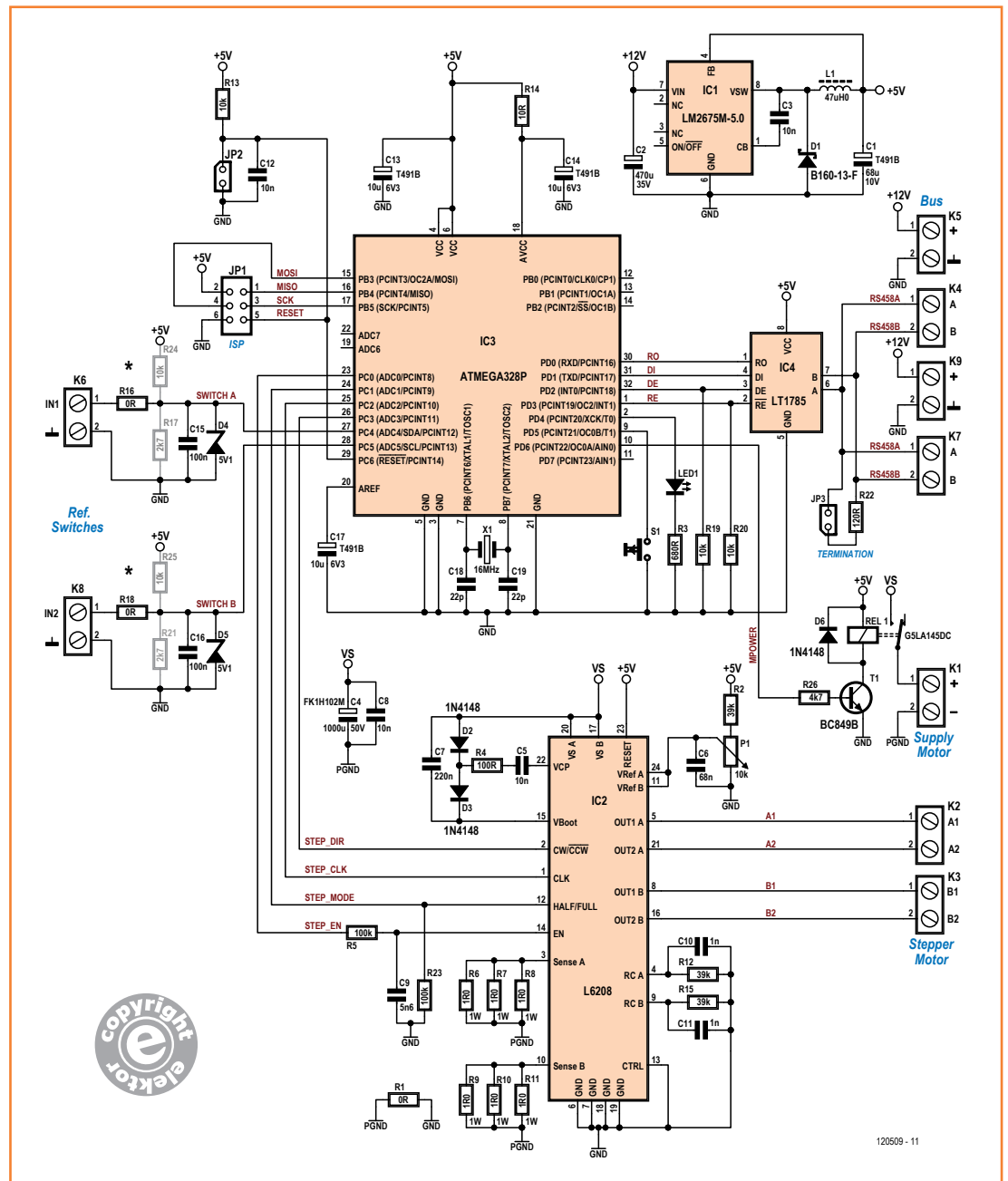
Figure 3.
Circuit diagram of the stepper motor board. See the box for the possible position detection switch configurations.

Position detection switch configurations

Switch:

If a mechanical switch is to be connected, fit a zero-ohm resistor for R16 and do not fit R17. The 10 kΩ pull-up resistor R24 is optional but must be fitted if a different microcontroller is used without internal pull-up resistors. When the switch is closed the microcontroller's corresponding input pin will be taken low. This is the configuration in which the ready-built boards are shipped.

Voltage input (5 V):

Bridge R16 and do not fit R24 to connect the filtered 5 V input signal to the microcontroller's input.

Voltage input (12 V):

A voltage divider with a ratio of 25 kΩ / 10 kΩ = 2.5 produces a voltage of 4.8 V at the input to the microcontroller.

Voltage input (24 V):

The voltage divider comprising R16 and R17 gives an output voltage of 4.9 V.

NPN open collector:

Switches with open-collector outputs require extra circuitry to generate an output signal, in the form of a pull-up resistor to adapt the signal level as required. If a current flows into the base of the transistor the collector-emitter diode will conduct and pull the microcontroller's input down to ground.
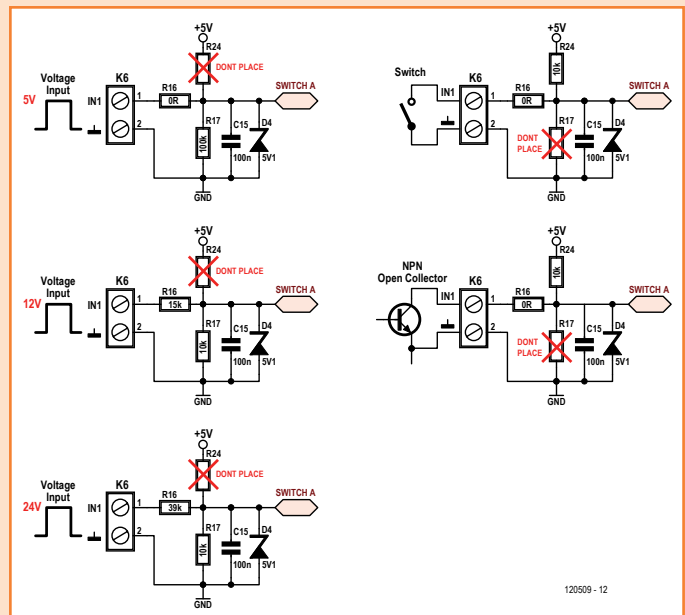
Capacitors C15 and C16 and Zener diodes D4 and D5 filter the input and provide some protection, and are needed in all cases; the resistors are fitted as required. The text box shows the various interfacing possibilities. As shipped R17 and R21 are not fitted and R16 and R18 are zero-ohm resistors: this arrangement is suitable for simple mechanical switches or buttons. Pull-up resistors R24 and R25 are optional, as the software-configurable internal pull-ups in the AVR microcontroller are adequate to ensure correct operation.

Software

The choice of an ATmega328 microcontroller was not made at random: the device has proven its capability in the ElektorBus experimental node design, and since the ATmega88 device used in the ElektorBus article series [5] is compatible with it, we already have some software available which we can modify for our purposes.

We can easily create highly reusable and adaptable code for the board using the 'Embedded Firmware Library' (EFL) described in the May 2013 issue of *Elektor* [6]. This includes a library supporting ElektorBus communications (**Figure 4**). Furthermore, a microcontroller description file for the ATmega328 is already available in the EFL. We have connected peripherals such as the RS-485 driver, LED and button to the same pins on the microcontroller as in the experimental node, and again the EFL includes a board description file that covers this design. We can therefore simply delete from this file the initialization of the second LED, the second button and the expansion connector, none of which appear on the stepper motor driver board. Now we simply need to add the stepper motor driver as an extra peripheral block to the board description file and we will be in a position to write the code for our application in a hardware-independent fashion. However, since we have plans for further Elektor-

Figure 4.
Firmware can be written quickly with the help of the EFL software modules.
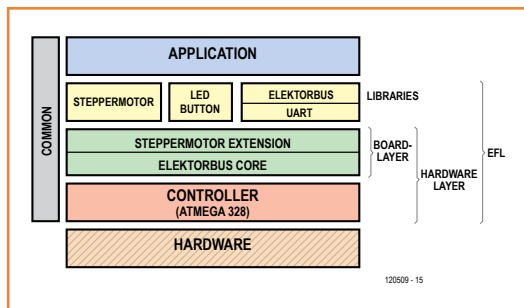
Figure 5.
The board is (virtually) divided into an ElektorBus core and an expansion module. The same board code can be used for all ElektorBus boards.
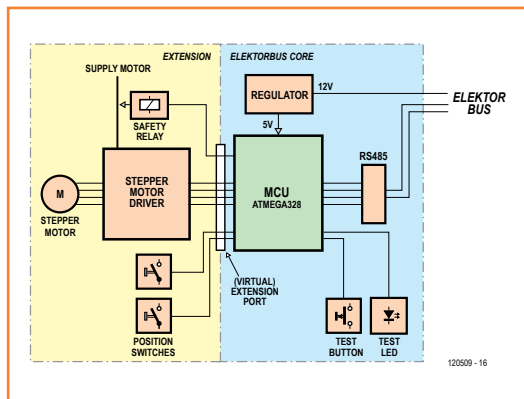
Bus boards offering other functions, we will take an even better approach here. Since all Elektor-Bus boards will include common hardware such as the RS-485 driver, test LED and test button, we encapsulate this in a single board description 'BoardEFL.h/.c' which we will keep in a subdirectory of the code base called 'ElektorBusCore'. The special functions of the board (such as in this example the stepper motor driver) are initialized separately in an extra file ('ExtensionEFL.h/.c' in the subdirectory 'ElektorBusStepperMotor'). This can be considered as equivalent to equipping the ATmega328 with a virtual expansion port to which the stepper motor driver is connected. **Figure 5** illustrates this idea.

Stepper motor code
The source code for this project can be downloaded at [2]. The EFL project in this example is called ElektorBusStepperMotor, and a click on ElektorBusStepperMotor.atsuo opens it in Atmel Studio. The project directory 'Hardware' contains the board description files BoardEFL.h/.c which initialize the core ElektorBus functions and the (virtual) expansion connector. The extension file includes the function Extension_Init(), which is

called whenever the application starts up. This function creates a peripheral block for the stepper motor driver. Three entries are added to the board pin table (for more information on this see the additional EFL documentation at [6]) for the expansion port corresponding to the signals EN, CLK and DIR. The MODE signal is not controlled by the software. We also allocate a block for the safety relay and a block for the two position switches. This can all be done in a way that is entirely independent of the underlying hardware: we do not need to know which microcontroller pins are involved, only the type and number of the peripheral block.

Although we are now in a position to control the EN, CLK and DIR signals without knowing the wiring of the board, that is not enough to guarantee that every stepper motor driver can be controlled in the same way using those three signals. To avoid the application developer having to delve into datasheets and to maintain the hardware-independence of the application code, it is necessary to implement a few more low-level functions for controlling the stepper motor in the extension file. So for example to set the direction of rotation we can use a call such as

```
void StepperMotorDirection(uint8
StepperMotorBlockIndex, uint8 Direction)
```

where the parameter `Direction` is 0 or 1, and to execute a single step we can call:

```
void StepperMotorStep(uint8
StepperMotorBlockIndex, uint8
MillisecondsDelay)
```

This latter function pauses program execution. For more demanding applications we have implemented a function that drives the stepper motor under control of a timer. This function takes an additional array as a parameter to specify the required speed ramp. The motor starts slowly and then the speed gradually increases to a specified maximum value. Towards the end of its travel the ramp is reversed and the motor comes to a gentle halt. (Note that in this mode it is not possible to read the state of the position detection switches.)
First we initialize the timer:

```
void StepperMotorTimerSetup(uint8
```

```
StepperMotorBlockIndex, uint8
MillisecondsDelay)
```

where we provide a delay in milliseconds per step which sets the basic speed of the motor. The function will make use of the lowest-numbered unused 16-bit timer.
Then a call to

```
void StepperMotorTimerSteps(uint8
StepperMotorBlockIndex, uint16 Steps,
uint8* RampData, uint8 RampMax, uint8
StepsShiftForNextRampIndex)
```

will start the motor running under control of the timer. The parameters `RampData` and `RampMax` describe the ramp. A value of 128 in this array stands for the basic motor speed, and smaller values represent higher speeds. The variable `StepsShiftForNextRampIndex` determines after how many steps the next speed value in the array will be used. The number of steps is two to the power of this value, so for example a value of 2 means that four steps will be executed at each speed value. After all the values in the array have been used the motor will continue to run at constant speed, and then at the end of the motion the array's values will be used again in reverse order.

The low-level functions in the hardware layer save the application developer a good deal of work. All that remains to be done is to write a function to read the position detection switches, and then perform any necessary calibration of the motor movements. We have included code for these functions in a small stepper motor library, which sits on top of the low-level interface described above. As always with the EFL the code can be used to drive several peripheral blocks of the same type, in this case up to eight stepper motor drivers on one board. The source code (StepperMotorEFL.c) is in the 'Libraries' project directory. The Doxygen documentation in the download archive [2] provides an overview of the functions available.

A quick demonstration
To test and demonstrate the design we built a simple set-up using a Nanotec 12 V stepper motor type SP2575M0206-A, a position indicator and two limit switches (see **Figure 6**). The stepper motor board was connected over the ElektorBus
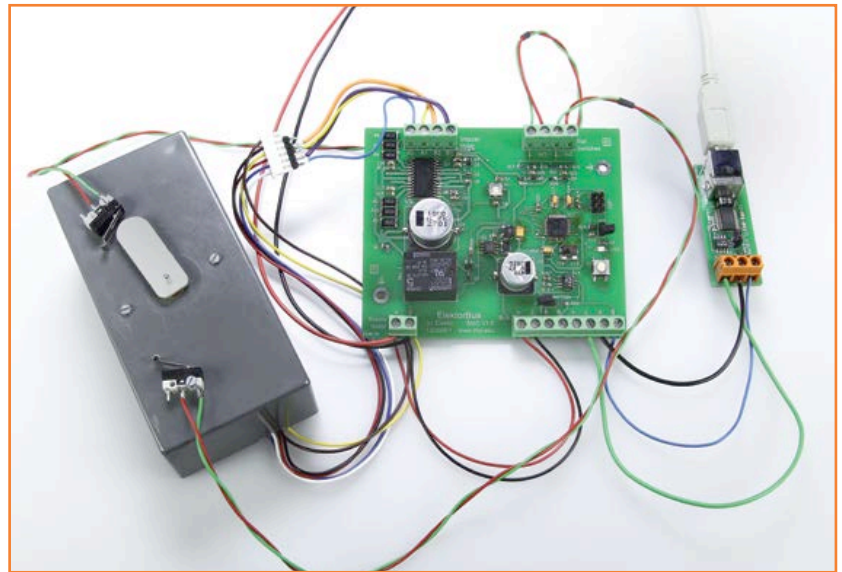
bus to the familiar USB-to-RS-485 converter and thence to a PC.
The application code proper is located in the main file of our project, ElektorBusStepperMotor.c. The customary EFL main function and the initialization of the LED and button, UART interface and ElektorBus libraries in the function `Application-Setup()` are described elsewhere [6][7].

With a call to

```
StepperMotor_LibrarySetup(SwitchEventCall
back, 0, 0);
```

we initialize the stepper motor driver library. The first parameter names the function in the application code which is to be called when one of the limit switches is actuated. In this case we have implemented this function (a little further down the main program file) so that it toggles the LED on the board. The remaining parameters, both zero here, give the block number offset between the stepper motor driver and the position detection switches and safety relay. In our example switch block number zero and relay block number zero are associated with stepper motor driver number zero, although in principle this could be different on a different board.

When the user presses the button on the board an automatic calibration is carried out. The motor first runs in one direction until a position detection switch is actuated; then it reverses and runs in

Figure 6.
Demonstration with position detection switches. The stepper motor board is connected to the PC using a USB-to-RS-485 converter.

the other direction until the other switch is actuated. The number of steps between these two positions is recorded. The library function StepperMotorCalibration() can now determine in which direction the motor turns in response to the direction signal being set low or high, and again this information is recorded.

Once calibrated the motor can be moved to a given desired position within its range of movement using a value from 0 to 1023, which represent the end positions.

For example, the call

```
StepperMotor_GotoMotorPosition(0, 512, 4);
```

causes the motor to move to its mid-position at a rate of approximately 4 ms per step.

Naturally we would like to be able to control the motor over the ElektorBus. It is easy enough to write an HTML control panel that runs in the ElektorBus browser on the PC, and as usual the same user interface can also be run on an Android smartphone or tablet [8][9]. The HTML file can be found in the directory UIBus, and it can simply be dragged from the download directory to the desktop. The rest of the procedure will be familiar: launch ElektorBusBrowser.exe, select the correct COM port, click on 'Connect' and fire up the scheduler. The HTML buttons cause Ele-

COMPONENT LIST

Resistors
(default shape: 0805)
R1 = 0 Ω
R2,R12,R15 = 39kΩ
R3 = 680Ω
R4 = 100Ω
R5,R23 = 100kΩ
R6-R11 = 1.0Ω (Vishay CRCW25121R00FKEG)
R13,R19,R20 = 10kΩ
R14 = 10Ω
R16,R18 = 0Ω (see text box)
R17,R21 = not fitted (see text box)
R22 = 120Ω
R24,R25 = 10kΩ, optional (see text box)
R26 = 4.7kΩ
P1 = 10kΩ trimpot (POT4MM-2)

Capacitors
(default shape: 0805)
C1 = 68µF 10V tantalum (AVX TPSB686K010R0600)
C2 = 470µF 35V electrolytic (Panasonic EEEFK1V471AQ)
C3,C5,C8,C12 = 10nF
C4 = 1000µF 50V electrolytic (Panasonic EEVFK1H102M)
C6 = 68nF
C7 = 220nF
C9 = 5.6nF
C10,C11 = 1nF
C13,C14,C17 = 10µF 6.3V (AVX TCJA106M006R0300)
C15,C16 = 100nF
C18,C19 = 22pF

Inductors
L1 = 47µH (744773147)

Semiconductors
D1 = B160-13-F, Schottky diode 1A / 60V
D2,D3,D6 = Diode 1N4148
D4,D5 = BZX384-B5V1 5.1V zener diode
T1 = BC849B, SOT-23
LED1 = LED, green (5988270107F)
IC1 = LM2675M-5.0
IC2 = L6208D (SO24)
IC3 = ATmega328P-AU
IC4 = LT1785CS8

Miscellaneous
JP1 = 6-pin (2x3) pinheader, 0.1'' pitch
JP2,JP3 = pinheader, SIL, 0.1'' pitch
K1-K9 = PCB screw terminal block, 0.2''
Rel1 = relay, SPDT, Omron G5LA145DC
S1 = pushbutton, Omron B3S-1000
X1 = 16MHz quartz crystal, 50ppm, 16pF, Epson Toyocom FA-365
PCB 120509-1
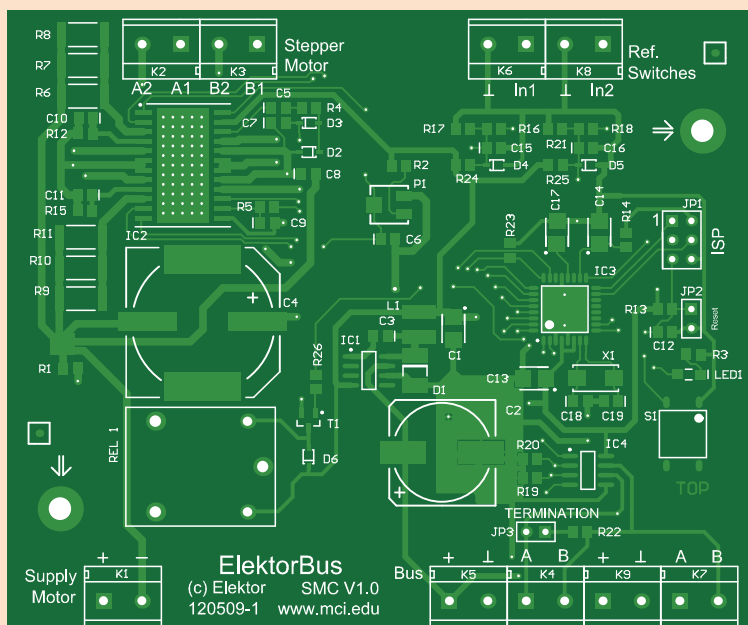
ktorBus messages to be sent from the PC to the stepper motor board, and the desired motor position can be sent as a 10-bit value on channel 0. In the firmware the ElektorBus library receives these messages and calls the function Process-Part(). This processes the incoming message parts and then causes the motor to move the the desired position.

As always this demonstration software is intended as motivation for further experimentation. A practical application might be the automatic control of a roller or Venetian blind to maintain the level of ambient lighting in a room. We have previously described hardware and software for an ElektorBus-based light sensor [6][10].

In the next issue we will take things further with the long-awaited Xmega web server board, which also features an RS-485 interface and which will make an ideal basis for further ElektorBus applications.

(120509)

Internet Links

[1] www.mci.edu

[2] www.elektor.com/120509

[3] www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00002294.pdf

[4] www.elektor.com/110258

[5] www.elektor.com/elektorbus

[6] www.elektor.com/120668

[7] www.elektor.com/130154

[8] www.elektor.com/110405

[9] www.elektor.com/120097

[10] www.elektor.com/110428

Wideband 70-cms FM Exciter
With 130 mW output power
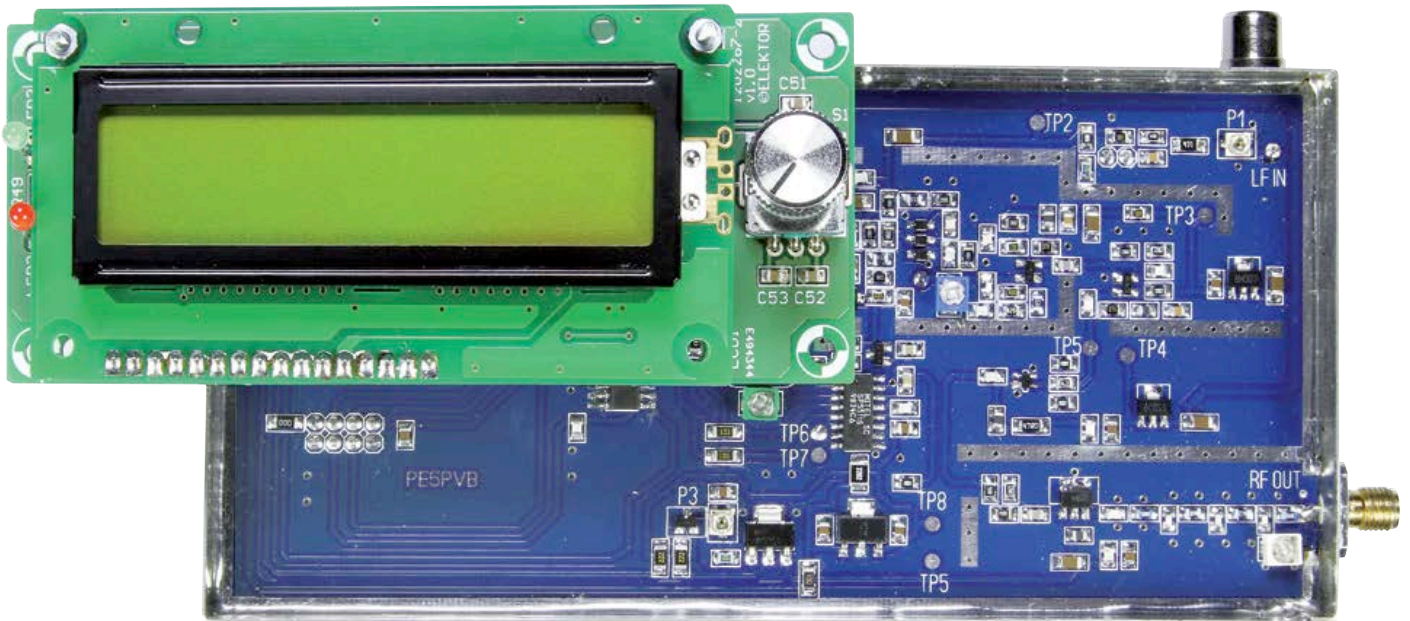
By **Sjef Verhoeven, PE5PVB**

(The Netherlands)

Most amateur radio transmitters are not actually designed for transmitting wideband audio signals with high fidelity. However, the wideband 70-cms (430–440 MHz) FM exciter described here is up to the task, with an audio bandwidth of 20 Hz to 100 kHz.

Technical Data

- Frequency range 430–440 MHz in 25 kHz steps
- Audio bandwidth 20 Hz to 100 kHz
- PLL lock time less than 1 s
- Supply voltage 12–15 V
- Current consumption approx. 250 mA at 130 mW RF output power

Just about everyone has played with walkie-talkies at some time. They're nice for wireless conversations, but radio links of this sort suffer from poor sound quality and the fact that only one person at a time can talk. Licensed radio amateurs ('hams') have been communicating in this manner since the early days. With the right choice of equipment, frequency and antenna, radio amateurs can connect to each other all over the world in various ways.

But what options are there if you want to establish a connection with good sound quality? In this article we describe the design of a complete broadband FM audio exciter for the 430–440 MHz section of the 70 cms band (in the US, the band extends from 420-450 MHz). The output signal is not unlike you find in the VHF FM broadcast band, which means that the audio bandwidth is

approximately 20 Hz to 100 kHz. Thanks to this large audio bandwidth, it is fairly easy to set up audio chat sessions with several transmitters and relatively low distortion. This is similar to Teamspeak or other chat tools, but without jitter, codec distortion, echo and delay. With the right antennas, antenna height and possibly an amplifier, you can easily make QSO's (connections) over a distance of tens of miles, or even as much as 200 miles under the right weather conditions.

An amateur radio license of the appropriate class is needed to operate this exciter. You can obtain this license from the relevant national or local authorities after passing a technical exam. The aim of this is to prove that you have sufficient theoretical knowledge to be able to operate radio transmitting equipment safely, in compliance with statutory provisions, and without causing interference. Among other things, this means that using this exciter as a transmitter to broadcast music for anyone who wants to listen is not permitted. The technical exams for amateur radio licenses are usually administered several times a year in various places. In many countries, volunteers provide ham radio training. In the US, see www.arrl.org/licensing-education-training. In the UK, http://rsgb.org/main/faq/how-to-become-a-radio-amateur-faq/.

You can use a scanner or a receiver with wideband FM capability to receive the signal. A downconverter, which converts the 430–440 MHz band to a different band such as 90–100 MHz, is often used for this purpose. This allows you to use an ordinary FM radio to listen to the signal.

Schematic

We opted for a Colpitts oscillator to generate the RF signal for the exciter. It is built around a dual-gate MOSFET (T2). This oscillator operates at the transmit frequency. In a Colpitts oscillator, the frequency is determined by a parallel resonant circuit consisting of a capacitive voltage divider and a coil. In this case capacitors C23 and C25 form the voltage divider and L4 is the coil. The coil consists of a short piece of thin coax cable, which essentially works the same way as a stripline. Modulation due to mechanical vibration can be largely avoided by using relatively flexible cable, such as RG174. Feedback is necessary to make the circuit oscillate, and this is provided by resistor R25. Many Colpitts oscillator circuits use direct feedback without a resistor, but practical

Construction Tips

- Use only ceramic capacitors, even for values above 1 µF.

- Most resistor and capacitor packages are 0805, but a few are 1206.

- If you make your own PCB, ensure that it has enough vias. This is particularly important in and around the RF section.

- Use inductors **without** ferrite cores. This is especially important in the oscillator stage, since ferrite is a known cause of noise.

- Fit the entire circuit in a sheet metal enclosure. The PCB layout is designed with standardized dimensions.

- First prepare the sheet metal enclosure (holes for DC, audio and RF connectors). Then place the bare board in the enclosure and solder it all around on both sides. Fit the components only after the board has been fully mounted in the enclosure.
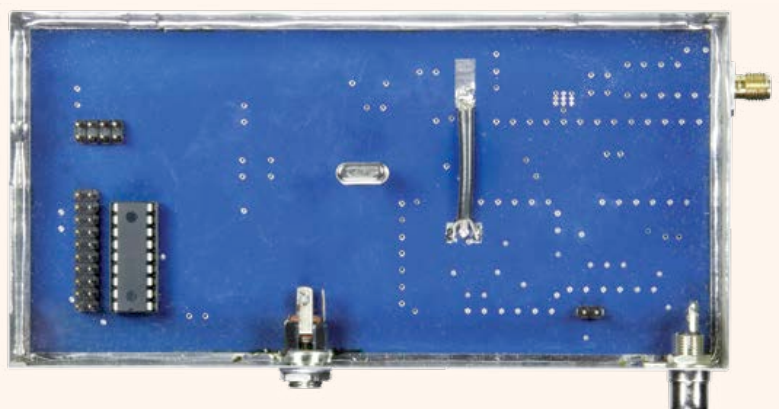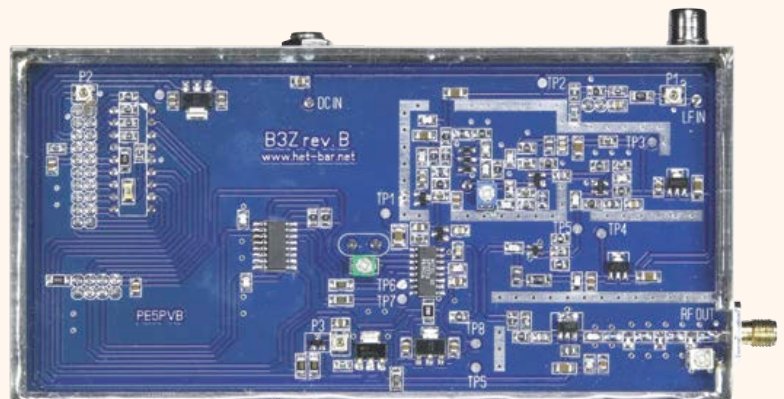
- Power the exciter from a clean supply voltage. Some switch-mode power supplies have frequency components on the output voltage that are directly audible as noise or whistles. A conventional power supply is always the best choice.
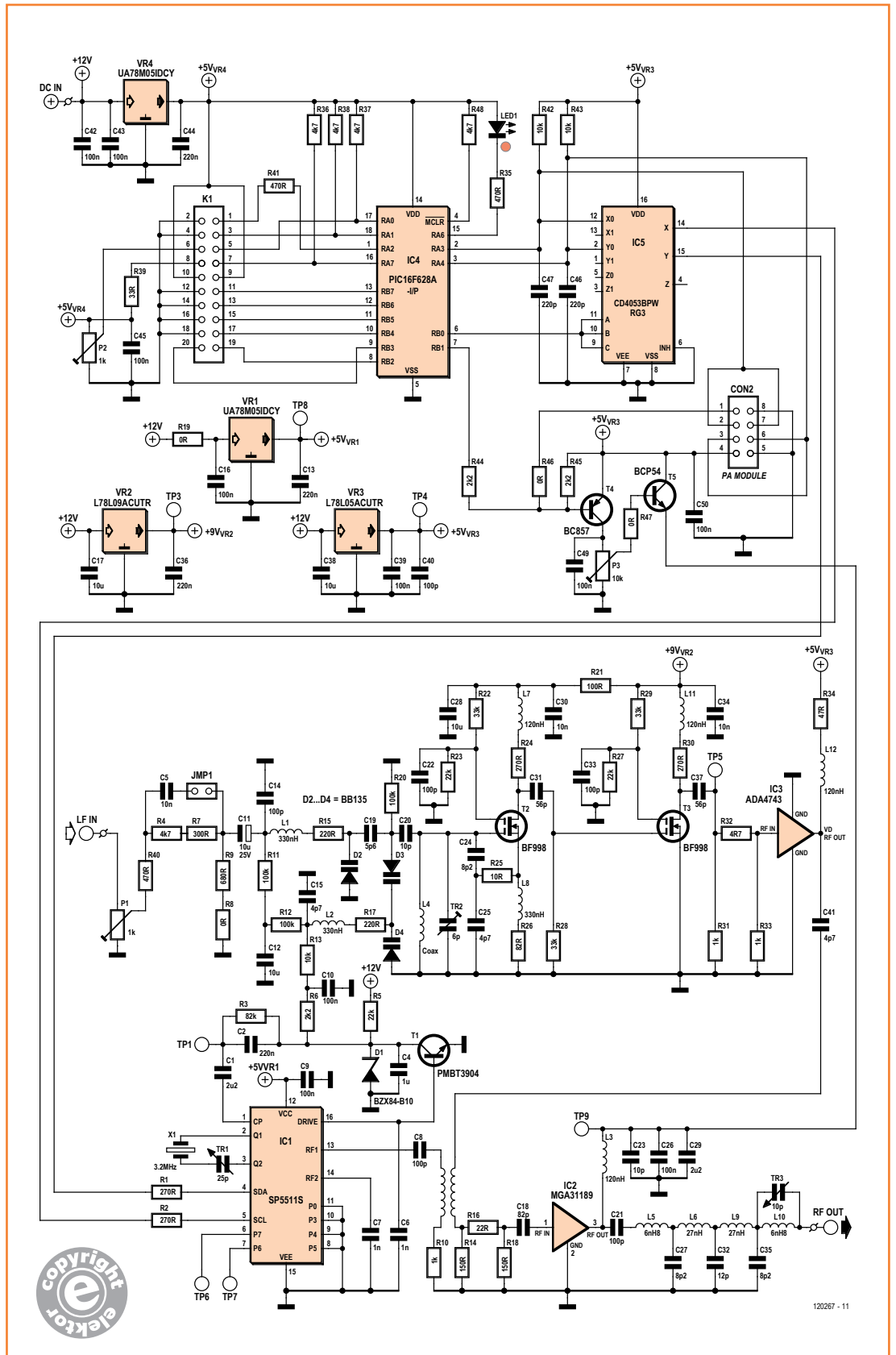
Figure 1.
Schematic of the main circuit board with the microcontroller and RF section.

experience shows that a bit of resistance makes the circuit more stable, with a cleaner frequency. Trimmer TR2 is provided to allow the oscillator to be adjusted to the desired frequency range. Diodes D3 and D4 are varicaps driven by a control voltage. This allows the frequency to be tuned electrically over a range of approximately 30 MHz.

Since this is an FM exciter, we need to be able to modulate the frequency of the carrier signal according to the amplitude and frequency of the audio signal. This is done with the aid of varicap diode D2. Varying the voltage on this diode affects the capacitance of the LC circuit, which causes a frequency deviation. To keep the modulation as linear as possible, we opted to use different varicap diodes for tuning and modulation. To avoid excessive differences in modulation depth between low and high frequencies, the modulation varicap is biased by a high-impedance voltage divider consisting of R11 and R12, with C12 added to improve stability. Inductor L1 prevents excessive capacitive load from the modulation circuit, which could otherwise cause the oscillator to stop working. Capacitor C14 provides HF decoupling for the modulation input and additionally ensures that the upstream part of the modulation path does not affect the resonant frequency of the oscillator circuit. Finally, components C5, R4, R7, R8 and R9 provide (defeatable) pre-emphasis. If necessary, you can increase the value of R8 as desired to reduce the pre-emphasis attenuation. The design is dimensioned for a pre-emphasis of 50 µs.

A buffer stage at the output of the oscillator, which is also implemented using a dual-gate MOSFET (T3), keeps the loading of the oscillator output as uniform as possible to maintain the stability of the RF signal.

The RF signal, at a power level of approximately 1 mW, is attenuated slightly to provide a good match to the next amplifier stage. In recent years the market has been flooded with a large variety of monolithic microwave integrated circuits (MMICs). These devices are specifically designed to amplify signals in the microwave frequency range. The main advantage of these ICs is that they make it easy to amplify RF signals without using tuned circuits. This avoids additional circuit complexity, cuts down on board space, and considerably reduces the cost of the overall cir-

cuit. For our design we chose MMICs from Avago Technologies, in part because they are a good fit with the circuit and in part because it's important to choose MMICs that are readily available. The first MMIC (IC3, type ADA4743) amplifies the RF signal to about 40 mW. This signal is attenuated and then fed to an MGA31189 (IC2). This MMIC has a rated output power of 250 mW. However, in this circuit we limit the supply voltage of IC2 to a maximum of 4.2 V (provided by the combination of T4 and T5), so the output power is a good deal lower—about 120 to 130 mW. This is a nice power level for driving a final amplifier, which in many cases will be an RF power module.

A phase locked loop (PLL) is used here to set the desired frequency and keep it tuned precisely. This involves passing the transmit frequency through a frequency divider and comparing the output signal with a reference signal. If there is any difference, a control voltage tries to adjust the frequency to reduce the difference until the divided-down RF signal and the reference signal are in phase. In our circuit this control voltage is applied to varicap diodes (D3 and D4) after filtering by a loop filter. The loop filter is essential and must be dimensioned to keep the PLL from responding too fast. This is because the oscillator is frequency modulated, so the frequency varies according to the applied audio signal. If the PLL responds too fast, the sound quality of the

Figure 2.
The operator control section includes a rotary encoder and a 2x16-character LCD module.
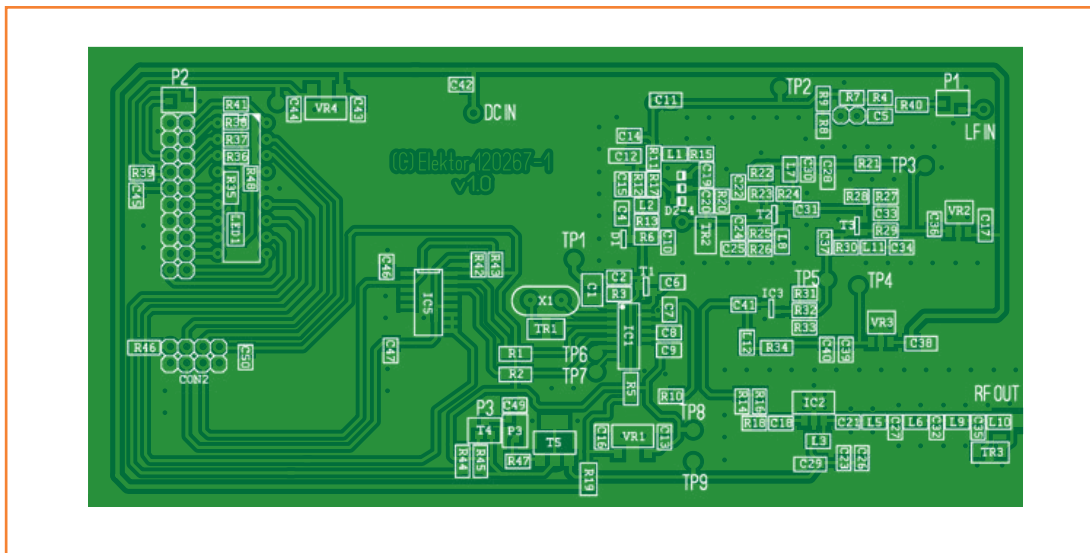
Figure 3.
The main PCB has components fitted on both sides.

modulated signal will be affected. It is therefore important to find a good balance between maximum bandwidth (approximately 180 kHz) and frequency stability. In this circuit we use an SP5511 or TSA5511 (IC1) for the PLL. These devices are readily available and affordable. These PLL ICs were originally intended to be used in TV tuners (including satellite tuners). They are often seen in older-model television, video recorder or satellite tuners. The smallest frequency increment this PLL can generate is 50 kHz with a 3.2 MHz crystal. However, what we want here is a step size of 25 kHz. This can easily be achieved by using the second harmonic of the RF signal as the input to the PLL. This signal is tapped off between IC3 and IC2 using inductive coupling. The PLL IC is controlled by a microcontroller (IC4, type PIC 16F628A) over an I²C bus. A major shortcoming of the PLL IC is that it generates crosstalk between the I²C bus and the charge pump and driver stages. This results in a ticking sound in the transmitted audio signal at the clock rate of the I²C data. The author's intention was to be able to control not only the exciter but also the power amplifier (if used) or other hardware over the I²C bus. The data for this, which in such case would not be intended for the PLL, would also be audible. In the present design this problem has been solved by placing an HEF4053 (IC5) between the I²C bus and the PLL IC. When the data on the I²C bus is not intended for the PLL, the I²C link to the PLL is blocked. This is a very low-cost and effective solution.
The exciter is controlled using a rotary encoder

with a built-in pushbutton. Along with two LEDs and the display module (see **Figure 2** for the schematic), it is located on a separate PCB which is connected to K1. In operation, you simply select the transmit frequency and then press the rotary encoder to confirm the setting. The corresponding data is sent to the PLL IC, and the PLL status is read out. When the PLL indicates that the phase difference between the divided-down RF signal and the reference signal has been reduced to zero, this information is fed back to the microcontroller. The microcontroller then lights up LED2 (the PLL Lock indicator). It also switches on the supply voltage to the final MMIC, so that the exciter only outputs an RF signal when it is actually operating at the selected transmit frequency. The supply voltage of the final MMIC can be adjusted with potentiometer P3, which allows the output power to be reduced to any desired level down to approximately 1 mW. This makes the exciter suitable for use with a wide variety of final amplifiers. The eight-pin header CON2 can be used to connect a external device to be controlled by the microcontroller, such as a final amplifier.

When the author began working on this design, he was not especially familiar with microcontroller programming. He started off with the PIC Simulator IDE. This is a very simple software development environment that uses its own version of Basic. The nice thing about this software is that it includes a complete simulator, so you can fully simulate the software before loading

it into a microcontroller. A drawback is the limited calculation options. To avoid making thing too complicated, the author chose to work with two different counters. One of these counters handles the frequency readout, while the other provides the divisor. In other words, if you raise the transmit frequency on the display by 25 kHz, the counter setting for the divisor is increased by 1. At the lower and upper ends of the frequency band covered by the exciter (430 MHz and 440 MHz), the counter for the divisor is preset to a default value.

All necessary information for this project (PCB layouts, components list, source code and hex code files for the microcontroller) is available on the Elektor website [1].

Operation

The entire exciter is operated using a single rotary encoder with a built-in pushbutton. You select a new frequency by turning the encoder knob. When you do this, "TUNE" appears in the top left corner of the display. When you press the encoder knob, the new frequency is saved in memory and the exciter is tuned to this frequency. When the PLL is locked, the Lock LED lights up and the output stage is switched on. If the PLL cannot achieve lock, the Lock LED remains dark and the indication "PLLERR" is displayed. If the PLL does not receive any data over the I²C bus, the indication "I2CERR" is displayed. In both cases the microcontroller will try to keep trying to configure the PLL until it achieves lock.

Construction

When building RF circuits, especially at relatively UHF, it is important to keep all connections as short as possible. The ground level should also be kept as 'cold' or 'earthy' as possible. That's why there are so many vias in the layout of the author's prototype PCB (shown at reduced scale in **Figure 3**). If you want to make your own PCB, it is very important to include these vias. The board is designed to be fitted in a sheet metal box measuring 74 x 148 mm, which is a commercially available size. The board must be soldered to the sheet metal of the box along all edges on both the top and the bottom sides. This provides an optimal ground connection. Fitting the board in a sheet-metal box and soldering it all around are very important. A DIY box made from PCB material is not suitable for this circuit.

The construction procedure used by the author
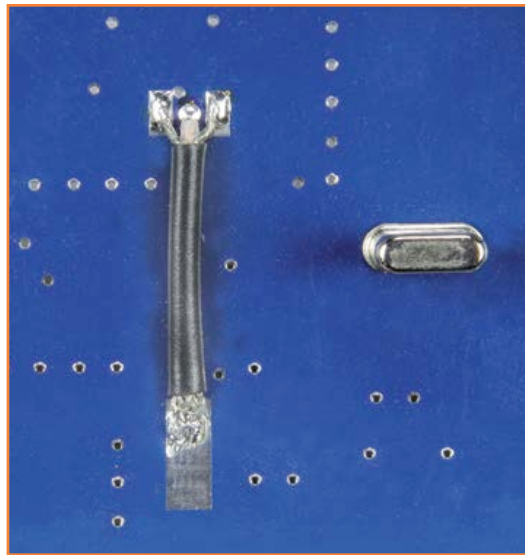
Figure 4.
Along with the microcontroller and several headers, a length of coax cable acting as the oscillator coil is fitted on the bottom side of the board.

is as follows. First clamp the bare board in the sheet-metal box and fit all necessary chassis components. An SMA chassis-mount connector is recommended for the antenna connection. If you use a bulkhead connector, it can easily be soldered to the sheet metal. This also eliminates the need for screws. The connectors for the supply voltage and the audio input can be fitted in any desired location. A Cinch (RCA) chassis-mount connector is a good choice for the audio connection.

All SMDs are located on one side of the board, while three pin headers, the coax stripline and the microcontroller are on the other side. **Figure 4** shows a detail view of the stripline. It consists of a length of flexible coax cable measuring 4 cm, with both ends soldered to the PCB. At one end the center conductor and the braid are joined together and soldered to the PCB. At the other end the braid must be divided in two and soldered to the ground plane as shown in the photo, with the center conductor soldered to the pad between these two points.

After this you should assemble the circuit section by section. Start with the oscillator. Check that it oscillates in the 400–500 MHz range. You can use a frequency counter or a spectrum analyzer for this. Then fit the components for the buffer stage and the two MMIC amplifier stages. Provide a temporary supply voltage connection for the final MMIC so that it also operates. Check the RF signal with an RF millivoltmeter or other suitable instrument.
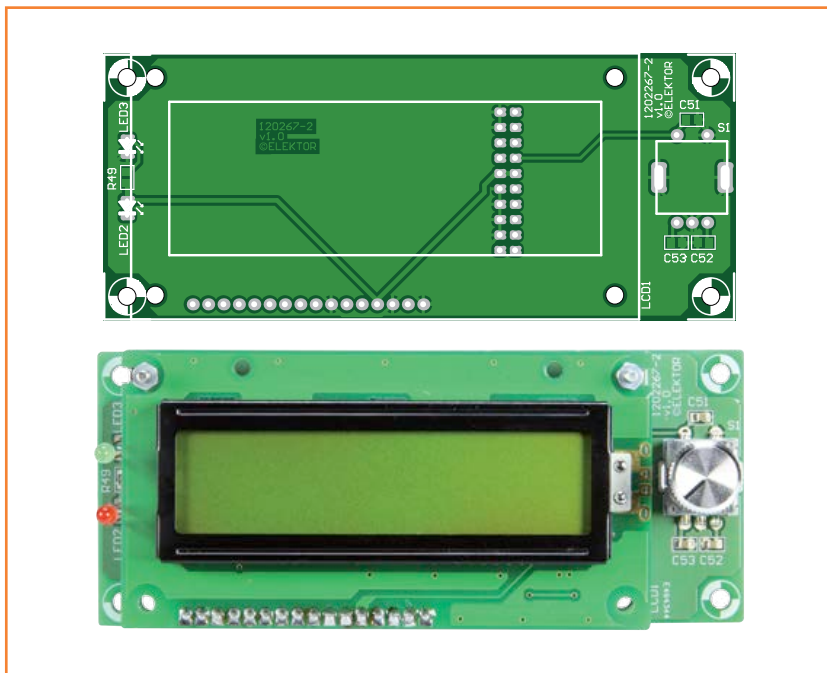
Figure 5.
The control board is fairly simple and easy to build. However, note that the 2x10-pin header for the connection to the main board must be fitted on the rear side.

If the RF portion works properly, you can then assemble the PLL and microcontroller portions of the circuit. Use an 18-pin socket for the microcontroller, and program the microcontroller before inserting it in the socket.

The operator controls are located on a separated PCB as illustrated in **Figure 5**. Assembly of this board is straightforward. All components (except one) are fitted on the side with the component overlay. You can use a 16-pin header with a matching connector to connect the display module. On the rear side of the board, fit a 20-pin header for connection to K1 on the main board with a 20-lead flat cable.

Adjustment
• Connect the exciter (with the display) to the supply voltage. Adjust P2 for the best display contrast.
• Then adjust the frequency to 435 MHz. Using a ceramic or plastic trimmer tool, slowly rotate TR2 until the transmit frequency is 435 MHz. Measure the voltage at TP1 and turn TR2 until the measured voltage is approximately 6 V. Now you can use a frequency counter or spectrum analyzer to set the transmit frequency to exactly 435.0 MHz with TR1.
• Then adjust P3 for maximum RF power. Next you have to adjust the notch frequency.

A spectrum analyzer is necessary for this. Measure the second harmonic at 870 MHz and adjust TR3 for minimum power. If you do not have a spectrum analyzer, you can use an RF millivoltmeter and adjust TR3 for maximum output power. However, you should bear in mind that filtering will be suboptimal in this case. This must be dealt with in the power amplifier connected after the exciter.
• Check that the exciter works properly at both 430 MHz and 440 MHz. Fit a jumper in the pre-emphasis jumper position and connect an audio source, such as a CD player. Then adjust P1 for optimal audio amplitude without distortion.

The 70-cm FM exciter is now ready for use. We realize that some radio amateurs do not have a lot of experience with soldering SMDs by hand. If there is sufficient interest, the author is willing to assemble and adjust a number of boards at a reasonable price. If you are interested in this, please contact the author directly at [2].

(120267-I)

Internet Links
[1] www.elektor.com/120267
[2] pe5pvb@het-bar.net

Adjustment Points

P1	Modulation depth (audio)
P2	LCD contrast
P3	Output power
TR1	Fine tuning (reference oscillator)
TR2	Coarse tuning (RF oscillator)
TR3	Second harmonic filter notch frequency

Test Points

TP1	PLL control voltage
TP2	Modulation signal after pre-emphasis
TP3	Oscillator voltage (9 V)
TP4	Driver stage voltage (5 V)
TP5	Output stage voltage (0–4 V)
TP6	PLL divider output frequency
TP7	PLL reference frequency

Smartphone A/V Remote Control

Transmitter plus App for Android devices
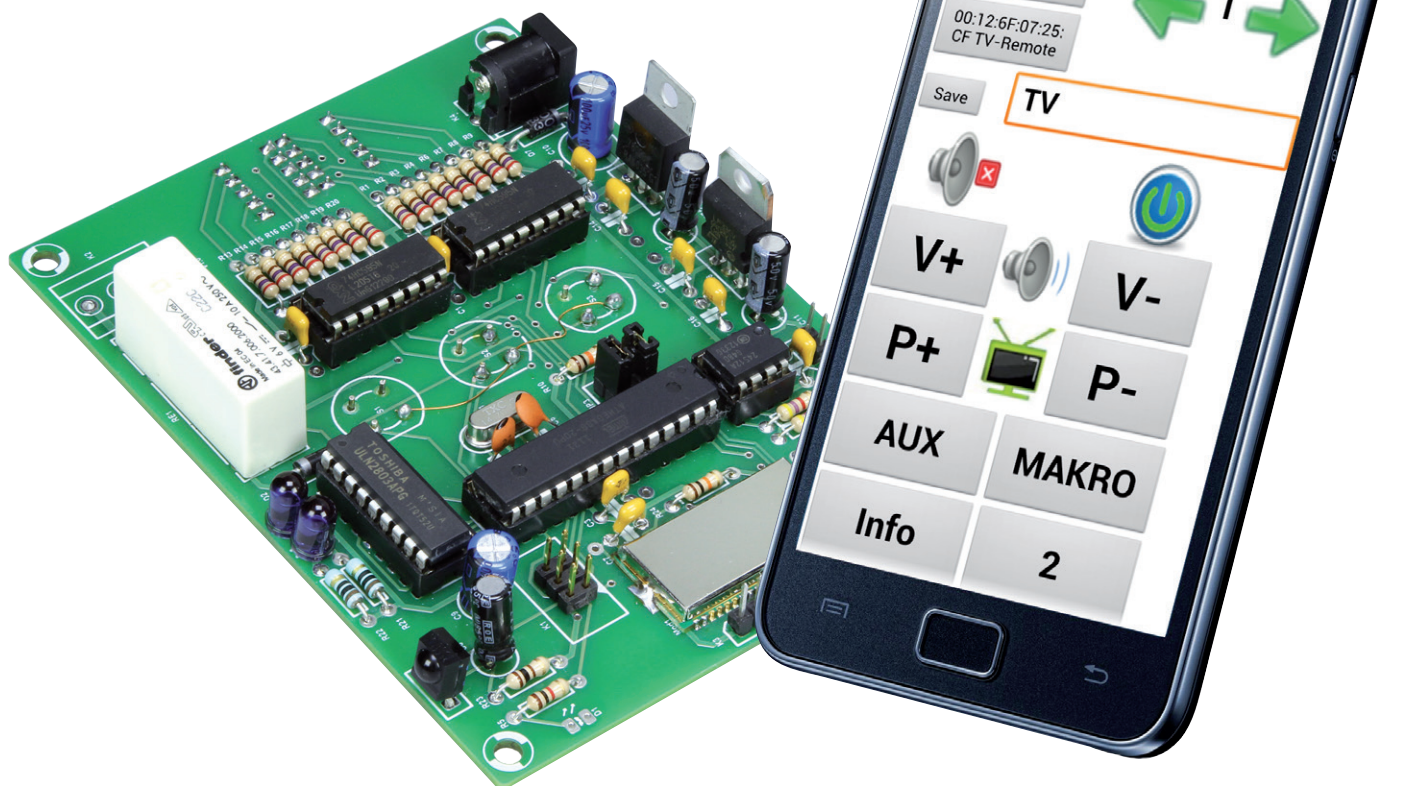
By **Peter Zirngibl**
(Germany)
(info@pezitec.com)

The best types of universal remote controller come equipped with a large touch screen, just like the one on your Smartphone. So why not use your phone to control all the A/V appliances at home?

Armed with a Smartphone you can surf the Web, send emails, text, chat, download and listen to music, take and post photos and video clips, listen and view TV stations, navigate, place bets and yes, even make phone calls to anywhere on the globe. Tens of thousands of Apps are also available to make the Smartphone the most universal, configurable control and communication device ever created. It is not surprising that there are also Apps which allow you to control the newer generation of home WLAN-equipped A/V equipment. The WLAN interface allows signals from the phone to be routed through to control the equipment directly without the need for additional hardware. Older A/V equipment generally only have the more traditional IR remote control interface and Smartphones still have no built-in IR transmitter.
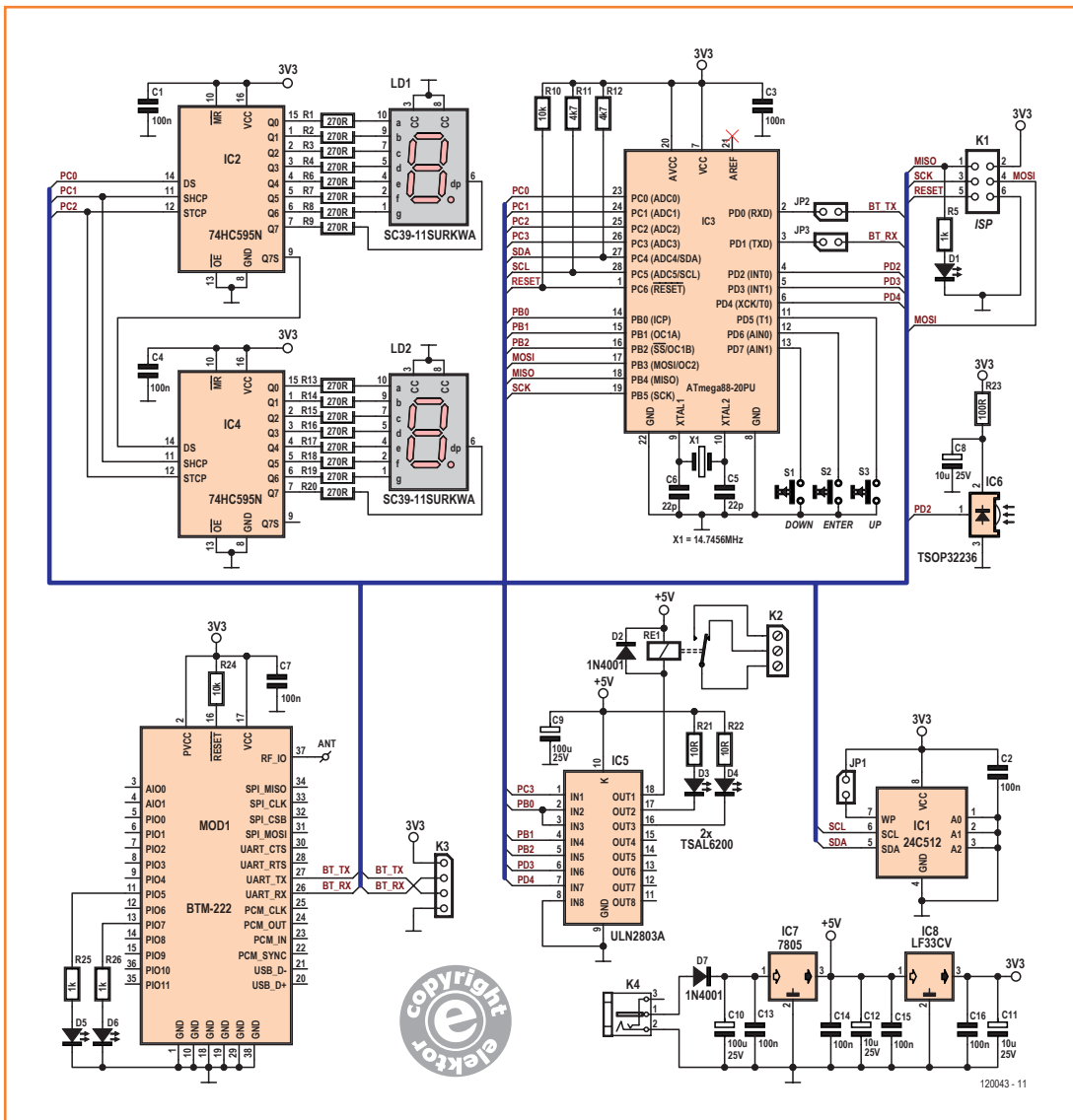
Figure 1.
The high level of integration in the Bluetooth module reduces circuit complexity.

The vital bit of kit missing here is an intelligent adapter that can on one side communicate with the phone using Bluetooth and on the other, transmit IR signals to control the A/V equipment. Intelligence, in this context generally means that a device has the ability to 'learn' a sequence of commands. This of course implies that a learn process is necessary. The adapter needs to know the particular variant of remote control that is in use and which button controls which feature of the controlled equipment. Simple, low cost universal remote controllers are pre-programmed with thousands of different command sets for all the different makes of equipment. They are generally only able to control a few basic commands like PLAY or STOP. It is not possible to

edit the commands and a learn function is only possible with the more expensive types of universal remote controller.

Some of the better types are able to execute macros which involve a sequence of commands (e.g. Watch a DVD = switch on DVD player, TV, AV receiver, then configure the correct video and audio channel). This procedure is also possible with the Bluetooth-IR adapter described here. The newer, better quality (and more pricey!) type of universal remote controllers can now be conveniently programmed using a PC. To cut down on the software expenditure for this project we decided against this approach. In this design the adapter is programmed by reading the IR signals directly from the original remote controller.
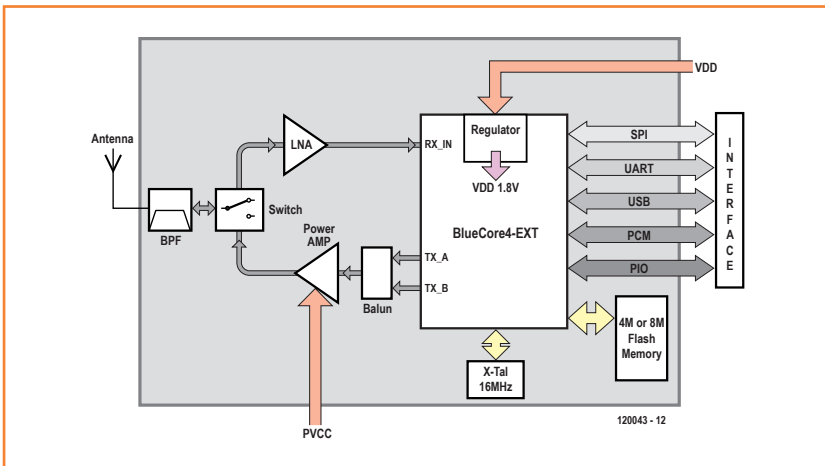
Figure 2.
Block diagram of the BTM-222 Bluetooth module.

Figure 3.
A terminal program is used to set up the Bluetooth module to the displayed parameters.

This means that the IR remote transmitter is placed in front of the IR adapter's receiver and the command transferred. How this is exactly accomplished will be covered in more detail later, first we will take a closer look at the hardware.

A radio-linked controller
The circuit diagram given in **Figure 1** indicates that there really isn't too much hardware used in the adapter design. It basically consists of a small Atmel type ATmega88 microcontroller together
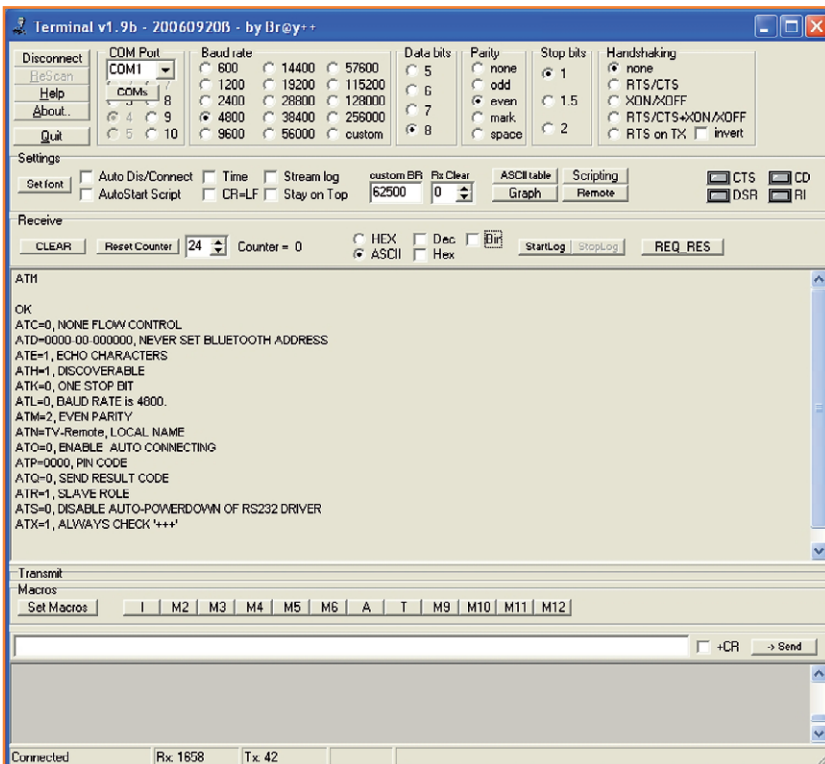
with some peripheral components:

• A 24C512 EEPROM with 512 kBit memory and I2C interface to store the programmed codes.
• An infrared receiver module type TSOP32236
• Two IR transmitter diodes type TSAL6200 driven by a ULN2803 driver chip.
• A two character 7 segment display with an HC595 8-bit shift register to show the programmed codes and program status.
• Three pushbuttons UP, DOWN and ENTER, for programming and operating.
• In-System-Programming interface K1.
• A relay with changeover contacts driven by the peripheral driver chip.
• Voltage regulators for +5 V and 3.3 V

You have probably seen all of these features many times before in many other circuits, however the real star of the show here is the BTM-222 Bluetooth module from Rayson. A description of the module is contained in the data sheet [1]. The module is small, portable, easy to program and above all (around $10 a pop) affordable. In time, as we move to a more wireless future, all peripherals will probably communicate using devices like this. The BTM-222 is a class 1 device giving it a range of up to 300 ft. As you can see from its block diagram in **Figure 2** it includes many serial interfaces and from these the UART must be favorite for microcontroller communications. Data throughput of the UART (and the USB interface) in the BTM-222 is guaranteed at the full data rate of 921 Kbit/s. The BTM-222 UART used with a microcontroller requires hardly any additional components. The communication parameters are factory set to the standard 8N1 setting:

• Baud rate 19,200 Baud
• 8 data bits
• No parity
• 1 stop bit

The microcontroller only needs to open the corresponding UART channel. As required, parameters and other properties of the module can be accessed using the so-called 'AT command' set of instructions. More information on this aspect is contained in the BTM-222 data sheet [1]. The settings are stored in an internal flash memory. The 'blue' core of the module is clocked by an internal 16 MHz oscillator. The output signal is

fed to a balun and then to an RF power amplifier producing a signal of +18 dBm at the aerial. For signal reception the BTM-222 switches the received signal to a low-noise amplifier block (LNA) followed by a band-pass filter to reduce out-of-band signal interference. The BTM-222 does not require any external aerial; a short length of PCB track can be used. Here we have used a short piece of wire.

The module provides some status output signals: the data status (LED D5 on pin 11) and the connection link status (LED D6 on pin 13) are both used in the circuit. There is also an operating voltage status indicator output from pin 14 but we have not used it in this application.

BTM222 set up

Initially the Bluetooth module needs to be configured with the correct parameters. First remove jumpers JP2 and JP3 to disconnect communications with the microcontroller. Connect the BTM-222 to a PC using, for example a TTL-232R cable from FTDI, connected to K3. The cable carries a 5 V line (connecting to pin 1 of K3) which must not be allowed to make contact with the BTM-222 supply (this is at 3.3 V, and its upper limit of 3.6 V must not be exceeded). It is therefore important to leave pin 1 of K3 unconnected and power the Bluetooth module from the board power supply.

Run a terminal emulation program such Hyper-Terminal or Hterm on a PC (**Figure 3**), select a (virtual) COM port, to which the Bluetooth module is connected and use the serial port settings given above for the BTM-222 serial interface. Check to see if the module reacts to the `ATI` command; this should cause all the module settings to be dumped to the screen. If there is no response check that the 3.3 V supply, the COM port settings and the connections on K3 (TxD/RxD switched?) are in order. If you are unsure of the module settings or maybe suspect that the BTM-222 is not operating correctly you can pull pin PIO4 high for a minimum of three seconds to reset the module to its factory settings.

Change the UART settings to 4800 Baud (ATL = 0), even parity (ATM = 2), no handshake flow control (ATC = 0). Adjust the terminal emulator settings after each parameter change. Finally you can change parameters such as the module name or the Bluetooth connection PIN code as you wish.

Turn on the Smart phone (or other Android device), check that the BTM-222 has been detected and enter the PIN for the connection. When everything is in order turn the remote control unit off, disconnect the PC and place the two jumpers in positions JP2 and JP3.

The Remote App

Once the *Remote_Control.APK* App has been installed the Android Smart phone will be capable of remote control. Bring up the App to begin pairing with the BTM 222 module. Choose the Bluetooth device using SELECT DEVICE and use CONNECT to establish a connection to the device. LED D6 lights up indicating that the module is connected to an Android device and D5 flashes when data is being received.

The transmitted value (0 to 99) plus a line feed character tells the adapter which memory location is referenced. In playback mode the command received from the Bluetooth module can be evaluated. The principle is really simple: '2/n' is received and the adapter jumps to memory position 2, and transmits the command stored in this position (in this case turning up the volume of device 1). Commands 73 and 74 do not produce any IR signal but instead operate a relay. Command 73 closes the relay contacts for two seconds while command 74 toggles the relay contact state between off and on. Connector K2 provides connection to the contacts, allowing external equipment to be switched on and off. **Figure 4** shows the App running in all its glory on a Smartphone. With the left/right arrows at the top you can choose between five different devices. For each device you can choose to enter a name in the text box and store it with the Save button. The name is retained so it appears
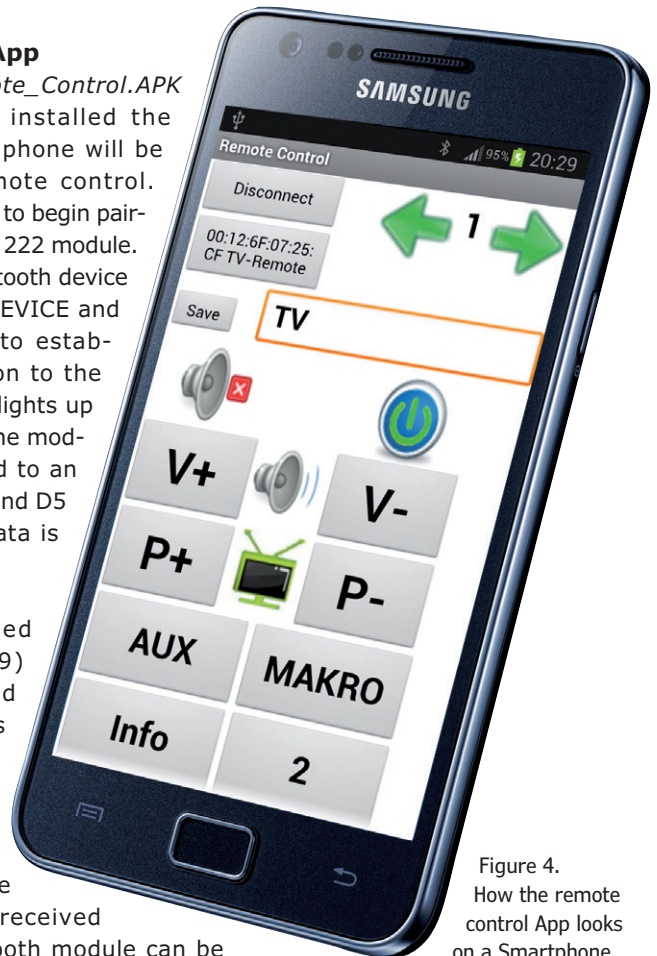
Figure 4.
How the remote control App looks on a Smartphone.

when the App is next started. In addition you can add a description to the functions 1 to 10. Press the EDIT button, choose the button to edit and enter the text. Press SAVE to store the text. This sequence is the same for every device.

Camera, action!

The controller software is divided into two parts; the programming mode and the operational mode. In programming mode the remote controller infrared command signals are recorded in the adapter memory in the following manner: First it is necessary to clear the serial EEPROM. This is accomplished by pressing all three buttons simultaneously and applying power to the adapter. When the seven segment displays show dE (delete) the three buttons can be released. The complete erase process takes a few minutes before 00 is shown in the display.

To record the commands in programming mode hold the ENTER button (pin PD6 to GND) before connecting the supply voltage. On the seven segment displays Pr should flash three times (Programming mode). Now find a memory position from 0 to 99 using the UP/DOWN buttons (pin PD5 and pin PD7) and select it using the ENTER button. The display will show the symbol − indicating its readiness. The remote controller can now be bought within 2 to 10 cm to the IR receiver and the required command button on the remote controller pressed once (once only!) Try to complete this process quickly to reduce the chance of recording some interference. The display should now show io.. as confirmation. Recording a macro is possible for each device, each macros can contain up to six commands. Macros are stored in the same way that single commands are stored.

Figure 5.
The circuit board layout. All of the user interface elements are located on the underside.

COMPONENT LIST

Resistors
R1-R4,R6-R9,R13-R20 = 270Ω
R5,R25,R26 = 1kΩ
R10,R24 = 10kΩ
R11,R12 = 4.7kΩ
R21,R22 = 10Ω
R23 = 100Ω

Capacitors
C1,C2,C3,C4,C7,C13,C14,C15,C16 = 100nF
C5,C6 = 22pF ceramic
C8,C11,C12 = 10µF 25V radial
C9,C10 = 100µF 25V radial

Semiconductors
D1,D5,D6 = LED, red, 3mm, low current
D2,D7 = 1N4001
D3,D4 = TSAL6200 (Vishay) 940nm IR transmitter diode
LD1,LD2 = SC39-11SURKWA (Kingbright) 7-segment-LED display, 10mm
IC1 = CAT24C512LI-G (On Semiconductor)
IC2,IC4 = 74HC595N
IC3 = ATmega88-20PU (Atmel), programmed, Elektor # 120043-41 [2]
IC5 = ULN2803APG
IC6 = TSOP32236IR (Vishay) 36kHz IR receiver
IC7 = 7805
IC8 = LF33CV (STMicroectronics)
Mod1 = BTM-222 (Rayson) Bluetooth module

Miscellaneous
X1 = 14.7456MHz quartz crystal

Re1 = 6V SPDT relay (1x c/o contact), Finder 43.41.7.006.2000
S1,S2,S3 = tactile switch, SPNO, round

When the IR signals are recorded the pulse widths are stored as *Unsigned integer Variables,* buffered and then saved to the EEPROM. To program another command it is necessary during the confirmation display (`io.` flashing) to hold the ENTER button after programming to generate a µC reset.

With no other button pressed during reset the program automatically jumps into playback mode. Now using UP/DOWN buttons you can select the command you have just stored and press ENTER to transmit it over the IR diode to switch the A/V equipment. This will show if the command was correctly stored.

On the PCB

A PCB for the remote control adapter is available from the Elektor shop [2], where you can also order a preprogrammed microcontroller. Use this link to find the compiled App and the firmware source code. A look at the PCB layout should allay any fears for the home constructor; although the Bluetooth module has an SMD outline connections are spaced at 1.25 mm and are relatively easy to solder by hand with a standard fine tipped iron. Correct placement of this module is important prior to soldering. Pin 1 can be identified by a small round dot on the metal screen (on the side by the aerial). This pin should be orientated near the PC1 label on the PCB. The thick pads along the long sides and at the ends are at ground potential and connected to the module's continuous earth plane.

Once the module has been soldered in position the other through-hole components can be fitted. It is recommended to mount the IC components

JP1,JP2,JP3 = 2-pin pinheader with jumper
K1 = 6-pin (2x3) pinheader
K2 = 3-pin PCB screw terminal block, pitch 5mm

K3 = 4-pin SIL connector
K4 = low voltage adaptor socket, 2.1mm
PCB 120043-1 [2]
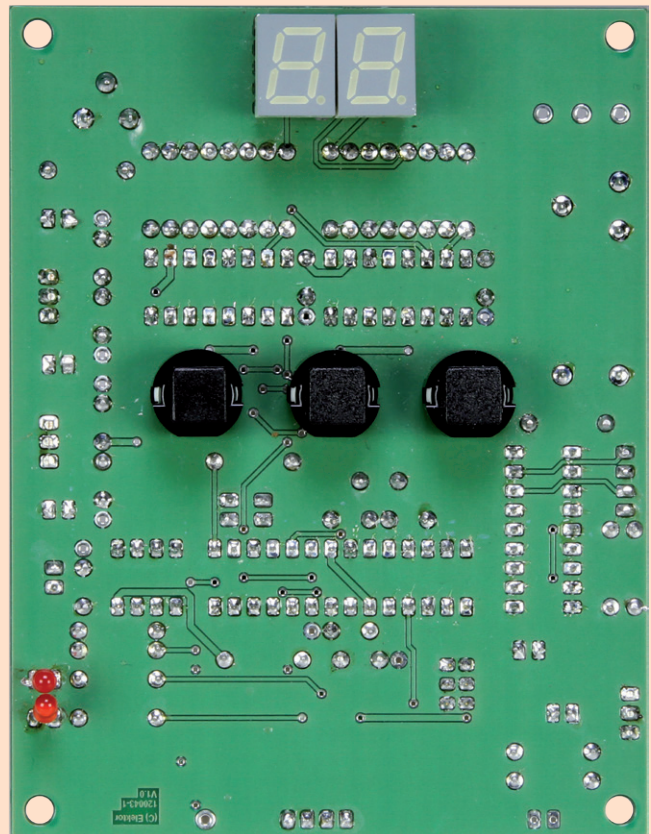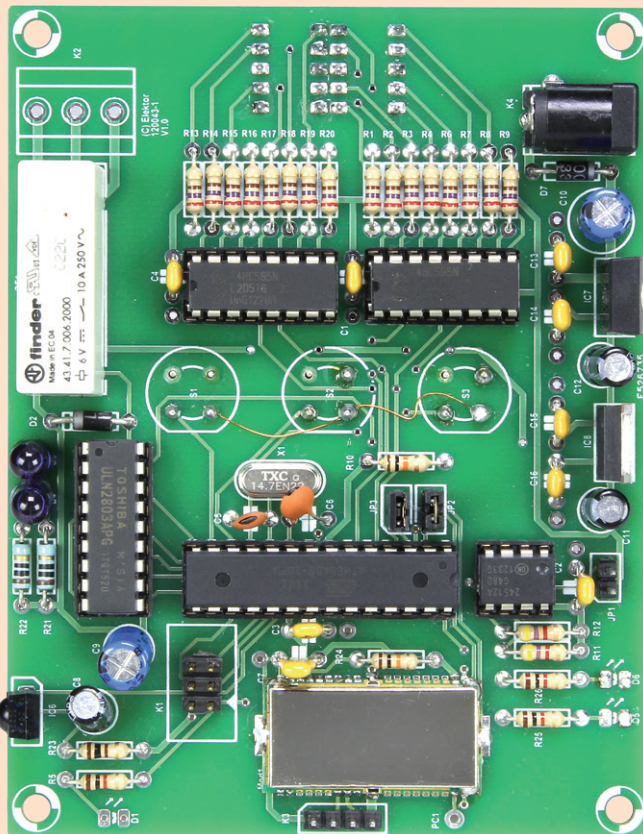
Table 1. Assignment of memory position and function					
Function	**Device**				
	1	**2**	**3**	**4**	**5**
MUTE	0	8	16	24	32
ON/OFF	1	9	17	25	33
VOL+	2	10	17	26	34
VOL-	3	11	19	27	35
PRG+	4	12	20	28	36
PRG-	5	13	21	29	37
AUX	6	14	22	30	38
MAKRO	7, 75-79	15, 80-84	23, 85-89	31,90-94	39, 95-99
Function 1	40	50	60		
Function 2	41	51	61		
Function 3	42	52	62		
Function 4	43	53	63		
Function 5	44	54	64		
Function 6	45	55	65		70
Function 7	46	56	66		71
Function 8	47	57	67		72
Function 9	48	58	68		73 [1]
Function 10	49	59	69		74 [2]
1) Relay on for 2 s 2) Toggle relay					

using sockets. Both seven segment displays, the three push buttons, D5 and D6 are fitted to the PCB underside. This leaves all the 'HID' devices on the same side of the board which makes it simpler to install into a case. A simple short length of cable is suitable as an aerial. According to the calculation at 2.4 GHz (13 cms band) the aerial should be around 3.1 cm (λ/4), but the length is not too critical and for this application we only need relatively short range communication. The main thing to remember is that if the unit is fitted in a metal housing please ensure that the aerial extends outside the case.

(120043)

Internet Links

[1] www.mikrocontroller.net/wikifiles/f/fc/
    BTM222_DataSheet.pdf

[2] www.elektor.com/120043

[3] http://appinventor.mit.edu

App Editing

The App was developed using MITs App Inventor [3]. To make changes to the App it is first necessary to create a Gmail account and then login to the web site [3]. Load *Remote_Control.ZIP* under 'My Projects' of the App-Inventor in 'More Options' -> 'Upload Source'. The App can then be edited and compiled on line and the resulting file '*Remote_Control.APK*' can be downloaded to the PC or installed directly to an Android device.

Ambience Lighting Controller
Setting the mood with RGB LEDs

By **Goswin Visschers**
(The Netherlands)
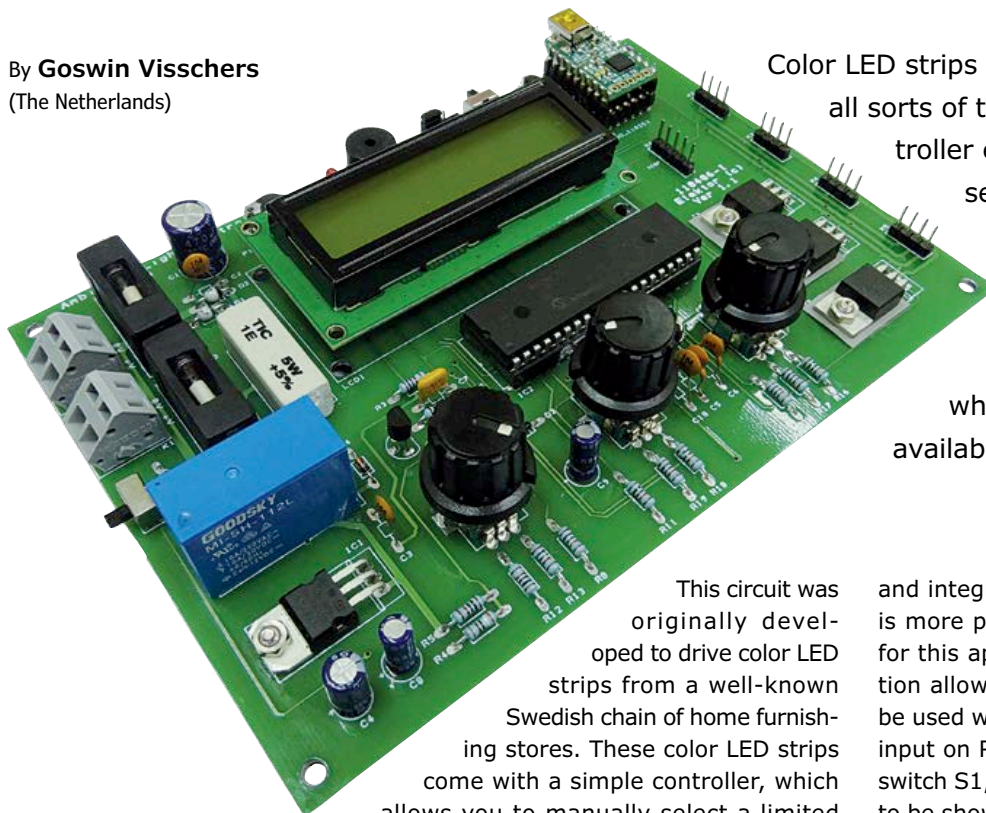
Color LED strips are now available at low cost in all sorts of types and sizes. With the controller circuit described here, you can set your own colors and even configure and run complete lighting programs. The controller is battery powered, so it can be used in places where AC power is not readily available.

This circuit was originally developed to drive color LED strips from a well-known Swedish chain of home furnishing stores. These color LED strips come with a simple controller, which allows you to manually select a limited number of colors. This restriction stimulated the author to develop a DIY controller with more capabilities. The resulting 'ambience lighting controller' is suitable for all RGB LEDs and LED strips that can operate from a 12 V supply voltage with a series resistor for current limiting.

In the author's intended application it was not possible to power the LED strips from the AC line, so the controller is designed to operate from a 12 V gel-cell battery.

The basic features of the circuit are described in the inset.

Schematic diagram

As you can see from the schematic in **Figure 1**, the circuit is fairly simple. The author chose a PIC16F887 for the microcontroller on account of its integrated EEPROM (for convenient storage of lighting programs), extensive I/O capacity and integrated ADC. Although the PIC16F877A is more popular, a sibling device was selected for this application because its ADC configuration allows the ADC inputs on RA0 and RA1 to be used without requiring any reference voltage input on RA2 or RA3. Here RC3 is connected to switch S1, which allows the battery charge state to be shown on the LCD module in two different ways. Connector K6 is the ICSP port for in-circuit programming of the microcontroller.

The microcontroller is clocked at 20 MHz by crystal X1. This relatively high clock frequency is necessary because the clock signal is divided by 4 inside the microcontroller. The resulting 5 MHz signal is essential for the PWM control function implemented in software.

The display module, a standard type with two lines of 16 characters (which is available in the Elektor Shop), is connected to port RB. If you use a different type of LCD, the polarity of the supply voltage for the backlight can be changed (if necessary) using jumpers in positions J1 and J2. Transistor T2 switches off the backlight after 10 seconds with no user input. The contrast can be adjusted with trimpot P1. Unlike most circuits

Features

- Supply voltage range 11–15 V
- Constant brightness over operating voltage range
- LC display (2 lines of 16 characters)
- Up to 13 user-definable colors with adjustable RGB values
- Three user-definable lighting programs with 20 color changes. The maximum duration for each color is 255 s, and the maximum duration of the transition to the next color is also 255 s. Both times can be set in increments of 1 s.
- Continuous operation with any one of the three defined programs
- Acoustic alarm when the battery is discharged, with automatic switch-off of the LED strips
- LED indicator for remaining battery charge
- Built-in charging circuit for the battery, with automatic switchover to trickle charge
- "Child lock" to prevent changes to color settings or programs
- Optional remote control via RS232/USB converter

Figure 1.
Schematic diagram of the RGB lighting controller, which is built around a PIC16F887 microcontroller.
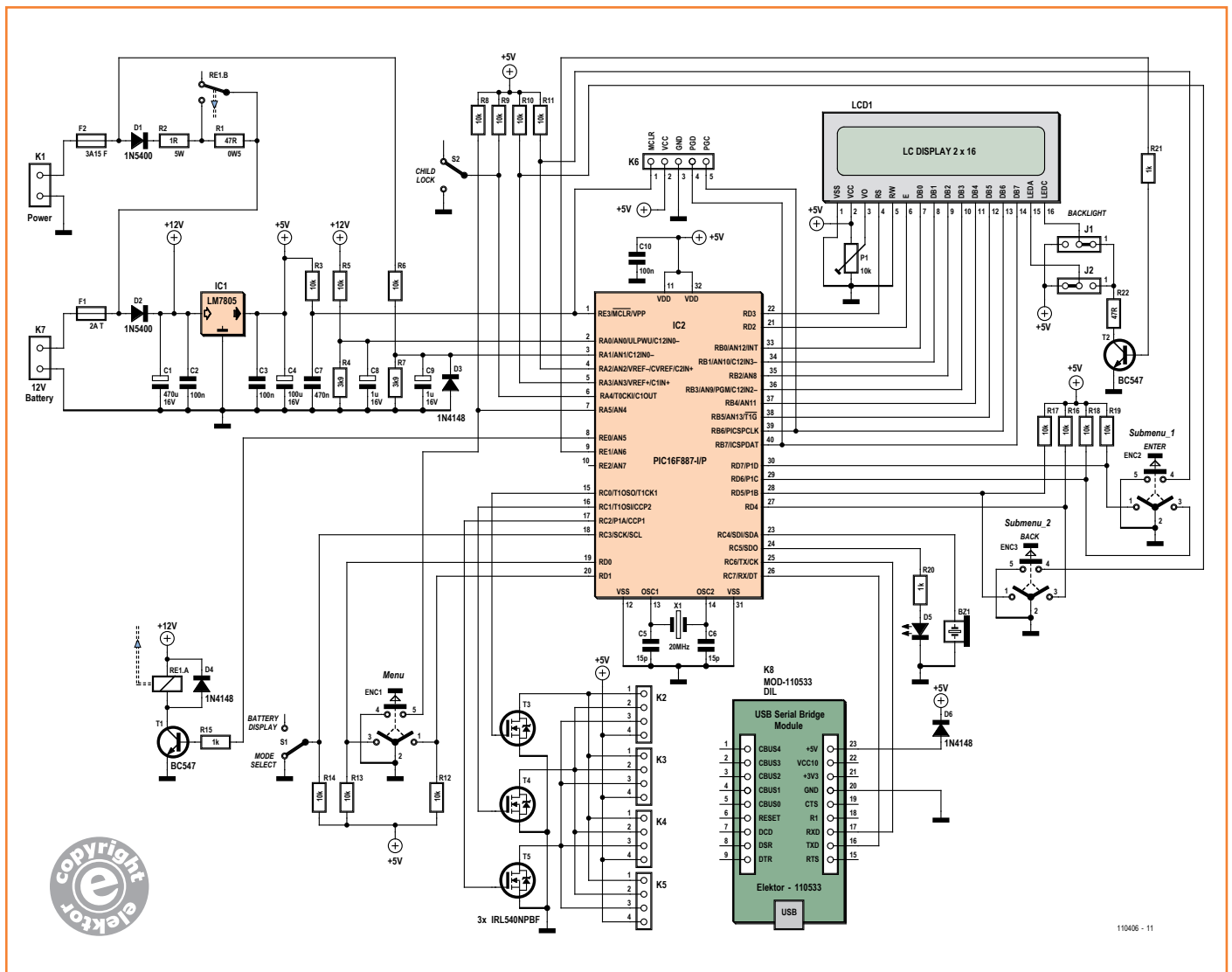
Figure 2.
The operator controls and LCD are also mounted on the PCB.
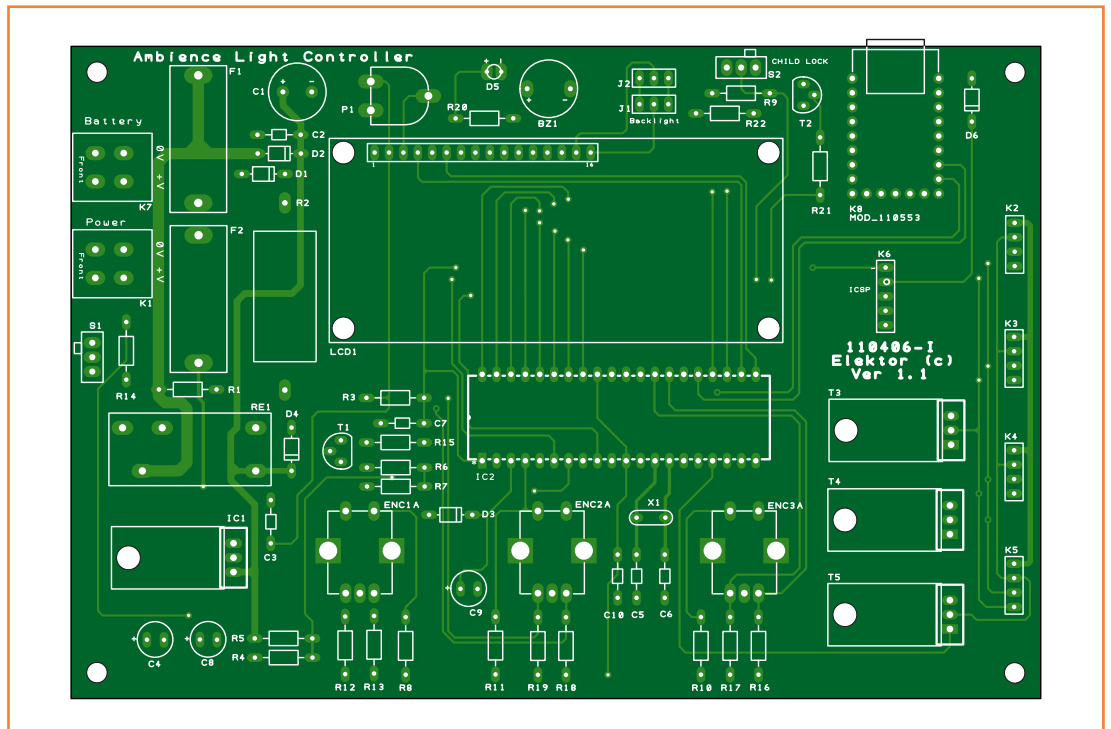
with an LCD module, here the LCD is driven in 8-bit mode instead of 4-bit mode. This simply represents a design choice, since the microcontroller has enough I/O pins available.

The color intensity of the connected LEDs is determined by pulse width modulation. Although the microcontroller has enough PWM outputs available for this purpose, the author decided not to

COMPONENT LIST

Resistors
R1 = 47Ω 0.5W
R2 = 1Ω 5W
R3,R5,R6,R8-R14,R16-R19 = 10kΩ
R4,R7 = 3.9kΩ
R15,R20,R21 = 1kΩ
R22 = 47Ω
P1 = 10kΩ preset, horizontal

Capacitors
C1 = 470µF 16V radial
C2,C3,C10 = 100nF
C4 = 100µF 16V radial
C5,C6 = 15pF
C7 = 470nF
C8,C9 = 1µF 16V radial

Semiconductors
D1,D2 = 1N5400
D3,D4,D6 = 1N4148
T1,T2 = BC547B
T3,T4,T5 = IRL540 (International Rectifier, Newark/Farnell # 8651078)
IC1 = LM7805
IC2 = PIC16F887, programmed, Elektor # 110406-41)

D5 = LED, red, 3mm

Miscellaneous
X1 = 20 MHz quartz crystal
F1 = fuse, 2AT (slow), with PCB mount holder
F2 = fuse, 3.15AT (slow) with PCB mount holder
BZ1 = active (DC) buzzer (with internal oscillator)
RE1 = relay, 12V, 1 c/o contact @ 2A min. (e.g. Finder 40.31.7.012.0000; Newark/Farnell # 1169158)
MOD1 = Elektor USB-FT232R breakout-board (BOB) [1]
S1,S2 = slide switch, angled pins, PCB mount (e.g. C&K OS102011MA1QN1; Newark/Farnell# 1201431)
S3,S4,S5 = rotary encoder with integrated pushbutton (e.g. Alps EC12E2424407; Newark/Farnell # 1520813)
K1,K7 = 2-pin PCB screw terminal block, 5mm lead pitch
K2,K3,K4,K5 = 4-pin pinheader, 0.1'' pitch
K6 = 5-pin pinheader, 0.1'' pitch
LCD1 = LCD 2x16 characters, with backlighting (e.g. Elektor # 120061-71))
J1,J2 = 3-pin pinheader with jumper, 0.1'' pitch
40-pin DIL-socket for IC2
PCB 110406-1 [2]

use them in order to simplify the routing of the prototype circuit board. For this reason, the PWM function is programmed in software. The LEDs are driven by the power MOSFETs T3, T4 and T5, which are designed to operated from TTL-level signals. The RGB LED strips (maximum 4) are connected to connectors K2 to K5. Each of the MOSFETs can easily supply several amps without extra cooling.

Bz1 is a buzzer with a built-in oscillator, so you only need to apply the supply voltage to get a penetrating acoustic tone. The control elements consist of three rotary encoders with built-in pushbuttons.

The RA0 and RA1 leads of the microcontroller are used as A/D converter inputs. The RA0 input measures the battery voltage, while the RA1 input detects whether a battery charger with a supply voltage above 14 V or so is connected. Voltage dividers R5/R4 and R6/R7 reduce the sensed voltages so they fall within the measuring range of the microcontroller. Capacitors C8 and C9 decouple any ripple voltage on the sense lines.

The schematic also includes a serial to USB converter module (Elektor BOB, order no. 110553-91 [1]), which can be used for linking to a PC if so desired. The circuit can be controlled remotely using a number of commands in a terminal emulator program. For more information about this, see the user guide (free download [2]).

The supply voltage is regulated by a conventional 7805 together with a few capacitors (IC1, C1–C4) and a diode (D2) for polarity protection. The value of fuse F1 in the power supply stage depends on the load. A 2-A slow blow fuse should be adequate with 6.5-ft (2-meter) LED strips, but to be on the safe side you should measure the load current in the actual application. Naturally, you should do this with all colors set to maximum intensity.

Connector K7 is the power input connector for a gel-cell battery, and an AC adapter with an output voltage of approximately 15 V at 2 A or more can be connected to K1. Transistor T1 drives relay RE1, which in turn shorts out resistor R1 when the battery has to be charged. The relay type is not critical—as long as the contacts can switch 2 A and the coil voltage is 12 V. See "Operation" for more information about the relay.

The circuit draws only 25 mA or so in operation, or approximately 50 mA with the backlight on.

PCB

Figure 2 shows the PCB layout designed at Elektor Labs for this lighting controller. Only leaded components are used, so board assembly is easy even if you don't have a lot of soldering experience. All components are fitted on the side with the component overlay. Using flat-jawed pliers, bend the leads of the voltage regulator and the MOSFETs at a right angle before fitting them to the board, so that they lie flat on the board after they are soldered. These components do not need heat sinks for normal use.

The microcontroller (optionally available preprogrammed) should be fitted in a socket. If you want to use a serial link to a PC, you should install the Elektor USB FT232R breakout board. It can be fitted directly on the PCB, or you can use a pair of 9-pin SIL socket strips.

Software

The program for this circuit was written in ANSI C using MPLAB and compiled using a full-function Hi-Tech C compiler running in evaluation mode (45 day trial license). The "lite" version of this compiler is not suitable in this case because it does not provide sufficient optimization, with the result that the executable code is too large for the 8 KB of program memory in the microcontroller. The source code and hex code, as well as the PCB layout, are available on the Elektor website [2] for free download. As usual, you can order a pre-programmed microcontroller in the Elektor Shop.

The key component of the software is the interrupt service routine (ISR). This routine was optimized using the stopwatch function of MPLAB to minimize its execution time.

The ISR is divided into several subroutines that can be executed every 100 µs, 5 ms, 100 ms or 1 s. The ISR is called every 100 µs, and it uses counters to ensure that the subroutines are executed at the previously mentioned intervals.

To reduce memory usage, a counter was initially used to determine the 1-ms and 100-ms intervals. A modulus calculation (which yields the remainder of a division operation) was performed each time the interrupt was called (every 100 µs), and if the remainder was zero, a 1 ms interval had expired. During debugging with the stopwatch function it turned out that this modulus calculation took so much time that it would be better to use a second counter to determine the 1 ms intervals.

The automatic light level control works as follows. The nominal PWM clock frequency is 100 Hz at 11 V. If the battery voltage is higher than 11 V, the LEDs will be brighter if the duty cycle remains the same. If the duty cycle is adjusted according to the battery voltage, a new duty cycle has to be calculated for each color. A much simpler method is to leave the 'on' time the same and reduce the PWM clock frequency as the voltage rises. This means that the calculation only has to be performed once to obtain the same result.

Listing 1

```
fade_step_red = current_red_value - next_red_value;
fade_step_green = current_green_value - next_green_value;
fade_step_blue = current_blue_value - next_blue_value;
fade_step_red = fade_step_red * 100;
fade_step_green =  fade_step_green * 100;
fade_step_blue = fade_step_blue * 100;
fade_step_red = fade_step_red / fade_time;
fade_step_green =  fade_step_green / fade_time;
fade_step_blue = fade_step_blue / fade_time;
```

Listing 2

```
tmp_red_value = fade_tmr * fade_step_red;
tmp_green_value = fade_tmr * fade_step_green;
tmp_blue_value = fade_tmr * fade_step_blue;
tmp_red_value = tmp_red_value / 100;
tmp_green_value = tmp_green_value / 100;
tmp_blue_value = tmp_blue_value / 100;
red_value = next_red_value + tmp_red_value;
green_value = next_green_value + tmp_green_value;
blue_value = next_blue_value + tmp_blue_value;
```

The difference between PWM clock frequencies of 100 Hz and 90 Hz is not visually noticeable.

Another bit of software that caused headaches for the programmer with the code for the color transitions. The calculation is very simple in principle: take the PWM value for each color, calculate the difference between this value and the next value, and spread the result over the transition time. Then raise or lower the PWM value for each color at each step during the transition interval (here the transition interval is given in increments of 100 ms).

The result of the division is typically a decimal fraction, which means that floating point variables have to be used for the calculation and for storing the values.

The PIC16F microcontrollers are simple 8-bit devices, and the compiler had a lot of trouble handling these "big" floating-point variables. This led to timing problems and errors in the compiled code. The solution to this problem was relatively simple. Multiplying and dividing integers takes less time and memory than working with floating point numbers. Accordingly, the difference between the PWM values for each color is first multiplied by 100 and then divided by the transition time, as shown in the following code segment (Listing1).

At every step during the transition interval (every 100 ms), the current PWM value is calculated and then divided by 100. The result is an integer "rounded off" to two decimal points, rather than a decimal number (Listing 2).

With this approach, an 8-bit microcontroller can handle color transitions without significant performance problems.

Operation

After the controller is switched on, it first shows a welcome message on the display consisting of its name and version number. The menu becomes available 1 second later. Operation of the controller is self-explanatory, but an extensive User Guide is also available as a free download [2]. You can scroll through the menu using rotary encoder ENC1. First you see the three options *Run Program <x>* (where *x* is 1, 2 or 3) for running one of the defined programs. Select one of the three programs and press *Enter* (the pushbutton of ENC2). To return to the menu, press *Back* (the pushbutton of ENC3).

The menu option *Charge Battery* selects monitored battery charging mode. First connect a 15-V battery charger, and then select this option. In this mode the relay is energized and shorts out resistor R1, so that more current can flow into the battery. The color LED strips are switched off to prevent potential damage from the higher than usual input voltage. When the battery voltage reaches 13.8 V, the relay is released. This reduces the battery charging current to the trickle charge

level, and the color LEDs are switched on again. The *Battery Charge* option shows the charge level of the battery in steps of 10%. The value is determined by measuring the battery voltage and expressing it as a percentage, where 0% corresponds to 0% and 13.8 V corresponds to 100%. LED D5 indicates the charge state of the battery. The LED is lit constantly when the battery is fully charged (13.2 V). When the battery starts to get low, it blinks for one second with a duty cycle that depends on the remaining battery charge. If the battery is nearly empty, the LED lights up very briefly. When the battery is so low that the color LEDs must be switched off, the buzzer start beeping.

Switch S2 provides a Child Lock function. When it is closed, the *Edit Program <x>* and *Edit <color>* menu options are not available.

To adjust a color setting, select *Edit <color>,* press *Enter*, and use the rotary encoders to set the red, green and blue levels over the range of 0 to 100%, in steps of 1%. Press *Enter* to save the new settings, or press *Back* to return to the menu without saving the new settings.

To configure a program, first select *Edit Program <x>* and then press *Enter*. Then use rotary encoder ENC1 to select the color, rotary encoder ENC2 to set the *Hold* time, and rotary encoder ENC3 to set the transition time.

(110406-I)

Internet Links

[1] www.elektor.com/110553

[2] www.elektor.com/110406

PWM Step Up Converter
Get up, stand up...
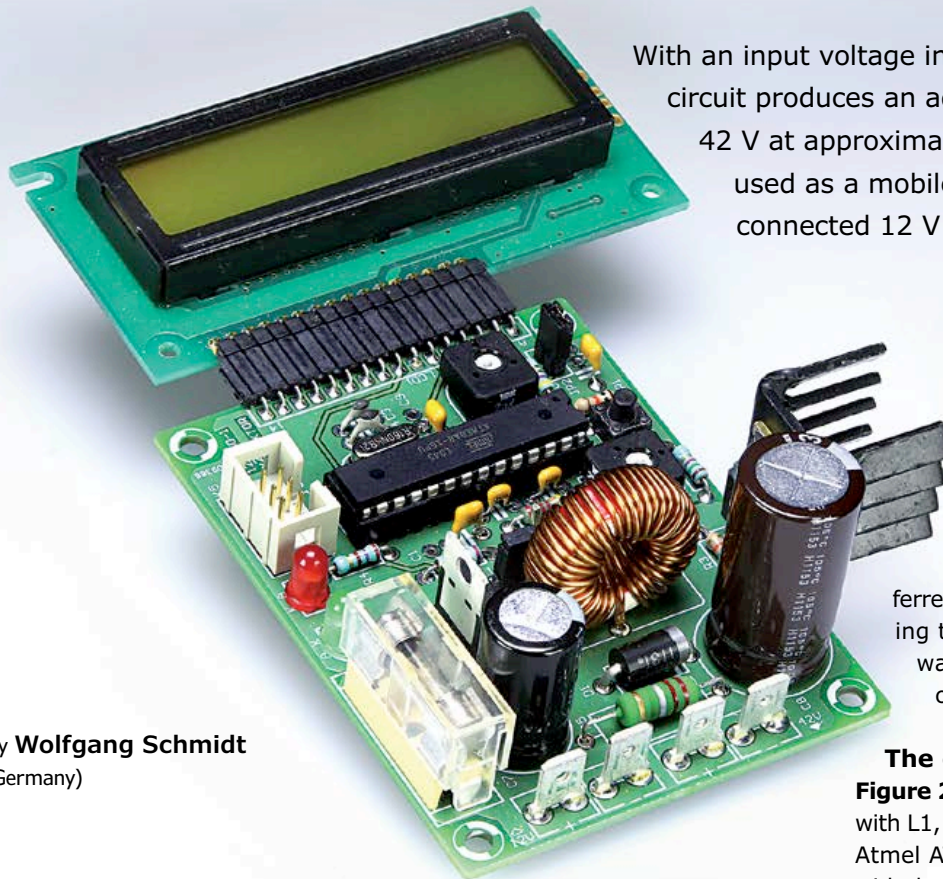
With an input voltage in the range from 8 to 16 V this circuit produces an adjustable output voltage up to 42 V at approximately 1 A. It can for example be used as a mobile charger for up to three series connected 12 V Lead-Acid batteries.

By **Wolfgang Schmidt**
(Germany)

A step-up or boost converter circuit converts a low voltage into a higher value output voltage. The circuit consists of an inductor, a capacitor, a diode and a switch (transistor) that's turned on and off by a pulsewidth modulated (PWM) signal. One switch cycle has a period T made up of an on-time t_1 and an off-time $T - t_1$.

During the PWM signal on-time the switch is closed (lower diagram **Figure 1**). The input voltage U_e is connected across the inductor L1 and providing the supply U_e has low enough impedance will produce a linearly rising current I_L through the coil, storing increasing energy in the magnetic field. When the switch opens, the coil's collapsing magnetic field induces a reverse voltage across the coil. This induced voltage is added to the supply voltage in the circuit, and provides a forward current flow through the diode where the energy is stored in the capacitor. It

can be said that the energy in the coil's magnetic field—which is stored largely in the ferrite core—is transferred through the diode to the capacitor during the switch-off stage. Those of you who want to explore the subject a little deeper can go to this introduction [1].

The circuit

Figure 2 shows how the step-up converter circuit with L1, D1, C8 and MOSFET T1 is configured. An Atmel ATmega8-16PU microcontroller together with the appropriate firmware produces the PWM signals to switch MOSFET T1. The PWM signal is produced from pin PB1 with a frequency of 66 kHz, using the internal fast-PWM mode. The output voltage is controlled by the mark/space ratio of the PWM switching waveform, and the microcontroller must be able to sense the output voltage level in order to control the waveform. This voltage feedback takes place over the voltage divider network formed by R6, R7 and P2. The preset is necessary because the data sheet indicates that the reference voltage level may be between 2.3 and 2.9 V and P2 allows some degree of calibration of the circuit. If the resistor values do not allow enough adjustment or if the value of R7 (43 kΩ) is difficult to source, adjustments in the firmware can be made to compensate. To set up the circuit you can use a known accurate DVM to measure the output voltage and tweak the preset until the displayed value corresponds

to the value on the DVM.

The microcontroller's built in A/D converter gives a resolution of 10 bits. The firmware calculates the voltage using a voltage divider network consisting of 47 kΩ (R7+P2) and 2.7 kΩ (R6). This set up gives a measuring resolution of 46 mV (((49.7 kΩ/2.7 kΩ)×2.56 V)/1023). The voltage reading shown on the 2×16 character LCD can be seen to change in steps of 0.04 V or 0.05 V. Step-up converters using this topology do not have any built-in current limiting. To reduce the possibility of overload a shunt resistor R5 is included in the ground output pin, and the voltage drop is measured by a second A/D input of the controller. The firmware now regulates the mark/space ratio of the converter switching waveform to reduce output current before the converter enters into discontinuous mode.

The networks formed by C10, C11/R8 suppress any RF noise on the analog A/D inputs. The LCD is used to display operational parameters (via menu selection) such as the output voltage and current. The circuit is provided with three push buttons: S1 resets the microcontroller while S2 and S3 provide increment/decrement control of the output voltage. With both buttons pressed at the same time the software will enter cur-

rent limit mode as shown on the display. In this mode S2 and S3 can now be used to increase and decrease the current limit setting. A short time after the last push button activity the display reverts to voltage display.

LED D3 indicates that an input voltage is present, if its goes out fuse F1 may have blown indicating that the circuit is possibly drawing excessive current or that the external power supply has developed a fault. D2 indicates that the current limiter is active.
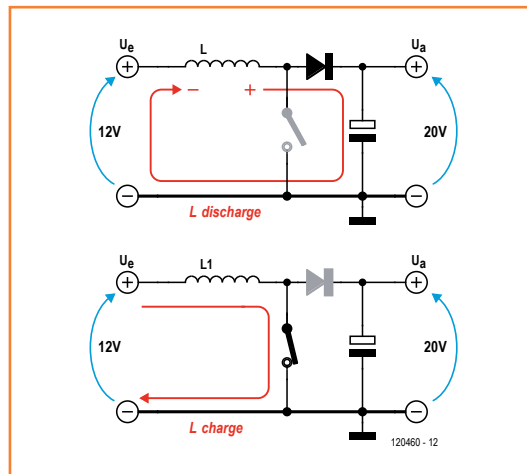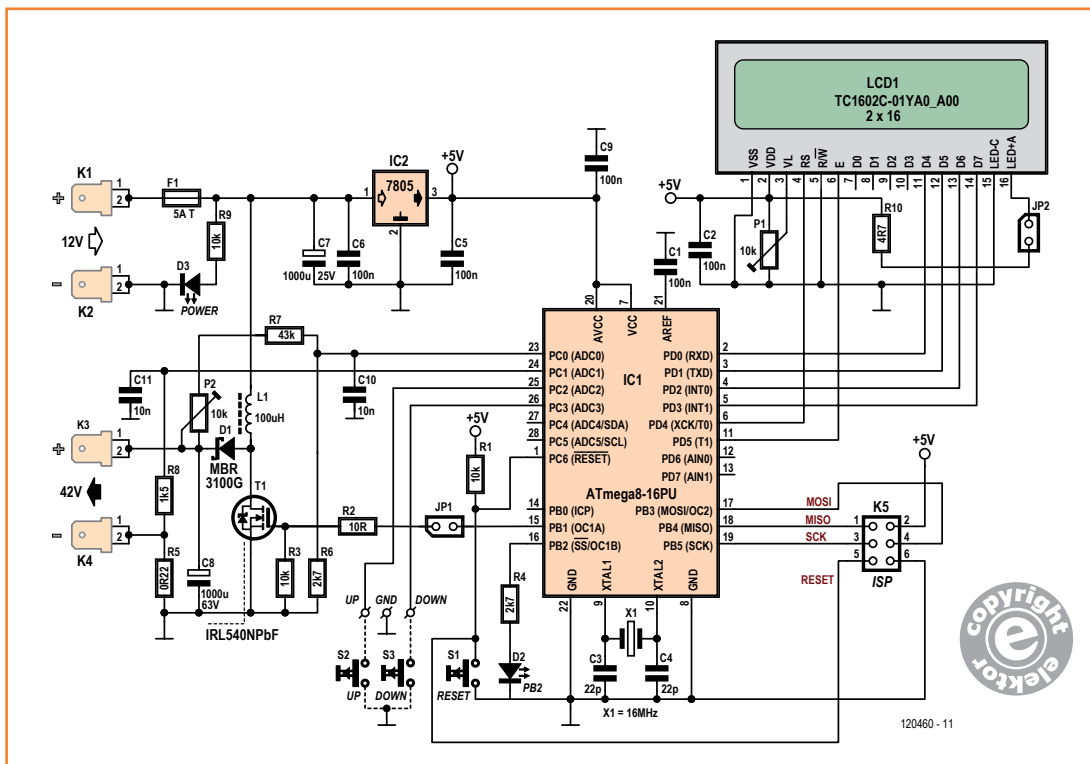

Figure 1.
The two phases of the step-up conversion process.

Figure 2.
The microcontroller-regulated step-up converter.

Putting it all together

The reset button S1 is fitted directly to the board because it should only be necessary to access it occasionally. Push buttons S2 and S3 are connected by flying leads to the pads on the PCB: Up, GND and Down. The buttons can be either two PCB mounted push buttons fitted to a small square of perf board or the larger panel mounted type fitted directly on the front face of the enclosure.

The two LEDs should, of course also be mounted where they can be seen. Preset P1 provides contrast adjustment of the LCD module. Jumper JP2 enables the LCD back light and can be replaced by a switch if required.

A small finned heat sink is used to keep the MOSFETs cool. A heat sink with a thermal resistance of 21 K/W is sufficient for output current up to 1 A. A standard radial-leaded or TO220 outline version of diode D1 can be used. Check the corresponding data sheet to ensure the correct polarity of the TO220 outline.

The electrolytics C7 and particularly C8 used in the switching circuit require some attention. At a switch frequency of 66 KHz it is important to use the low-loss type capacitors specified in the parts list. Standard electrolytic capacitors are unsuitable for this application.

For testing the unit in the lab we fitted the LCD display board with a pin header strip and plugged it directly into a box header strip fitted to the edge of the PCB. This proved ideal for testing purposes but when the unit is fitted into a project box or some other form of enclosure it may, for example be more practical to mount the display on the underside of the PCB. Fit a small square of insulating material between the display and the PCB to avoid any possible short circuits. (I normally fit a thin sheet of Pertinax or Paxoline between the boards).

A standard ISP connector is fitted at K5 to enable microcontroller programming. The controller needs to be powered up via the 7805 voltage regulator IC2 during the programming process. The supply voltage on pin 2 of K5 is used by the programming adapter (the AVRISP mkII for example) to determine the microcontroller's supply voltage (3.3 V or 5 V).

During the programming phase the controller's outputs are undefined so a jumper (JP1) is included in the connection to the MOSFET's gate.

This should be removed during programming to ensure that the MOSFET remains off. Otherwise the MOSFET can switch on via PB1 and short circuit the supply voltage causing the fuse to blow. R3 pulls the gate to ground when JP1 is removed ensuring it remains off. Be careful to remember to remove this jumper before programming! R2 reduces switching point instability produced by the high gate capacitance of T1.

The MOSFET gate capacitance introduces a delay whenever the MOSFET switches on or off. This gives rise to increased power dissipation in T1 because the source-drain is not switching immediately between off and on but passing through a resistive stage where power is dissipated in the device. A higher switching current would be able to remove the gate charge faster and result in a faster MOSFET switching times with less heat dissipation. A standard ATmega output can only supply around 30 mA drive current and is therefore a relatively weak current source.

The maximum output voltage level is limited by the voltage rating of D1 and T1. These two components would therefore be some of the first candidates to consider replacing to improve the circuit specification. The circuit as it stands is only really intended to show how a basic step-up converter can be built and act as a stimulus for further improvement.

Work in progress; the firmware

The source code as it stands is written in BAS-COM-AVR, and as usual it is available for free from the Elektor website [2]. As it stands it implements a very basic charge pump regulator and there is certainly plenty of room for improvement. These improvements include some critical points like the implementation of a true lead-acid battery charger with several charge phases.

The input current is approximately 3.5 times higher than the output current, hence the slow-blow fuse with a value of 5 A at the input.

The prototype was tested on two different lead acid batteries. A curious behavior was noted during testing; with a current setting of 0.2 *C* for example (which is relatively high for a Lead-Gel battery) the battery voltage quickly dropped after the maximum voltage setting had been reached and the charger switched off. In practice, using this software it is necessary to observe the charging process and know when to terminate the charge cycle.

COMPONENT LIST

Resistors
R1,R3,R9 = 10kΩ 5% 250mW
R2 = 10Ω 5% 250mW
R4,R6 = 2.7kΩ 5% 250mW
R5 = 0.22Ω 5% 1W
R7 = 43kΩ 1% 600mW
R8 = 1.5kΩ 5% 250mW
R10 = 4.7Ω 5% 250mW
P1,P2 = 10kΩ 20% 0.15W, preset,
   horizontal

Capacitors
C1,C2,C5,C6,C9 = 100nF 5% 63V,
   ceramic, 5mm or 7.5mm pitch
C3,C4 = 22pF 5% 50V, 5mm pitch
C7 = 1000μF 20% 25V, radial,
   Ø12.5mm, 5mm pitch (Panasonic
   EEUTP1E102, Avnet/Farnell #
   1890543)
C8 = 1000μF 20% 63V, radial,
   Ø16mm, 7.5mm pitch (Nichicon
   UPW1J102MHD, Avnet/Farnell #
   2112865)
C10,C11 = 10nF 10% 100V, 5mm
   pitch, ceramic

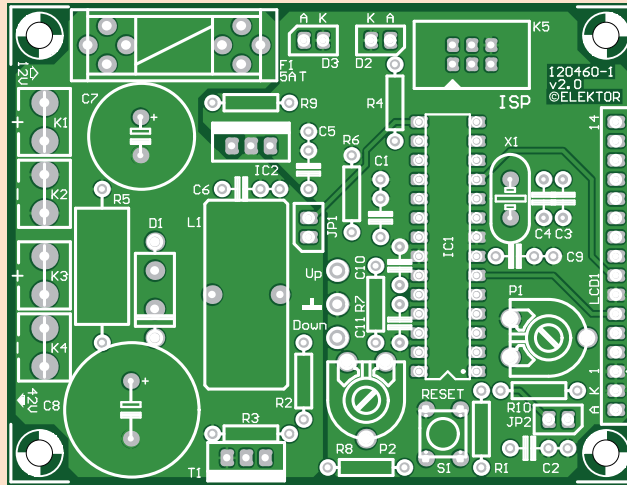### Semiconductors
D1 = MBR3100G
D2 = LED, red, 3mm

D3 = LED, green, 3mm
T1 = IRL540NPbF
IC1 = ATmega8-16PU, programmed, Elektor
 # 120460-41)
IC2 = 7805

Inductors
L1 = 100μH 5A 20%, radial
25mm, 8mm pitch (Würth Elektronik 7447070, Avnet/Farnell #
2082537)

Miscellaneous
F1 = fuse, 5A, slow, with PCB
20x5 mm holder and cap
JP1,JP2 = 2-pin pinheader, 0.1''
pitch, with jumper
K1-K4 = AMP plug, PCB mount,
0.2''
K5 = 6-pin (2x3) pinheader, 0.1''
pitch
S1 = tactile switch, 6x6 mm,
SPST-NO
S2,S3 = pushbutton, SPNO, PCB or
chassis mounting *
PC1,PC2,PC3 = solder pin, 1.3mm
diam. for S2, S3
Heatsink type FK230SAL1 (Fischer
Elektronik)
X1 = 16MHz quartz crystal, HC49/
US, 50ppm, C_{load} 18pF
LCD1 = LCD 2x16 characters (Elektor # 120061-71)
PCB # 120460-1

Figure 3. The step-up converter PCB.

The software as it stands does not provide a slick user interface offering a selection of sophisticated battery charging methods. The aim of this project is more to demonstrate how such a charger could be made and due to the open nature of the design and software, gives an opportunity for interested readers to hack as required and implement their own improvements.

A suggestion: The complete recharge cycle of a lead-acid (Gel) type battery should consist of two to four phases [3]. A (not fully) flat battery can be charged in the 'bulk-phase' using a constant current (0.1 to 0.2 C is reasonable), until a terminal voltage of 2.4 V per cell is achieved (This is as far as we go in the firmware in its present form. The battery will have approximately 80 % of a full charge at this point). Now the voltage is limited to the final terminal voltage, while the charge current is measured until it sinks to below one tenth of its maximum value. This second, so-called 'absorption-phase' almost completely charges the battery to around 98%, and the final 'float-phase' requires the terminal voltage to be reduced to 2.23 V per cell. The battery can now remain connected to the charger in this phase without the cell starting to gas.

You may be wondering what happened to the fourth phase; well this is only necessary when the battery is deeply discharged (below 1.75 V per cell). In this state the battery is nursed back to health using a small trickle charge until the cell reaches its lower voltage threshold.

If you have been working on software improvements to this design or have started, but reached an impasse, why not visit our project page [4] and share your experiences!

(120460)

Internet Links

[1] Switch mode basics:
    http://schmidt-walter.eit.h-da.de/smps_e/
    smps_e.html

[2] www.elektor.com/120460

[3] Charging lead-acid batteries:
    www.batterystuff.com/kb/articles/battery-articles/battery-basics.html#9

[4] www.elektor-projects.com

Charge-a-Phone on NiMH

with the Elektor USB Power Pack

By **Ton Giesberts**
(Elektor Labs)

The goal of this project is to allow standard AA size rechargeable batteries, like NiMH, to effectively charge portable devices like smartphones and tablets through the USB connector.
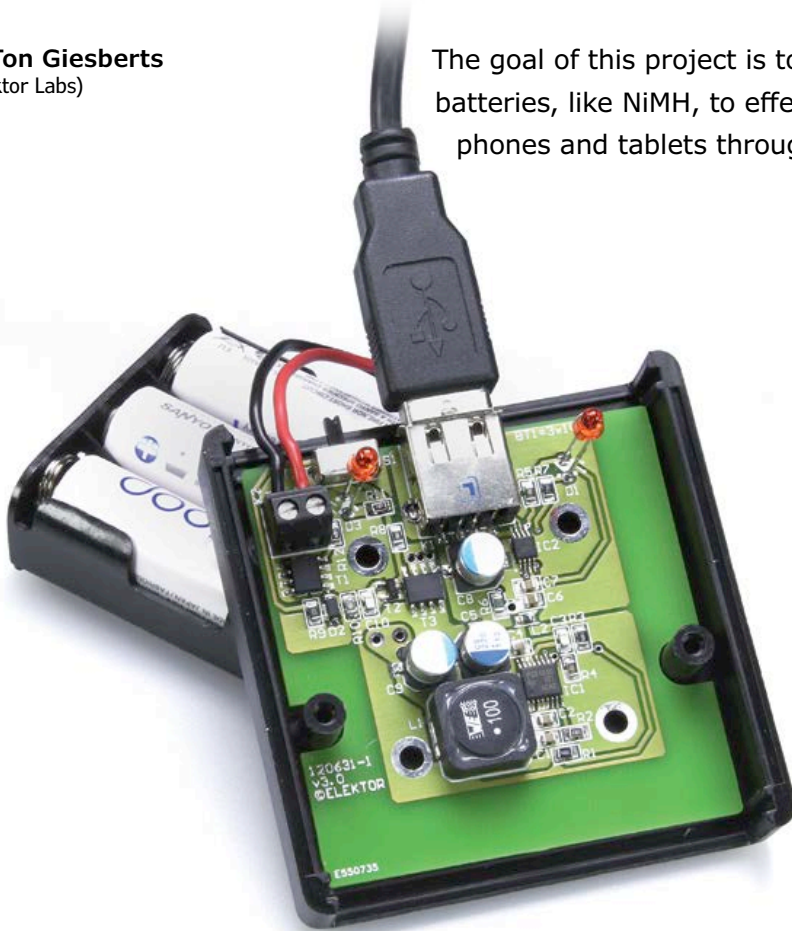
In numbers, there are probably more chargers around for NiMH than for Lithium-ion or Lithium-polymer batteries. If you wanted to use the latter you'd have to integrate the charger circuit, making the 'battery pack' more expensive and complex. Keeping the batteries separate from the enclosure still leaves the *option* to use Li-ion or LiPo. Luckily their initial voltage (3.6–3.7 volts) is almost the same as three NiMH batteries in series. Also, by using a separate battery holder, you're able to exchange discharged batteries with fresh ones without having to charge first, or open the enclosure. That's a big plus if your phone, tablet or e-gizmo is in serious need of charging and you're in the middle of nowhere.

How many batteries to use?

The circuit has to produce 5 V and be able to deliver up to 1 A of output current. Four freshly charged NiMH batteries can have a voltage well above 5 V, so it seems prudent to keep the number at three. However, in USB speak '5 V' is nominal, the actual range being 4.35 V to 5.40 V. Although that's in favor of four batteries again, we still wish to produce a 5.00 V supply that's accurate, if only because some designers use the USB voltage as a reference (keep that limited to non-critical applications). Three.

Boost converter TPS61030

The lower voltage of three batteries implies the use of a smaller battery holder but also the need for a boost converter. There's an excellent device available from Texas Instruments, the TPS61030. It's a synchronous boost converter with an internal 4-amp switch and an efficiency of 96 % (dependent on input voltage and output current of course). The converter also has an (optional) Low Battery Comparator to prevent deep discharging of the batteries. An extra undervoltage lockout (1.6 V) prevents the converter from malfunctioning. The internal reference voltage is 0.5 V, making it easy to calculate the voltage divider for the correct output voltage. Here 1.8 MΩ is used for R3, and 200 kΩ for R4. According to the datasheet, only if R4 is significantly lower than 200 kΩ then an extra capacitor for stability is necessary in parallel with R3. Here a 10 pF cap is used just to be sure.

Resistor R2 should be low enough to eliminate the input current of the comparator (about 10 nA). A value of 500 kΩ is recommended. The comparator level is around 500 mV with a hysteresis of

10 mV. A threshold of 1.1 V was chosen to define one fully drained battery. Values of 1.8 MΩ for R1 and 330 kΩ for R2 result in a theoretical threshold of 3.23 V. If the total battery voltage drops below this threshold the output of the comparator goes Low (LBO). This output is used to disable the output circuit.

The decoupling of the input voltage by C1, C2 is in accordance with the recommendations in the datasheet. The decoupling of the output voltage depends on the maximum output ripple. A few millivolts is ideal, but the ESR of the capacitors in particular, as well as the board layout will result in a higher value in practice. Theoretically the ripple should be around 1 mV with an output buffer capacitor of 220 µF. In practice about 60 mV was measured across C5 (3.50 V in, and 1 A load). C5 has a rated ESR of 20 mΩ at 100 kHz. The switching frequency of 600 kHz is a lot higher, and the higher switching current accounts for the higher ripple voltage. To suppress switching noises a ferrite bead (L2) is placed in series with the output circuit. This way the ripple voltage is also reduced. The final output capacitor (C8) reduces the ripple voltage even further. For the calculation of the inductor a change of 10 % of the maximum average inductor current was taken into account. At 3.20 V the average
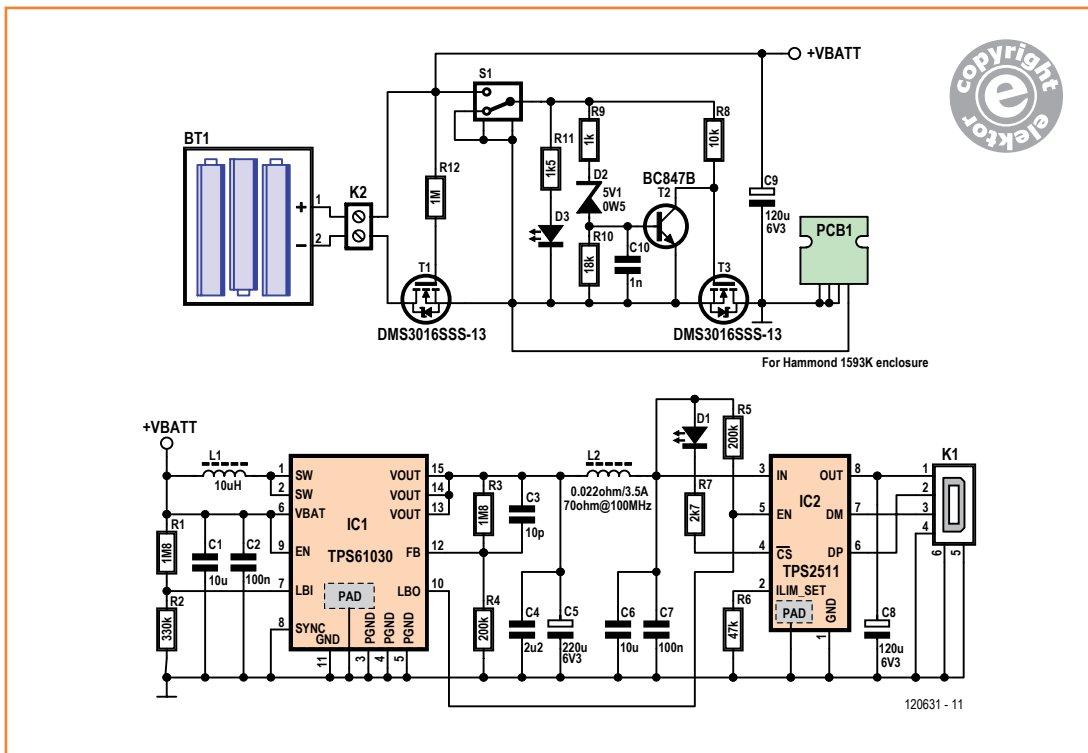
inductor current is close to 2 A. Given de formula in the datasheet (*SLUS534E*), this gives an inductor value of about 10 µH.

The Sync pin can be used to operate the converter in different modes. We selected Power Save by connecting Sync to ground, which improves efficiency at light loads (the device then operates discontinuously). The converter only oper-

Measurements and Specifications

Input voltage range . . . . .3.3 – 4.1 V
Maximum input current . . . . .1.7 A (V_{in} = 3.33 V)
Output voltage . . . . .4.93 V (no load)
. . . . .4.92 V (0.3 A load)
. . . . .4.82 V (1 A load)
Low battery threshold . . . . .3.25 V
Overvoltage protection . . . . .4.30 V
Efficiency . . . . .95 % (3.52 V_{in}; 0.3 A out)
. . . . .90 % (3.52 V_{in}; 1 A out)
Supply current (no load) . . . . .4.5 mA (V_{in} = 3.6 V)
Power LED lights at . . . . .1.6 V

Losses measured at 1 A output current:
Across T1 and T3 (each) . . . . .23 mV (1.7 A input current)
From L2 to USB connector on PCB. . . . . . .85.3 mV
Across IC2 . . . . .80 mV
USB connection (each) . . . . .13 mV

120631 - 11

ates when the output voltage drops below a set threshold. On the down side, the output ripple voltage increases slightly. With no load, an 80-mV sawtooth was noticed with a 150-ms period. But it got better rapidly with increased loading.

TPS2511: glue for USB

A special IC type TPS2511 is used for controlling the output. Texas Instruments calls it a *USB Dedicated Charging Port Controller and Current Limiting Power Switch* but we still liked it. Here's why. It's often not enough to just put 5 V on a USB connector and get a device to work.
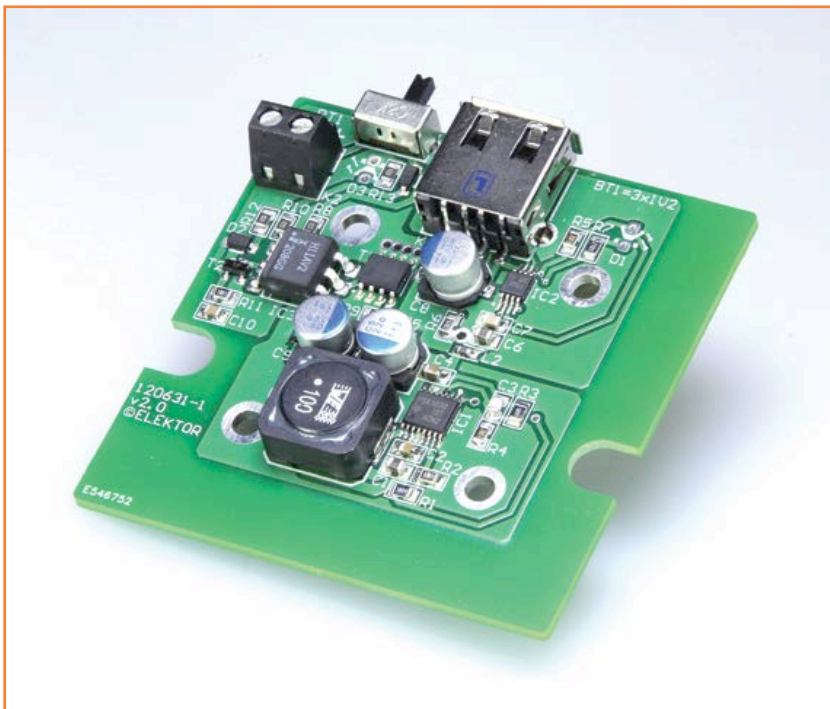
It's great that phone & gizmo manufacturers increasingly fit their devices with USB connectors as the charge port, but chargers are unlikely to be compatible between fruit and non-fruit platforms, and different manufacturers. For example, some devices expect specific voltages on the data lines, or simply a connection (resistor) between the data lines to recognize a charger (Dedicated Charger Port or DCP). The TPS2511 supports three of the most common protocols:
• USB Battery Charging Specification, Revision 1.2 (BC1.2);
• Chinese Telecommunications Industry Standard YD/T 1591-2009;
• Divider Mode.

For an exhaustive description of all possibilities of the TPS2511, please refer to the Texan datasheet (*SLUSB18*).

The TPS61030 can deliver 2 A at a battery voltage of 3.3 V, and the TPS2511 can handle this current also. But at 1 A output current and 3.33 V input voltage the converter already draws 1.7 A from the batteries. At 2 amps output current this will be more than doubled, because of the higher losses. Also, the battery capacity drops at higher output currents. That's why the TPS2511 is connected to work as a 5-watt charger. Its DP pin is connected to the D– line, and the DM pin to the D+ line of the USB connector. The current limit is set marginally higher than needed (R6 = 47 kΩ), preventing the TPS2511 from premature output voltage limiting.

The Current Sensing Report pin is not used in the expected way. Instead of compensating voltage loss by changing the feedback of the converter (not really necessary at 1 A maximum output current) the pin is used to drive an LED (D1). When D1 lights up you know that more than half of the maximum output current is being drawn. The LED current is a little over 1 mA. As already mentioned, the Low Battery Comparator output drives the EN (Enable) pin of the TPS2511. This way the output voltage is cut off in case the batteries are flat. R5 is needed because the comparator output is in high impedance state when not active.

Polarity and overvoltage guard circuits

The battery pack connection to the PCB is by way of a screw header (0.15'' lead spacing). So in practice it's possible for the batteries to be connected the wrong way around. To prevent damage to the circuit and still have virtually no losses when properly connected, a small n-channel power MOSFET (T1) is used, purposely the wrong way around. When connecting the batteries with the proper polarity, the body diode is in the conducting direction, and the MOSFET is fully turned on, its gate positive with respect to the source through R12. There's no problem with the current flowing from source to drain. In case the batteries are connected the wrong way around the gate is negative and the MOSFET is turned off and the body diode effectively blocks the battery voltage. The maximum permissible gate voltage of the MOSFET used is 12 V, which also constitutes the maximum voltage the circuit will sur-

vive. At 1.7 A input current the MOSFET(s) drop a minuscule 23 mV (measured on prototype). To avoid having to use an expensive heavy duty on-off switch, the overvoltage protection is combined with a smaller—hence much cheaper—switch. The overvoltage protection is kept simple. When the supplied voltage is too high a zener diode (D2) is used to switch on an n-p-n transistor (T2) which in turn cuts off the gate voltage of MOSFET T3, which is connected as you would expect. The 5.1-V zener diode already conducts below the specified zener voltage. At a battery voltage of 3.60 V the current through D2 is about 12 µA. At 4.25 V, it's over 30 µA. This can easily be measured across R9, which prevents the current through the zener diode from snowballing when the input voltage exceeds 5.70 V or so. In case the overvoltage protection acts too early (due to possible tolerance of the zener diode), feel free to adapt R10, remembering that a lower value gives a higher threshold. The overvoltage protection is needed in case an AC power adaptor—hopefully set to less than 12 V out—or a 9-V battery is connected. The TPS61030 can withstand 7.00 V (absolute maximum, 5.50 V recommended). The problem with the boost converter is that the output voltage rises when the input voltage exceeds the regulated output voltage (here, 5.00 V nominal).

Construction

The PCB is specifically designed for a Hammond Manufacturing enclosure (see parts list). It's cheap and easy to adapt to our application. The PCB is fixed with four self-tapping screws, and the top and bottom halves with two longer ones. The front and back are separate panels. In one panel only, three holes have to be drilled. The holes for the USB connector and switch should be aligned with the parts on the PCB. The same is true for two holes for the LEDs in the top cover. The exact placement of the hole for the two wires to the external battery holder is not that critical—there's a large margin to play with. It can be located anywhere else for that matter. A power jack may also be used—there's more than enough room in the other panel. Avoid any extra contact resistance where possible, as it will reduce the efficiency of the device as a whole.

The holes for fixing the PCB are also used to connect the board's top and bottom power planes. Be aware that the hole next to screw header K2

is not connected to ground but the net between T1 and T3. Connecting this net to ground will not cause any damage but simply turn on the circuit. The other three holes are connected to ground, however the hole next to IC2 is specifically output ground. It's assumed the PCB is placed in the above mentioned hard plastic (ABS) enclosure. Finally, do not touch junction R3/C3/R4 with the circuit in operation. This is a high impedance point and any hum introduced here may destroy IC1.

(120631)

COMPONENT LIST

Resistors
(0805, 125mW)
R1,R3 = 1.8MΩ 1%
R2 = 330kΩ 1%
R4,R5 = 200kΩ 1%
R6 = 47kΩ, 1%
R7 = 2.7kΩ, 5%
R8 = 10kΩ, 5%
R9 = 1kΩ, 5%
R10 = 18kΩ, 5%
R11 = 1.5kΩ, 5%
R12 = 1MΩ, 5%

Capacitors
C1,C6 = 10µF 10V 20%, X5R, 0805 (Taiyo Yuden LMK212 BJ106MG-T)
C2,C7 = 100nF 50V 10%, X7R, 0805
C3 = 10pF, 50V, ±0.5pF, C0G/NP0, 0805
C4 = 2.2µF 6.3V, 10%, X5R, 0805
C5 = 220µF 6.3V, 20%, SMD, Ir=2.8A (Nichicon PCS0J221MCL1GS)
C8,C9 = 120µF, 6.3V, 20 %, SMD, Ir=2.8A (Nichicon PCS0J121MCL9GS)
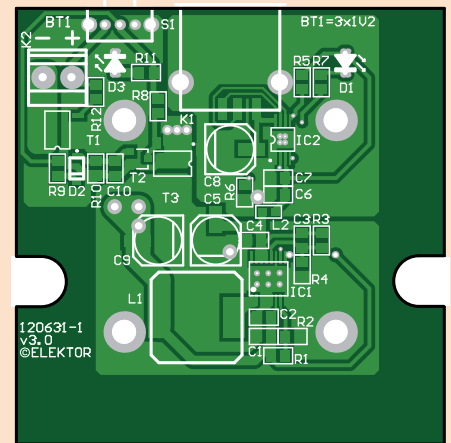C10 = 1nF, 50V, 10%, X7R, 0805

Inductors
L1 = 10µH, 5A, 0.025Ω, 20% (Würth Electronics 74477110)
L2 = ferrite bead, 70Ω @ 100MHz, 3.5A, 0.022 Ω, 0603 (Murata BLM18KG700TN1D)

Semiconductors
D1,D3 = LED, red, 3mm through hole (low current)
D2 = 5.1V zener diode, 0.5W (SOD123), Diodes Inc. MMSZ5231B-7-F
IC1 = TPS61030PWPG4 (Texas Instruments)
IC2 = TPS2511DGN (Texas Instruments)
T1,T3 = DMS3016SSS-13 (SO8)
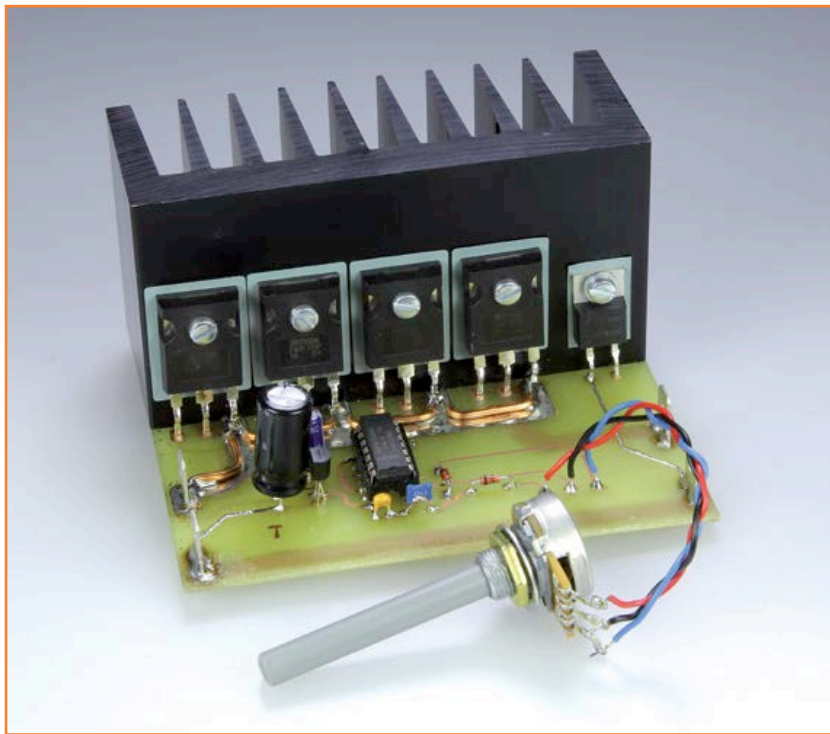T2 = BC847B

Miscellaneous
K1 = USB connector, type A, receptacle, PCB mount, SMD
K2 = 2-way vertical screw header, 0.15'' (3.81mm) pitch (Phoenix Contact MKDS 1/2-3.81)
S1 = Slide switch, SPDT, right angle, 100mA (C&K Components OS102011MA1QN1)
Enclosure, 66.22 x 67.22 x 28.00 mm (Hammond Manufacturing 1593KBK)
PC board screws (#4 x 1/4" self-tapping, 6.4mm, Hammond Manufacturing 1593ATS50)
BT1 = 3 AA battery holder, snap contact (Keystone 2475) + battery clip (BUD Industries HH3449)
3 NiMH batteries. PCB # 120631-1 v3.0

Big Amps DC Motor Driver

By **Ton Giesberts (Elektor Labs)**

This simple circuit is designed for use with all kinds of DC motors up to 40 amps. Basically it's just a simple oscillator driving a bunch of power MOSFETs. The oscillator is a rudimentary RC type around a single Schmitt trigger device (IC1a) from a 40106 hex inverter package. When the wiper is turned towards D2, potentiometer P1 gives maximum voltage to the output. The two diodes prevent short-circuiting the output to the input. At the extremes of P1, the charge and discharge times are minimal. In the prototype of the circuit the negative-going pulse was found to be 1 μs wide, and 1.6 μs for the positive pulse.

The next two inverters, IC1b and IC1c, clean up the oscillator signal, driving a buffer stage comprised of three inverters in parallel, IC1d, IC1e and IC1f. Resistor R1 was added to hold off the MOSFETs in case the 40106 is absent. The total input drive capacitance of the four MOSFETs amounts to almost 8 nF—clearly too much for the buffer to fully charge and discharge when P1 is turned to its extreme positions. That's convenient however because in practice it allows the motor driver to manage the full output voltage span (i.e. 0–100 %).

The operating frequency is in the 1 kHz ballpark. On a prototype 1.07 kHz was measured. Diode D3 at the output suppresses the reverse energy (back emf) generated by inductive loads, which includes all DC motors.

High output currents and back emf are issues here. On an early prototype of the board, the tracks to D3 were too narrow, and when testing the circuit with one of the motors in an Elektor Wheelie one of the tracks burned out. Fully loaded, each of the two motors used in the Wheelie draws up to 20 A at 24 V. The circuit was tested at 40 A and 24 V with a resistive load. However, the PCB as designed and supplied is not able to handle such high currents. The solution is to beef up the copper tracks carrying high current with pieces of 13 or 14 AWG (approx. 2.5 mm²) massive copper wire. Possibly two paralleled pieces of 16 AWG (approx. 1.5 mm²) are easier to get into place. For this reason the PCB does not have solder stop masks. The thicker lines in the schematic provide a global indication of where high currents can be expected to flow. The supply for the 40106 is no more than a 78L12 voltage regulator (IC2) with the usual entourage of decoupling capacitors large & small.

The speed control potentiometer may be mounted off the board and connected with light duty wires. The heatsink is best secured to the PCB with 3-mm (6 BA) screws. Make sure the heatsink doesn't come in contact with the solder pads for the MOSFETs. Then determine the correct positions of the transistor mounting screws, and D3. To prevent mechanical stress within the semiconductor legs, give them a light bend—there are special tools available for this—and only then locate the positions for the holes. Tap 3-mm (approx. 1/8'', 6 BA) threading. Don't forget to isolate all semiconductors on the heatsink. Because of the low switching frequency there's a good chance you can hear a whine from the DC motor—it's pretty normal and no cause for alarm.

(120406)

COMPONENT LIST

Resistors
R1 = 10kΩ, 5%, 0.25W
P1 = 100kΩ, 20%, linear potentiometer, 0.2W

Capacitors
C1 = 470µF 35V, 20%, 3.5mm lead spacing
C2 = 10µF 25V, 20%, 2mm lead spacing
C3 = 100nF, 50V, 20%, ceramic, 5mm lead spacing
C4 = 22nF, 100V, 20%, ceramic, 5mm lead spacing

Semiconductors
D1,D2 = 1N4148
D3 = RURP8100

T1-T4 = IRFP150N
IC1 = 40106
IC2 = 78L12

Miscellaneous
4 pcs. FastOn spade terminal (tab), straight, PCB
  mount, 0.2'' (5.1mm) lead spacing
Heatsink, 1.9K/W, 100 x 40 x 50 mm, Fischer Elek-
  tronik type SK 92/50 SA
TO-3P silicone elastomer insulation (T1-T4)
TO-220 device insulating kit; mica sheet + bush (D3)
PCB # 120406-1 v1.0

X-Treme Inrush Current Limiter
A controlled start for Big Electrolytics & Co.

By
Raymond Vermeulen
(Elektor Labs)

Figure 1.
A Bob Pease style sketch
of an idea for an inrush
current limiter.

This all-analog, microcontroller free (!) project got designed in response to cries for help from a diehard model plane enthusiast on the Elektor staff. He likes to fly large high powered models. One problem he ran into was self-destructing power connectors when connecting the battery pack to the plane (i.e. the motor controller). Every time the damage was due to heavy sparking, due in turn to high inrush currents.

Those were expensive sparks as it turned out, because the connectors are 6-mm diameter, gold plated types. Clearly an inrush current limiter is called for to ensure a controlled, spark-free initial current flow rather than a thump and a small explosion. Such a regulator did not drop from the skies however, and took some time to develop at Elektor Labs. Below is a digest of how the project evolved from doodling to a working model keeping everyone happy.

Good guidance was found in Motorola Application Note number *AN1542* [1]. Using rough concept sketches (**Figure 1**) an inrush current limiter got designed for 37 volts battery power and a 200 amp load in normal operation. To achieve a low overall $R_{ds(on)}$ it is best to use a couple of MOSFETs in parallel. After an LTspice simulation run, the pain appeared to be not in the amps but in the load capacitance responsible for the inrush current, so the circuit got designed for the worst case scenario. Still, there were concerns about the safe operating area of the MOSFETs. To test the water, measurements were carried out on a small 10-A 3-phase BLDC motor driver, and that turned out to have "just" 120 µF input capacitance. A bit later a bigger motor controller turned up specified for 120 A, and this was found to represent an input capacitance of 13,800 µF (13.8 mF) at an ESR of about 2.7 mΩ.

Moving towards a practical circuit the type IPB017N06N3 MOSFET from Infineon was chosen mainly based on the promise of 1.7 mΩ of 'on' resistance per device, not forgetting relatively low cost and ready availability from the distributors.
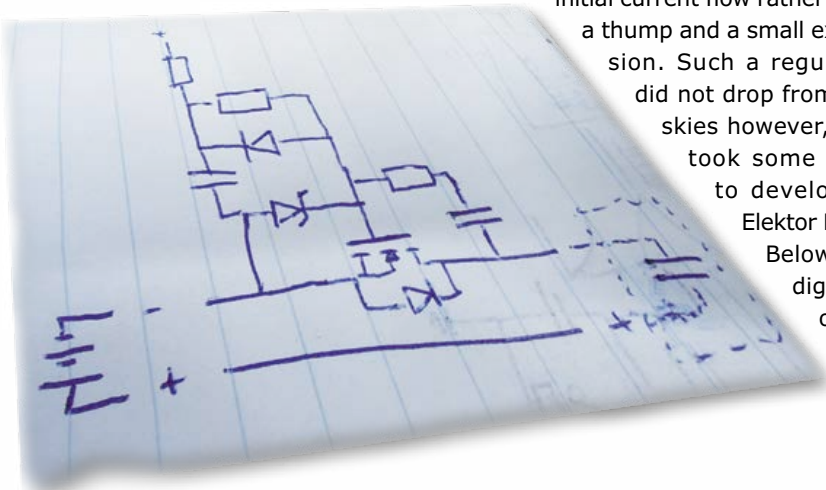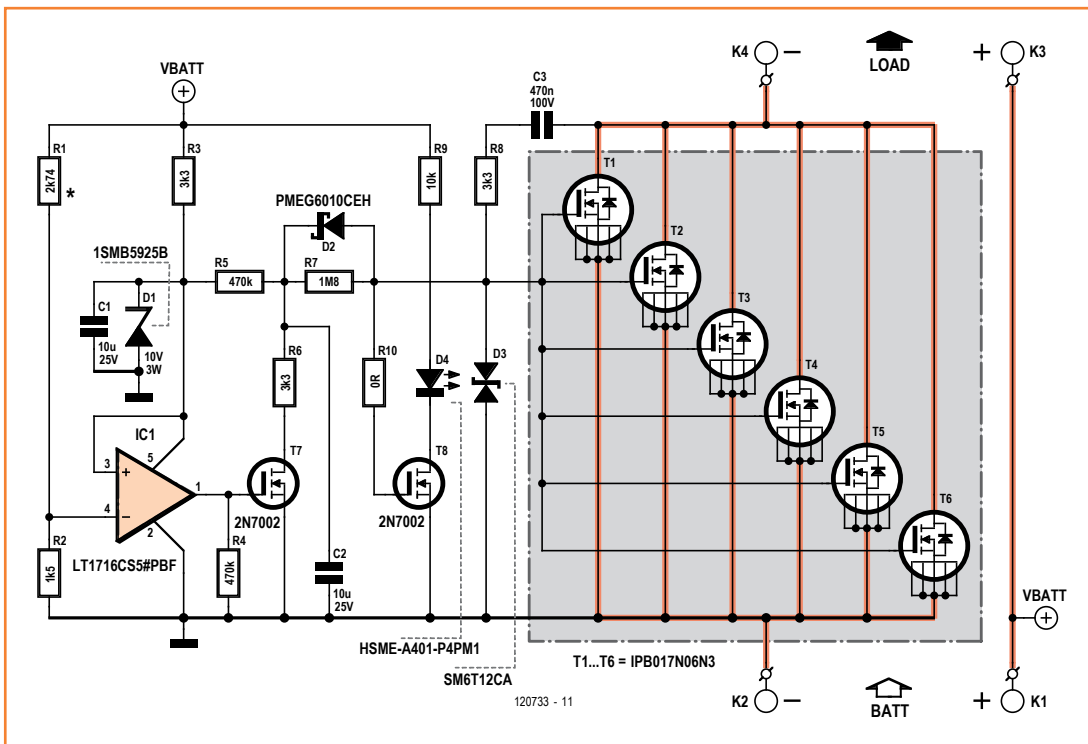
Figure 2.
Schematic of the X-Treme Inrush Current Limiter. High current PCB tracks are highlighted and thicker.

Now the question remains: how many MOSFETs do we need?

Back to the LTSpice simulation, now using the IPB017N06N3 model, some component values were in need of tweaking. Also, a heatsink was found—cheap, standard size (1/2-brick) and with predrilled holes.

Looking at the schematic in **Figure 2** there are some marked differences with the version proposed in *AN1542*. Motorola shape the current into a square wave, causing a sudden current and power surge which slowly dies out. By contrast, the circuit shown here has the current increase slowly, resulting in a sawtooth shaped current. Consequently, the power dissipation graph (P_{FET}) looks like an inverted parabola. **Figure 3** shows the basic waveforms—arguably they respect the safe operating area of the MOSFETs far better than *AN1542*.

A TVS (transient voltage suppression) diode, D3, helps to protect the MOSFETs in case of accidental polarity reversal.
An early prototype was tested with a 15,000 µF (15 mF) capacitor with and without a resistive load, connecting to a 40-V supply through the

Table 1. Trip value / battery voltage dependency			
Battery Type (Lithium)	**Battery Volts**	**V_trip**	**R1**
4S	12 – 16.8 V	11 V	180 Ω
5S	15 – 21 V	14.2 V	620 Ω
6S	18 – 25.2 V	16.8 V	1 kΩ
8S	24 – 33.6 V	22.9 V	1.87 kΩ
10S	30 – 42 V	28.9 V	2.74 kΩ
11S	33 – 46.2 V	31.8 V	3.16 kΩ
12S	36 – 50.4 V	34.9 V	3.6 kΩ

X-Treme circuit. Everything seemed to function as expected, although with no resistive load connected, the undervolts lockout did not function correctly on the falling edge.

As a final test, the circuit was used with a BLDC controller driving a 10 kW motor, unloaded, drawing 8.5 amps at continuous speed and 20 to 30 amps when throttling. Tests were done at 37 V and 48 V, doing 'cold starts' several times over. Although cables and connectors got noticeably warm, the MOSFETs and the rest of the circuit remained cool. No "thump" sounds were heard (so customary from high-current loads), or exploding capacitors.
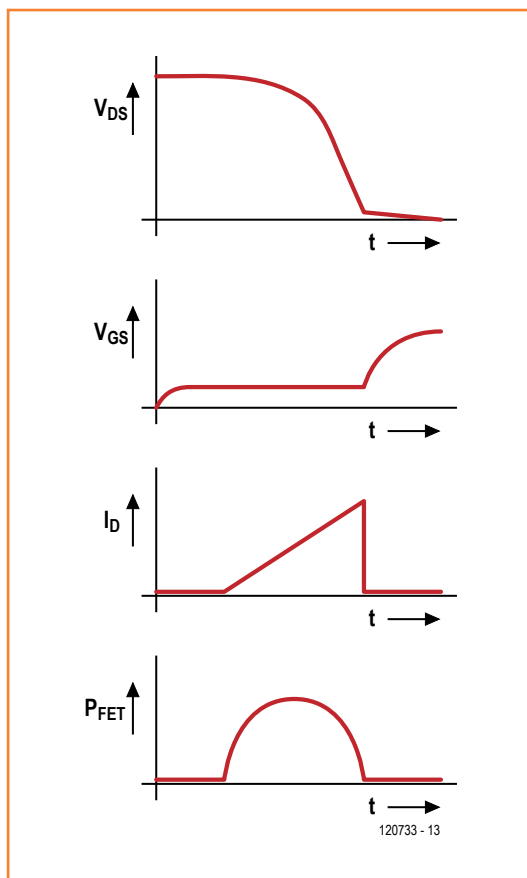
Figure 3.
Our circuit results in a
reverse-parabolic shape for
the power dissipation of the
MOSFETs.

tice, the circuit should not be used with battery voltages lower than 12 volts. Fortunately that's a rare occurrence in high-power (BLDC) motor applications—you can easily see why.

The circuit board potentially carries extremely high currents, both 'surge' and 'continuous', meaning **you have to strengthen all MOS-FET source and drain PCB tracks, and the whole length of the BATT– and BATT+ PCB tracks, with pieces of 2.5 mm² (13 AWG) solid copper wire, preferably two in parallel**. Most of this plumbing work is in the area covered by the heatsink later. If you find 1.5 mm² copper wire (16 AWG) easier to juggle with, that's fine also but do three or even four pieces in parallel. Also apply generous amounts of solder along the tracks and the copper wires—it's a bit like Plumbing-4-Beginners. If for some reason your board comes with a solder mask on the above mentioned tracks, remove the masking material and expose the copper by scratching with a sharp hobby knife. The pre-tin and install the helper wires.

The battery and load connections K1-K2 and K3-K4, must be made using high quality terminals of your choice, preferably gold plated. Get the best you can find, round or flat ('FastOn' / spade type), whichever you prefer, as long as you solder them straight to the PCB tracks. Remember, every milliohm counts in this circuit and you do not want to lose motor power or torque during takeoff, now do you. To prevent polarity reversal, consider using a 'socket' (female) and

This flagged the go-ahead for the design and production of a single-sided (!) TH/SMD circuit board—the component layout is shown in **Figure 4**. The value of R1 sets the trip voltage, hence is dependent on the battery voltage. The interdependencies are listed in **Table 1**. In prac-

COMPONENT LIST

Resistors
(All 0.25 W, 1%, SMD 1206)
R1 = 2.74kΩ *
R2 = 1.5kΩ
R3,R6,R8 = 3.3kΩ
R4,R5 = 470kΩ
R7 = 1.8MΩ
R9 = 10kΩ
R10 = 0Ω

Capacitors
C1,C2 = 10μF 10% 25V, X5R, 1206
C3 = 470nF 10% 100V, X7R, 1206

Semiconductors
D1 = 1SMB5925B zener diode, SMB (Newark/Farnell # 1894811)
D2 = PMEG6010CEH, Schottky diode, NXP, SOD-123F
 (Newark/Farnell # 1510694)

D3 = SM6T12CA, TVS diode, STmicroElectronics, SMB (Newark/Farnell # 9885870)
D4 = HSME-A401-P4PM1, LED, green, Avago, PLCC-4 (Newark/Farnell # 1058419)
IC1 = LT1716CS5#PBF, comparator, Linear Technology, SOT-23-5 (Newark/Farnell # 1417738)
T1,T2,T3,T4,T5,T6 = IPB017N06N3, N-MOSFET, Infineon, TO-263-7 (Newark/Farnell # 1775519)
T7,T8 = 2N7002, N-MOSFET, Diodes Inc., SOT-23 (Newark/Farnell # 1713823)

Miscellaneous
K1-K4 = high current connectors, male & female pairs, gold plated *
Heatsink, ½ brick form, Aavid Thermalloy type 241204B92200G, dim. 60.96mm x 57.91mm x 11.4mm (Newark/Farnell # 1703176)
PCB 120733-1

* user configurable component, see text
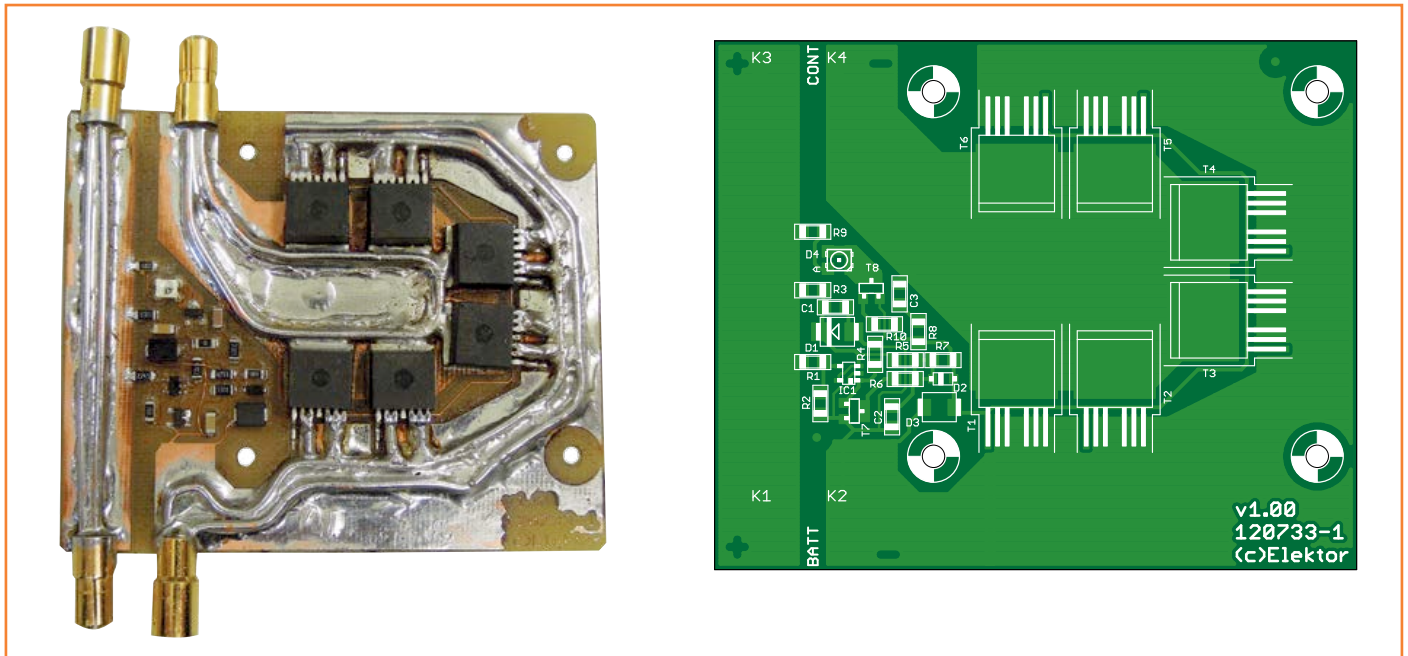
a 'plug' (male) connector on the + and – battery lines. The same can be done on the + and – output lines.

The MOSFETs are flat on the board, and the heatsink is on top of them with thermally conductive sheet material held pressed in between. The heatsink is secured with four corner M3 bolts or screws, with two M3 nuts on each bolt acting as standoffs, i.e. between the board surface and the flat side of the heatsink. The total standoff height is approximately 5 mm. The bolts should be lightly tightened so as to barely compress the heat conductive sheet material.

Although we've talked mostly about motor controllers for R/C models here, the circuit is suitable for any 12-40 V DC load that represents a very low resistance initially, including big electrolytic reservoir capacitors and lamp filaments.

(120733)

[1] AN1542:
www.bonavolta.ch/hobby/files/Motoro-laAN1542.pdf

[2] IPB017N06N3 datasheet:
www.infineon.com/dgdl/IPB017N06N3_Rev2.2.pdf?folderId=db3a30431441fb5d01148ca9f1be0e77&fileId=db3a30431ddc9372011e264a7ab746ea

Figure 4.
The circuit board design is compact and designed for the heatsink to physically cover the MOSFETs. The copper track layout as shown is not suitable for direct use. You have to strengthen all PCB tracks carrying the load current with pieces of solid copper wire.
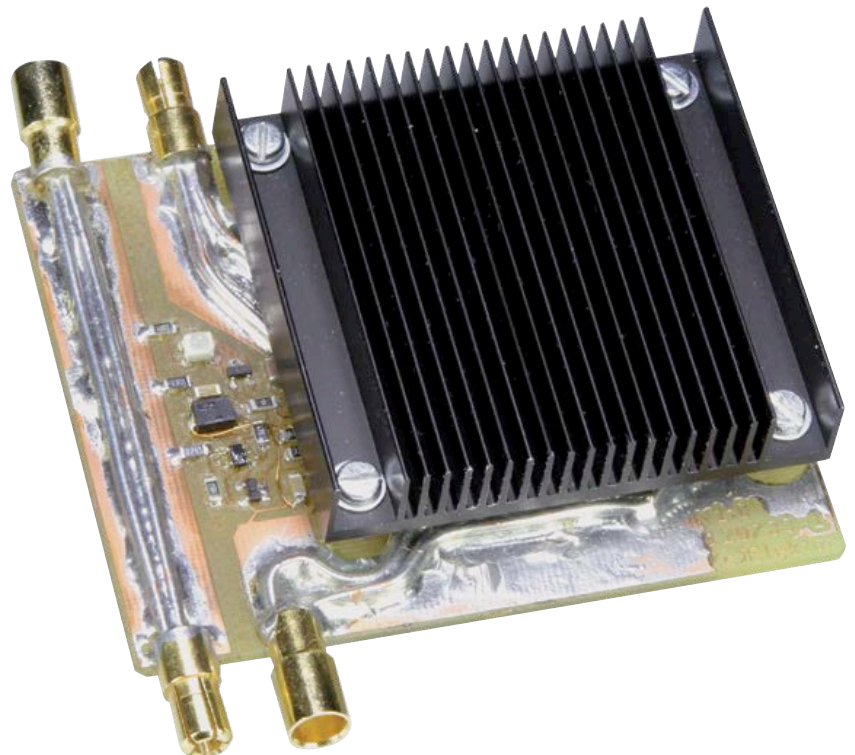
Acoustic Spirit Level / Tilt Alarm

An ATtiny45 design with many uses

By **Jörg Trautmann**
(Germany)

This little project was inspired by a tilt sensor circuit which first appeared in our 2010 Special Projects Edition. The idea was to build a simple multi functional tilt sensor. The resultant design has two main uses; it functions as an acoustic three-axis spirit level or a security movement detector.

The original purpose of this device was to assist in leveling large garden tables on uneven ground. Away from buildings it is difficult to find any reference points to assist in setting up the tables, and conventional spirit levels can be a bit cumbersome. It is however not only useful for leveling tables, it also serves as a security monitor to sense the movement of some object; place it on a table or any other item worth protecting, if anyone tries to move it the alarm sounds and the thief is sent off with a start.

How it works

The circuit shown in **Figure 1** comprises an Atmel ATtiny 45 microcontroller and an MMA7260QT. The MMA7260 is an integrated 3-axis acceleration sensor which has already featured in this magazine back in 2007 and also in the Special Projects (summer) Edition of 2010 where it was used in a large USB tilt sensor with an LCD screen [1]. The small integrated circuit is fixed to a small PCB

(**Figure 2**) and has three analog output signals. The signal produced is proportional to acceleration; at +1 g the output voltage is 2.45 V. The ATtiny45 microcontroller from Atmel [2] includes a number of built-in A/D converters, three of which we use here to measure acceleration or level of tilt from the three sensors. The A/D converters use an internal voltage reference of 1.1 V so it is necessary to scale the three sensor output voltages using a voltage divider network. Based on the sensor parameters the resistors R1, R3 and R5 have a value of 470 kΩ and resistors R2, R4 and R6 have a value of 330 kΩ. The 2.45V maximum output voltage from each of the sensors is thereby scaled down to around 1 V and optimal measurement resolution is achieved in the A/D conversion process.

The microcontroller firmware uses changes in the X, Y and Z parameters to influence the output frequency of three tone generators.
When the sensor is on a flat and level surface the tone generators remains quiet. As soon as one of the sensors detects a movement of more than approximately ±2° on any axis, the tone begins to sound and varies as the tilt increases. The push button S1 is use to calibrate the unit and also to select operational mode. When the button is held down for longer than 5 s the unit is switched into alarm mode.
The voltage on the board is regulated by a low-drop voltage regulator type LP2950CZ3.3, producing an optimal 3.3 V for both the microcontroller and tilt sensor. A 9-V 6LR22 (PP3) size battery will provide enough energy to keep the circuit running for a long time. During testing

Figure 1.
The tiny circuit can be fitted onto a small piece of breadboard.
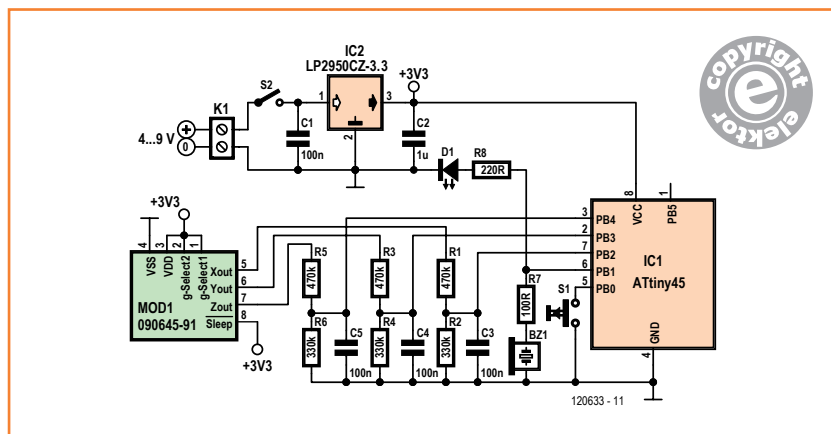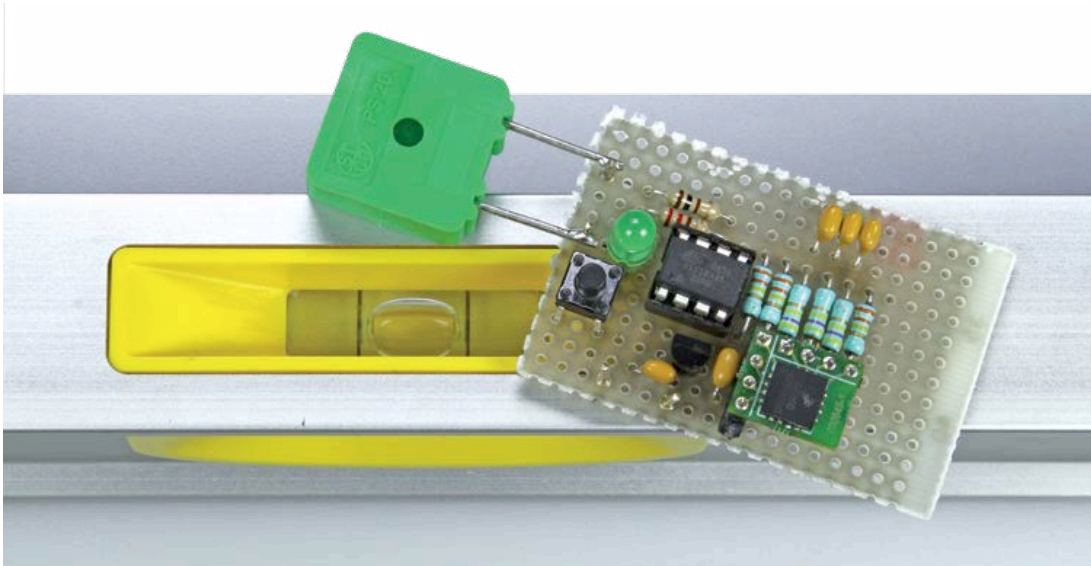
it was found that the circuit would still function with a supply voltage as low as 3.6 V. Maximum current was measured at 4.56 mA and averaged around 3.06 mA with the LED blinking and the tone sounding.

Construction and operation

Construction of the circuit is relatively simple and can be made using small piece of breadboard such as the prototyping board called ELEX-1. When the finished circuit is first powered up the red LED will light continuously and the loudspeaker should remain silent. If this is not the case then remove power and double check your circuit construction. The first time the circuit is switched on it is necessary to carry out a calibration process which will then act as the reference attitude. Place the circuit board on a flat, level surface and hold down button S1 for approximately 1 s. When the push button is released the LED will extinguish indicating that the calibration is complete and the values have been stored. The unit should not be emitting any sounds now. When the board is waggled you should hear three over lapping beep tones. The LED will also flash and the tone frequencies increase as the angle of tilt increases. With the PCB returned to a level position the tones cease and the LED turns off. Once the unit is fitted into a project case you can quickly begin to develop a feel for when the unit is level.

To use the unit as a movement alarm first place it on the device you want to protect (the surface does not need to be horizontal) and press button S1 now hold down S1 again for a few seconds until the LED starts to flash regularly. Once the

push button is released the circuit is primed. Now when the unit is tilted by more than approximately 20 degrees it sets off a loud rising and falling alarm siren. A brief press of S1 silences the siren. The unit still functions in alarm mode until power is turned off. It will always power up in 'spirit level' mode. The most recently stored attitude calibration values are again used as the reference plane.

The program

The firmware for this project is written in BASCOM AVR and can be downloaded from the project web page [3]. Port pin PB1 is configured as an output to drive the piezo buzzer. PB0 is used as an input with its internal pull up resister enabled. The A/D converters ADC0, ADC1 and ADC2 use the internal voltage reference of 1.1 V. When push button S1 is pressed (PB0=Low) for approximately 1 s the measured values are stored in EEPROM and used as the calibration values. The next time

Figure 2.
The sensor chip on the adapter PCB.

the circuit is powered up, the same values will be used as reference. The logic used to evaluate the switch status on PB0 is so programmed that calibration of the unit is only possible when the unit is not in alarm mode. In alarm mode a press of S1 resets the alarm. The button S1 therefore performs two functions.

A full measurement cycle consists of seven readings from each of the three analog channels taken within 210 ms, the values given are then averaged. This method has shown to give excellent measurement accuracy and stability. It is relatively easy to alter the sensitivity of the unit operating in either mode by changing the Trigger_value variables declared in the software.

If you want to use the firmware as it stands and don't feel the need to make any alterations, it's a simple job to order a pre-programmed controller from the Elektor shop. Alternatively, go ahead and program your own micro.

(120633)

Internet Links

[1] www.elektor.com/070829

[2] www.atmel.com/devices/ATTINY45.aspx

[3] www.elektor.com/120633

[4] Low-g acceleration sensor: www.freescale.com/webapp/sps/site/taxonomy.jsp?nodeId=01126911184209#2

2-Wire Interface 3.0

By **John Hind** (UK)

130115 - 11
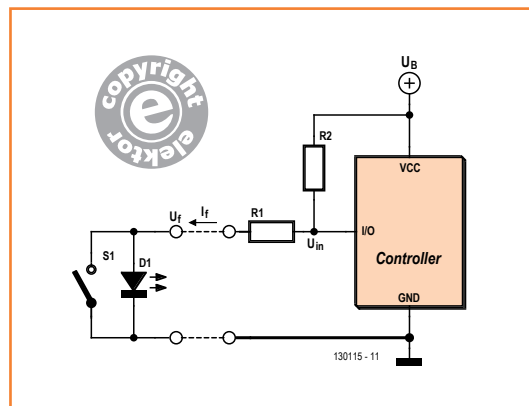
Klaus Jürgen Thiesler's '2-Wire Interface' was published in Elektor magazine in the form of both basic [1] and low-current [2] versions. Each variant used two transistors and a handful of other components to hook up an LED and a pushbutton to a microcontroller. The author still felt driven to simplify this arrangement further and can now provide a solution that, with only two resistors and a single I/O pin, could hardly be minimalized any more. Unless you know better!

Reducing the number of components means the microcontroller has to work somewhat harder. This solution makes the assumption that an I/O pin can be toggled between input and output, which is virtually always the situation. The circuit is able to light the LED as output in 'High' state and assess the status of the switch when the LED is not alight. If you take a look at the state table, the first two lines (numbered 1 and 2) should

Table 1

Status	I/O Pin	State	Switch	LED	I_f	U_{in}
1	Input	High	open	off	$(U_B - U_f) / (R1 + R2)$	$U_f + I_f * R1$
2	Input	Low	closed	off	$U_B / (R1 + R2)$	$I_f * R1$
3	Output	High	open	on	$(U_B - U_f) / R1$	U_B
4	Output	High	closed	off	$U_B / R1$	U_B

Table 2

LED color	U_f	I_f	@ 5 V: R1 | R2	@ 3.3 V: R1 | R2	@ 2.1 V: R1 | R2
Red	1.7 V	10 mA	330 Ω | 470 kΩ	160 Ω | 220 kΩ	39 Ω | 56 kΩ
Orange, Yellow	2.1 V	10 mA	300 Ω | 430 kΩ	120 Ω | 180 kΩ	-
Green	2.2 V	10 mA	270 Ω | 390 kΩ	110 Ω | 160 kΩ	-
Blue, White	3.6 V	20 mA	68 Ω | 200 kΩ	-	-

make things clear. The I/O pin is switched as an input and the voltage U_{in} applied to it, according to the status of the switch, is interpreted as 'low' or 'high' so long as resistors R1 and R2 have been selected correctly to match the supply voltage U_B (most microcontrollers have an upper switching threshold in the region of 0.5 U_B).

So far, so good. But how can the switch be polled in states 3 and 4, when the I/O pin is functioning as an output? Quite simply in fact. Several times a second the pin is turned into an input for an extremely short period. In this way, for less than a blink of the eye, we have state 1 or 2, which thanks to the sluggishness of the human eye (persistence of vision, as it is called) is not even noticeable. If it is established then that the pushbutton switch is depressed, the pin remains in state 2 until it is released again, as state 4 would not make any difference (the LED remaining unlit), other than unnecessary current flows. Following this the controller switches back immediately into state 3, making the LED illuminate again.

In the firmware of the microcontroller we can implement not just basic debouncing but also 'de luxe' functions such as variable brightness for the LED, achieved very simply by toggling rapidly between states 1 and 3. Imagination knows no boundaries here!

The author developed his solution around the PIC16F883 [3]. This type operates with internal pull-up resistors that can be activated exactly as

on the well-known AVR controllers. In principle, particularly if you reduced U_B, you could replace R2 with this internal resistor. Unfortunately these pull-ups have values exclusively in the range from 10 to 50 kΩ, which could lead to the LED lighting dimly (but definitely visibly) in state 1. The firmware therefore enables the pull-up only as long as necessary for polling the pushbutton switch, to ensure this effect is not bothersome.

In any event R2 must be selected so that the switching threshold of the input is definitely exceeded, since the forward voltage U_f drops with small currents. This very effect can become a problem when using a red LED and 5 V operating voltage. In this situation an ordinary silicon diode in series with the LED will help. Dimensioning the resistors relative to the supply voltage and according to LED color is set out in another table. With differing currents you will need to do some calculation.

(130115)

[1] 2-Wire Interface for Illuminated Pusbuttons, Elektor April 2012, www.elektor.com/110572

[2] 2-Wire Interface version 2.0, Elektor January & February 2013, www.elektor.com/120071

[3] Firmware: www.elektor.com/130115

Accurate Universal Measurement Interface
Accuracy—quite simply

By **Michel Defrance**
(France)

Most microcontrollers have a built-in digital-to-analog converter (DAC), but what can we do when this isn't accurate enough? Look no further: the solution is right in front of you.
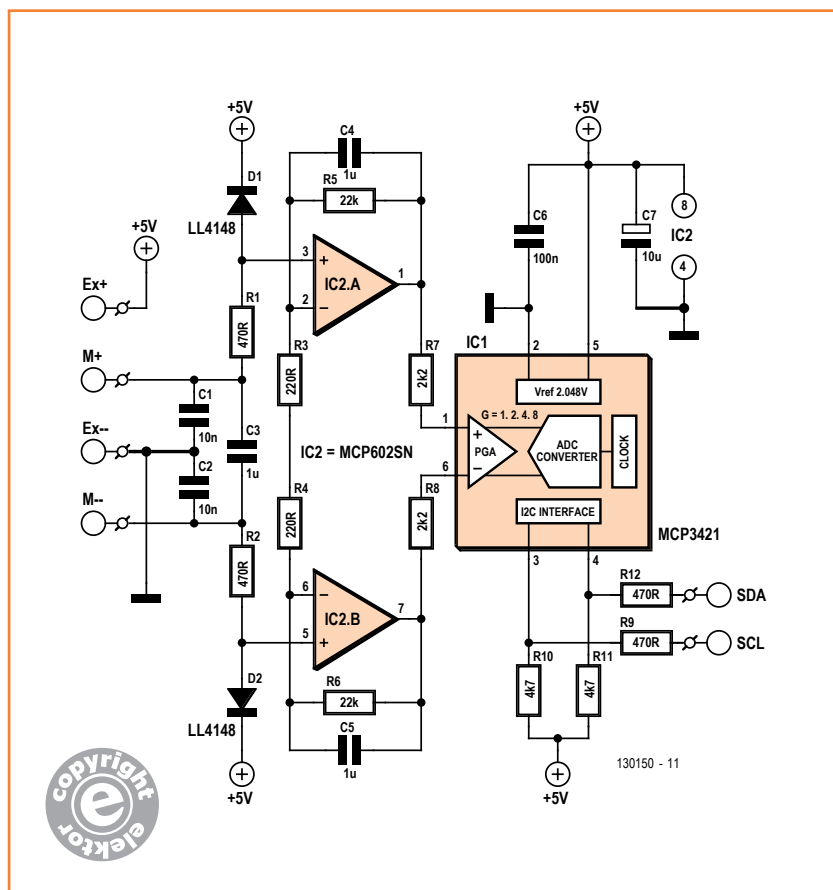

Figure 1.
Circuit of the measurement interface.

Electronics engineers very often need to measure low voltages with great accuracy, for example, the voltage from a pressure or temperature detector, or the output voltage of a Wheatstone bridge (often of the order of a millivolt). Microcontrollers are handy and fairly easy to program, but everyone knows that the accuracy of their converters isn't brilliant (often 8/10/12 bits). In addition, to measure low voltages, these need to be processed, often with the help of op amps.

Confronted yet again by this same problem, I decided to find a solution that would be both satisfactory and reusable. The inspiration was there alright, all it now needed was the perspiration bit. The requirements were clear: I needed an interface that would be easy to connect to any microcontroller and easy to put together using common, cheap components. Two ICs in the Microchip catalogue caught my eye:
- the MCP602 high-performance amplifier;
- the MCP3421 18-bit programmable ADC with I²C interface, built-in 2.048 V reference, and programmable amplifier.

Instrumentation amplifier

When we want to process low voltages for digitizing, we often use a configuration called an instrumentation amplifier, built using three op amps. Here, I'm using the MCP602's two op amps IC2A and IC2B in a differential configuration to drive the programmable-gain amplifier in the MCP3421. The output voltage of this first stage will be proportional to the voltage difference between the circuit's measuring inputs M+ and M− (**Figure 1**). The gain of the first stage is given by:

$$G1_+ = 1 + R5/R3$$

Table 1. Input voltage range as a function of MCP3421 gain.	
MCP3421 gain	**Input voltage range**
1	−20 mV to +20 mV
2	−10 mV to +10 mV
4	−5 mV to +5 mV
8	−2,5 mV to +2,5 mV

$G1_- = 1 + R6/R4$

Since we want the amplification of the voltage at the M+ input to be the same as that for the M− input, we'll choose R5 = R6 and R3 = R4. With the values shown on the circuit, we'll have:

$G1 = 1 + 100 = 101$

Make sure you select 1% tolerance resistors for R3–R6, otherwise you risk having a serious asymmetry in the input stage. Let's move on to the second stage, built around amplifier/ADC IC1. Used in symmetrical mode, it accepts voltages from −2.048 V to +2.048 V between its pins 1 and 6. Since the gain G2 of the amplifier it contains is software-programmable, it will be possible to select different ranges for the input voltage from the first stage. The total gain of the circuit $G = G1 \times G2$ will thus vary between 101 for G2 = 1 and 808 for G2 = 8.

Table 1 gives a list of the possible values. If these ranges don't suit you, change the value of R3–R6.

Powering

The interface is powered at 5 V. In the presence of weak signals, noise from the power supply can become a problem—and this circuit is no exception. Whatever type of supply you choose, it must be accurate and generate as little noise as possible. Using 18-bit accuracy conversion, the slightest supply noise will interfere with your measurements. I also recommend using a soft-start power supply so as to reduce drift due to variations in component characteristics. It would be possible to use software delay timing for this, but that wouldn't benefit all the components on the board. In [1] I suggest just such a power supply, based around an MIC2941 low-loss regulator from Micrel.

Construction and use

Construction of the 35 × 25 mm PCB (**Figure 2**) ought not to cause any problems for readers familiar with SMDs. If you design your own PCB, to obtain optimum performance, do adhere to the advice given in the MCP3421 data sheet. This is also very helpful when it comes to using the project. Watch out for the MCP3421's I²C address: this depends on the exact type number of the device you buy. This is also detailed in the data sheet.

Specifications

- 18-bit conversion
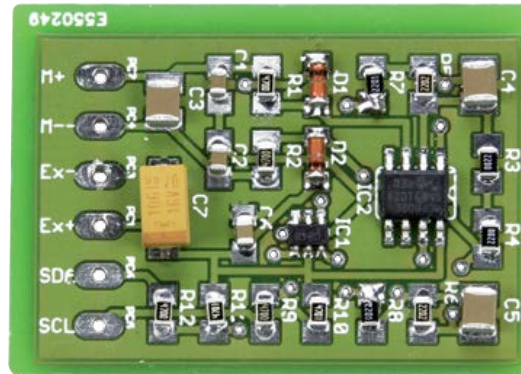- I²C interface
- Software-programmable gain

Figure 2.
View of the author's PCB.

By way of an example, I've developed an application around a PIC18F452 (or PIC16F876A) microcontroller which displays the voltage read from the MCP3421 via the I²C bus on the LCD. You'll find that elsewhere in this issue. It measures the low voltage (a few millivolts) from a strain gauge wired into a Wheatstone bridge. The PIC receives the digitized voltage from the MCP3421, and the PICBASIC program converts the value into pressure. The ADC output voltage and the pressure then appear on an LCD. This program shouldn't be too hard to port to an Arduino, for example. You'll be able to use this inexpensive circuit (less than $20) in lots of different projects. And you won't have a reason any longer to curse the ADC in your favorite microcontroller.

(130150)

Internet Links

[1]   www.elektor-labs.com/node/3053
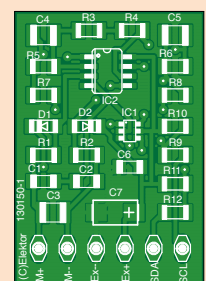
COMPONENT LIST

Resistors
R1,R2R9,R12 = 470Ω 5%
R3,R4 = 220Ω 5%
R5,R6 = 22kΩ 1%
R7,R8 = 2.2kΩ 5%
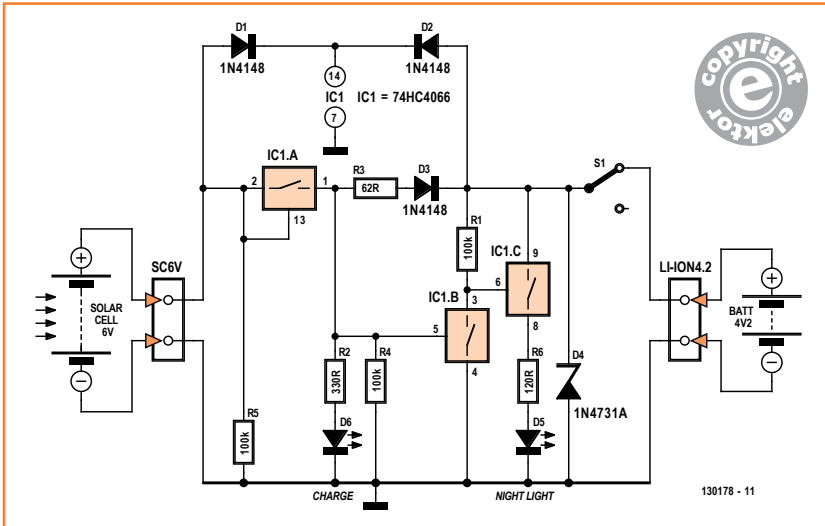R10,R11 = 4.7kΩ 5%

Capacitors
C1,C2 = 10nF 50V 10%
C3,C4,C5 = 1µF 50V 10%
C6 = 100nF 50V 10%
C7 = 10µF 16V 10%, electrolytic

Semiconductors
D1,D2 = LL4148
IC1 = MCP3421A1T
IC2 = MCP602SN

Solar-Powered
Night Light with Li-ion Backup

By
Michael A. Shustov
(Russia)

This night light has two power sources: a solar cell with a peak output voltage of about 6 V, and a Li-Ion cell with a voltage between 3.7 V and 4.2 V. Three (of four) electronic switches in a 74HC4066N (IC1) control the device operation. IC1 gets its supply voltage through diode D1 or D2 depending on which power source supplies the highest voltage. Consequently the 4066 gets any value between 3.7 V and 6 V to operate off.

At daytime the voltage supplied by the solar cell reaches the peak value typically around 6 volts.

IC1a is closed due to the High level at its control input (pin 13), so the Li-ion battery gets charged with about 10 mA through resistor R3 and diode D3 connected in series. At the same time LED D6 lights to indicate the battery is being charged. Switch IC1b is closed too, causing switch IC1c to be open and LED D5 to remain dark.

If the voltage supplied by the solar cell drops below 1/3 of IC1's supply voltage, i. e. below 1.3 V or thereabouts, switch IC1a opens and the 'Charge' LED goes out. The voltage at the control input of switch IC1b drops to zero, causing he switch to open. Consequently switch IC1c closes, connecting the 'Night Light' LED to the battery through resistor R6, which sets the LED current to 10-13 mA. Feel free to select the color—the prototype had a green LED.

The battery charging rate as well as the intensity of the LEDs may be adjusted by adapting R3, R2 and R6, observing a maximum current of 20 mA through the '4066 switches. Zener diode D4 prevents excessive battery charge voltage levels. Switch S1 when opened prevents the battery from being discharged when the circuit is in storage, or not in use for some reason.
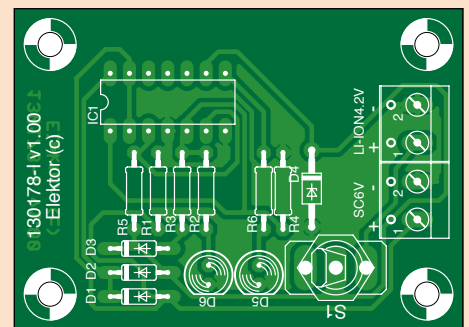
(130178)

COMPONENT LIST

Resistors
R1,R4,R5 = 100kΩ 1% 0.25W
R2 = 330Ω 1% 0.25W
R3 = 62Ω 1% 0.25W
R6 = 120Ω 1% 0.25W

Semiconductors
D1,D2,D3 = 1N4148
D4 = 1N4731A zener diode (4.3V)
D5 = LED, 5mm, color of choice
D6 = LED, red, 5mm
IC1 = 74HC4066

Miscellaneous
S1 = toggle switch, Newark/Farnell # 1310879
PCB # 130178-1

Connectors marked SC6V and LI-ION4.2V = PCB terminal block, lead pitch 5mm

Need to Stay Connected?
Learn About Microchip's USB and Networking Solutions

USB

USB technology can be found in practically all applications and markets—from consumer, industrial and automotive segments. The extreme proliferation of USB has even led to the adoption of the technology as an embedded chip-to-chip high bandwidth interface. Microchip has been enabling such uses, applications and markets with seamless USB connectivity by delivering integrated value rich solutions, such as USB hub controllers, power delivery and charging, transceivers/switches, Flash media controllers and security.

Ethernet

Ethernet devices have become ubiquitous in communications and networking products, servicing a wide variety of applications across multiple market segments. This well understood technology provides a robust link to ensure reliable communication between devices in a network. Microchip has a broad portfolio of reliable, high quality and high performance Ethernet solutions. From Ethernet Switches, Controllers, Bridges, and PHYs, to a variety of standard interfaces, serving consumer, industrial, and automotive applications, Microchip can provide a solution to address your varied application needs and offers you the support needed to reduce your time to market.

GETTING STARTED IS EASY

If you need USB or Networking connectivity in your product design, Microchip's broad line of proven and reliable solutions cover all main market segments and applications including:

- Consumer electronics
- Mobile
- Industrial equipment and control
- Automotive
- Medical

Visit www.microchip.com/usbsmsc or www.microchip.com/ethernetembeddednetworkingsmsc for more information.

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

CDI Ignition

For Spartamet and Saxonette mopeds

By **Jan Visser**
(Elektor Labs)

This article describes a home-made CDI unit for Spartamet and Saxonette motor-assisted bicycles (mopeds).

Having been virtually forced to use a Spartamet to travel between home and work for three weeks, it was noticeable that although the moped ran fine, at full throttle and at top speed (15 mph) the ignition began to misfire. The fuel consumption at full throttle also increased dramatically: from 118 mpg at 3/4 throttle to 71 mpg at full throttle. There was a strong suspicion that the higher fuel consumption was related to the misfiring of the ignition; this was confirmed after some further thought and having checked the spark plug and exhaust after several rides.

The ignition starts to skip sparks when the 30 cc two-stroke engine is at full throttle and at top speed. The manufacturer has used this method on purpose to build in an electronic speed limiter to ensure the moped is road legal. However, the carburetor is not limited and it happily continues to deliver the fuel mixture, which ends up unburnt in the exhaust. Apart from the fact that this has a negative impact on the fuel consumption, it doesn't do the exhaust any good either. There will be more of a carbon build up in the exhaust, which means it has to be replaced sooner.

You could of course open up the existing CDI (Capacitive Discharge Ignition) unit and modify it, but since this is completely encased in potting compound this is not something we would recommend. Instead, we investigated what was required to produce the sparks without limiting the rpm. The result of this can be seen in the schematic shown here.

Since the ignition coil and pickup coil are mounted next to the flywheel of the engine we only have to concern ourselves with the electronics that make a capacitor discharge into a coil at just the right moment.

The input is connected to a pickup coil that delivers a single pulse for every revolution of the flywheel. The output is connected to the ignition coil that supplies the high voltage pulse to the spark plug. Capacitor C1 stores the electrical energy and is charged up via D3. When there is
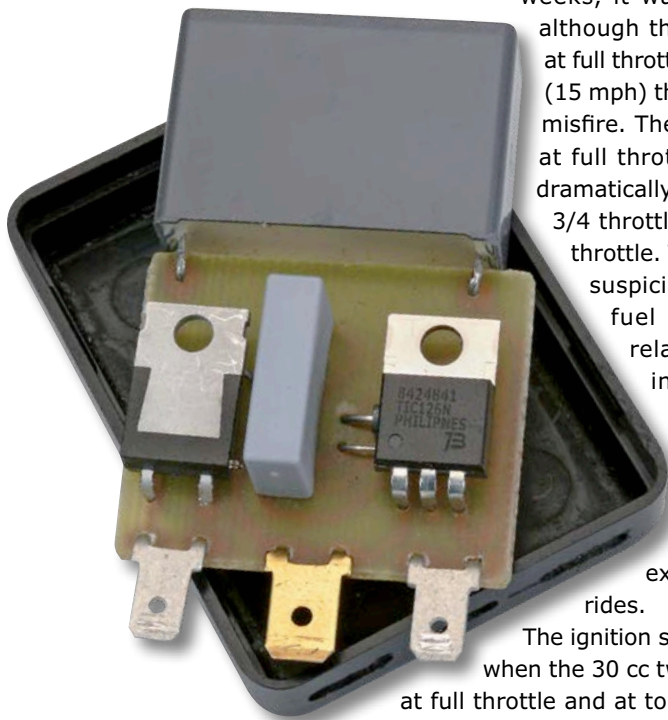
a pulse at the input it triggers the thyristor into conduction, which connects C1 to ground so it can discharge into the ignition coil. That is all there is to it!

A single sided PCB has been designed for the circuit (the layout can be downloaded from [1]). However, note that the components are mounted on both sides of the board. This was necessary in order to keep the circuit the same size as the original CDI unit. Its dimensions are 59x38x24 mm. The photos of the prototype make this clearer. First mount D1, D2, DI1 and C2 onto the component side. You should then solder diode D3 and thyristor TH1 onto the board. These should be bent over so they're level with the board, with D3 ending up on top of D2 and T1 on top of D1 and DI4. The MKP capacitor (C1) ends up alongside the board. The varistor (VR1) and resistor (R1) are then mounted onto the solder side of the board. And finally you should solder the three spade terminals onto the board.

For the enclosure you can use a small box from Hammond (001100), Conrad Electronics part number 540830-89, although an acrylic home-made box (cassette or cd case) is an alternative. Once the board has been populated and connected you can check if the ignition produces any sparks. If it all works and the spark plug is sparking happily you can put the circuit in its enclosure and fill it with potting compound. If you fail to do this it is very likely that the circuit will soon stop working properly, since the ignition is subject to quite a lot of vibration.

There are two types of CDI unit in use, one made by Motoplat (red) and one made by Prüfrex (blue). In both cases the earth is connected to the middle connector of the CDI unit. If you accidentally connect the input and output the wrong way round the CDI unit won't produce a spark. All you need to do when this happens is to swap the red and blue wires over.

When the circuit was installed and put into use the effect was immediately noticeable. The engine runs much smoother at full throttle and it no longer misfires. The average fuel consumption was also found to have improved considerably to 166 mpg.

Since the little engine has its own mechanical limitations (carburetor, exhaust, compression ratio), the top speed won't increase by a huge amount: we found it to be about 2 to 2.5 mph higher. The biggest advantages are of course the

120601 - 11
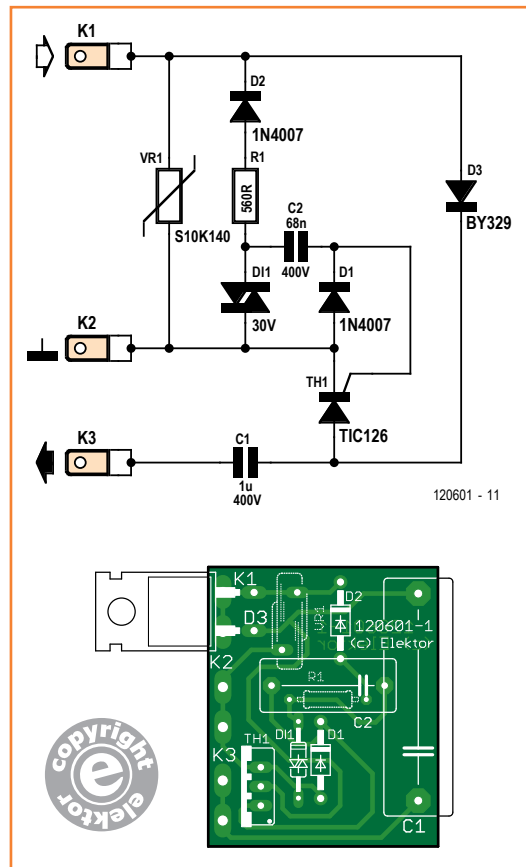
COMPONENT LIST

Resistors
R1= 560Ω
VR1 = S10K140 varistor

Capacitors
C1 = 1µF 400V MKP
C2 = 68nF 400V MKS

Semiconductors
D1,D2 = 1N4007
D3 = BY329
DI1 = diac D30 (alternative: ER900 or DB3)
TH1 = TIC126N

Miecellaneous
3 pcs 6.3-mm (0.25'') Fast-on (spade) terminal plugs, PCB mount
PCB 120601-1, see [1]

better running of the engine and the improved fuel consumption.

(120601)

Internet Link

[1] www.elektor.com/120601

Connection Details for the CDI Unit

Motoplat: black coil with a red CDI unit
Prüfrex: blue/gray/red coil with a blue CDI unit

Motoplat	**Prüfrex**
a = yellow	a = black
b = blue	b = red
c = red	c = blue

The connection details are shown on top of the CDI unit. Should you connect the red and blue wires the wrong way round you won't get a spark, and you won't damage the coil or CDI unit.

Simple Servo Tester
Basic test gear for modelers

By **Bernhard Kaiser** and **Michael Gaus** (Germany)

When a servo motor starts to malfunction there is generally not much to see from the outside to help diagnose the problem. That's why every modeler's toolbox should have one of these handy units!
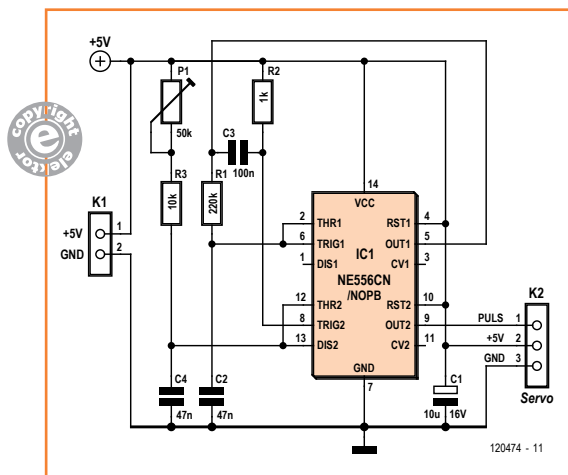

Figure 1.
The circuit uses a dual timer IC.

Servos are one of the basic components used in all branches of model building. They are small, lightweight, low cost and are remarkably easy to control. Model building servos connect directly to an RF receiver unit. They typically have just three connections: positive supply (+5 V), ground (GND) and control (Pulse) lead, which supplies a control signal to move the servo arm. The signal on this lead is pulsewidth modulated and supplied by the receiver. Positive pulses with a length of 1 ms cause the servo arm to move fully to one end of its travel while 2 ms pulses move the arm fully in the opposite direction. Pulse widths between these limits move the arm to an intermediate position proportional to the pulsewidth. A pulsewidth of 1.5 ms centers the arm. The pulse repetition rate is approximately 20 ms i.e. 50 Hz but this rate is not too critical.

When you suspect that the model is not behaving as it should it could be a problem with the remote control transmitter, receiver or a servo motor. This handy unit allows you to quickly test the servo and eliminate it (or otherwise) from your lines of enquiry. This pulse generator design

COMPONENT LIST

Resistors
R1 = 220kΩ
R2 = 1kΩ
R3 = 10kΩ
P1 = 50kΩ linear potentiometer

Capacitors
C1 = 10µF 16V, 7.5mm pitch
C2,C4 = 47nF
C3 = 100nF
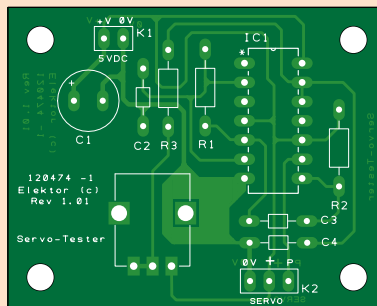
Semiconductors
IC1 = NE556CN

Miscellaneous
K1 = 2-pin pinheader, 0.1'' pitch
K2 = 3-pin pinheader, 0.1'' pitch
PCB 120474-1
DesignSpark-project files can be downloaded from [1].

Figure 2.
The ready made PCB makes circuit construction a cinch.

shown in **Figure 1** is one of the basic bread and butter circuits known to almost all engineers.

A Two timing circuit

The pulse generator is made up of a dual timer chip type NE556, the output pulse width is controlled by the position of a potentiometer. The combination of resistor R1 and capacitor C2 in timer 1 of the NE556 produces the repetition rate of the pulse. This timer output signal at pin 5 has an approximately symmetrical mark space ratio. The negative going edge of the output signal is used via C3 to trigger the second timer which then produces a positive going output pulse at pin 9. The width of this pulse is defined by the values of capacitor C4 and the combined resistance of R3 and P1. Pot P1 thereby gives control of the pulse width.

During tests it was found that the circuit with the component values specified here produced a pulse width in the range of 0.5 to 2.6 ms which more than covers the standard pulse width range used by these types of servomotors. For this reason P1 should not be turned fully to either end of its travel otherwise the connected servo will go past its intended end position and hit the mechanical stops, possibly damaging the servo. Before the circuit is powered up ensure that the control knob P1 is roughly mid position. The pulse repetition rate of the circuit was found to be 18 ms.

The vast majority of servos operate with a supply in the range of 4.8 to 6 V. Here the operating voltage is in the range of 5 to 6 V which can be supplied by four AA primary cells or rechargeables. To make a neat job and simplify construction we have made a PCB for this design (**Figure 2**) which is available from the Elektor Shop [1]. All components have standard (non SMD) outlines so fitting the components should not pose any problems.

(120474)

Internet Link

[1] www.elektor.com/120474

Slow-Start Stabilizer

By Michel Defrance
(France)
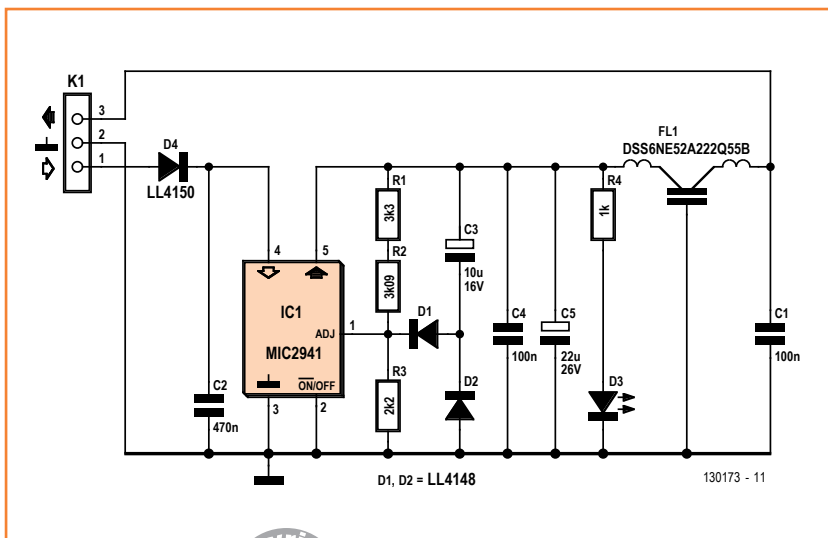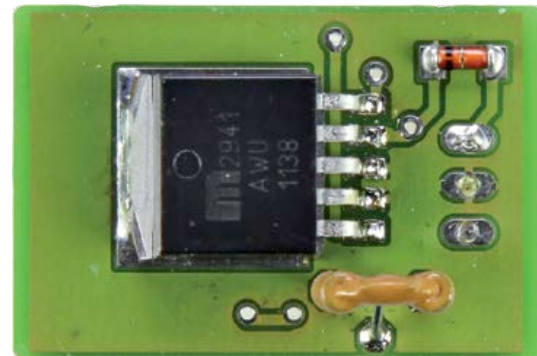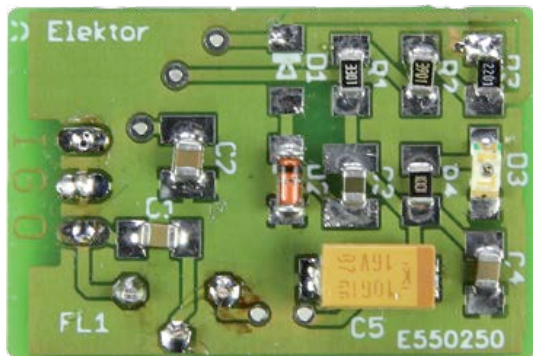
Your everyday 7805 regulator is not the best choice for powering accurate measurement circuits and A/D converters, mainly because it generates too much noise, and exhibits spurious behavior at power on. Taking our Universal Precision Measuring Interface as an example we have a type MCP3421 A/D converter with a resolution of 18 bits. To be able to exploit the high resolution to the last bit, the supply rail must be absolutely stable and free of noise. In addition, the supply voltage should rise slowly when turned on, allowing the components in the measuring circuit to stabilize in terms of bias voltages and temperature. Of course, that can also be accomplished by using a software timer, but doing so has an effect for a couple of components only.

The circuit described here meets all conditions mentioned and can easily serve as a replacement for an ordinary 7805, because the circuit board has about the same size, and the connections are identical. That does mean however that everything got designed in SMD technology due to limited space.

The regulator used is a MIC2941 from Micrel. It's is a low-dropout regulator in which the output voltage is set using a resistance divider, just as with an LM317. The design is simple but effective. The supply voltage is set by $(R1 + R2)/R3$, resulting in 5 V here. Diode D4 serves as polarity protection. Furthermore, a bunch of capacitors is present for decoupling and noise suppression. At the output an EMI filter is included (FL1). The DSS6NE52A222Q55B is a 3-pin component from Murata, containing two coils separated by a capacitor to ground.

The delayed appearance of the supply voltage is accomplished by capacitor C3. When the supply is switched on, initially the voltage at junction R2/R3 remains at virtually 0 V. Next, the capacitor is charged via R3 charged in a about 20 ms, causing the output voltage to rise slowly (see screendump). Diodes D1 and D2 prevent negative voltage ending up at the regulation input, causing the capacitor to be discharged via R2.

The circuit can deliver an output current of at least 1 A. With no cooling however a few tens of mA are possible at an input voltage of 12 V. The

COMPONENT LIST

Resistors (SMD 1206)
R1 = 3.3kΩ
R2 = 3.09kΩ
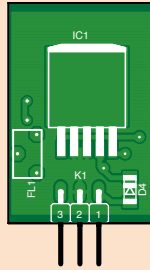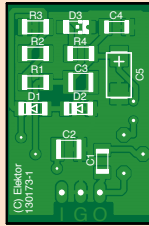R3 = 2.2kΩ
R4 = 1kΩ

Capacitors
C1,C4 = 100nF (SMD 1206)
C2 = 470nF (SMD 1210)
C3 = 10µF 16V (SMD 1210)
C5 = 22µF 10V (SMD 2312)

Semiconductors
D1,D2 = LL4148
D3 = LED, low current, shape 1206
D4 = LL4150
IC1 = MIC2941AWU TR (TO-263)

Miscellaneous
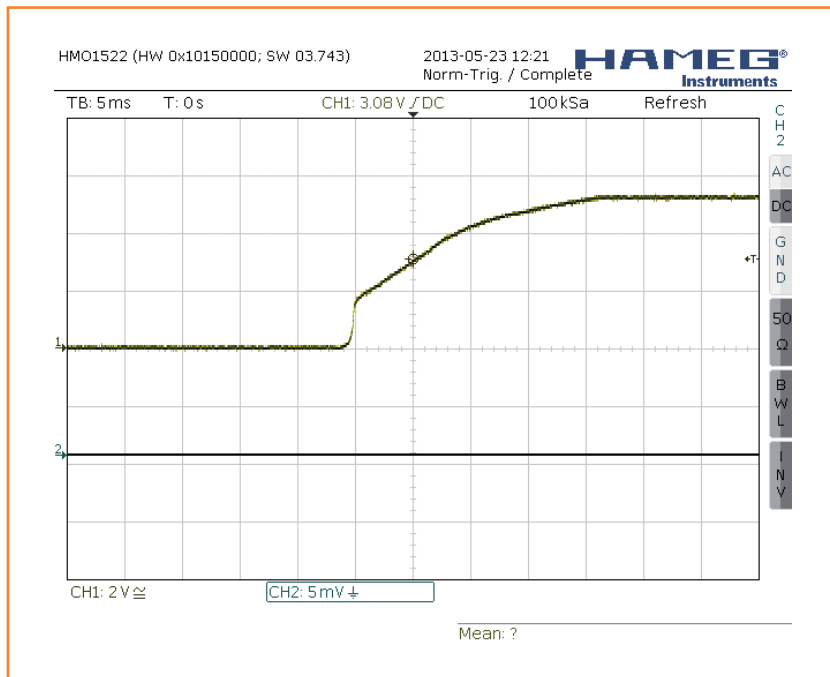EMI suppression filter type DSS6NE52A222Q55B (Murata)
PCB # 130173-1, see [1]

PCB artwork is available as a free download of the Elektor website [1].

(130173)

Internet Link

[1] www.elektor.com/130173

HMO1522 (HW 0x10150000; SW 03.743)   2013-05-23 12:21   **HAMEG®**
Norm-Trig. / Complete   Instruments

TB: 5ms   T: 0 s   CH1: 3.08 V ⌁ DC   100kSa   Refresh

CH1: 2 V ≅   CH2: 5 mV ⌁

Mean: ?

8x Relays and Much More

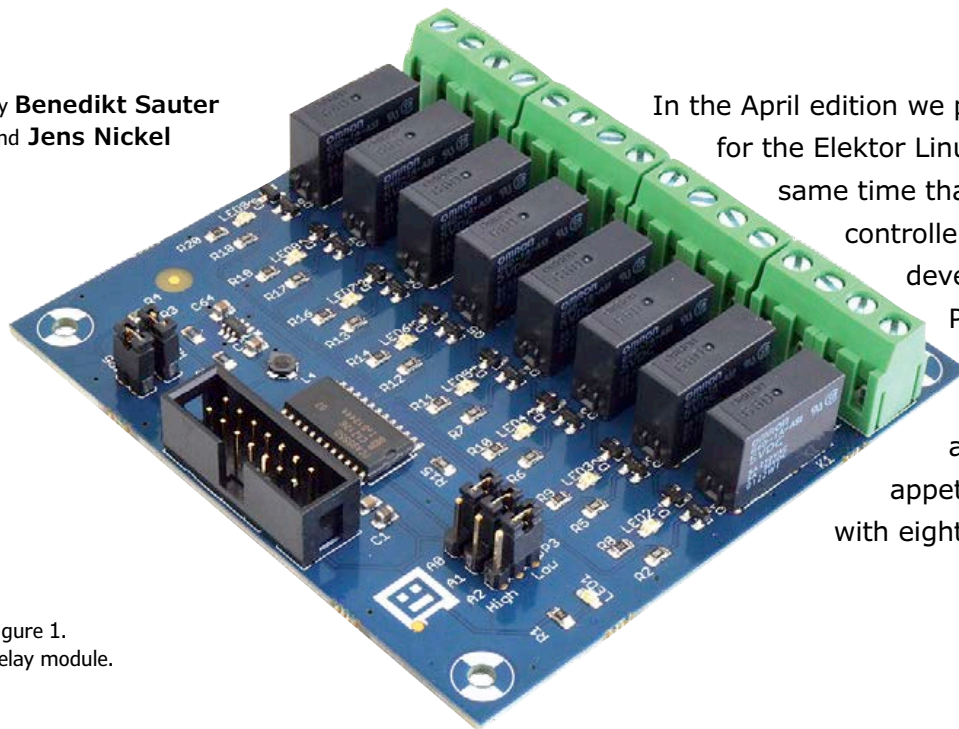## Expansion modules for Linux and other controller boards

By **Benedikt Sauter** and **Jens Nickel**

In the April edition we presented an expansion PCB for the Elektor Linux board and mentioned at the same time that this could be used with other controller boards as well. Meanwhile the development team at Embedded Projects has been working hard designing a whole raft of extra extension boards that are available from Elektor. As an appetizer, so to speak, here is a card with eight relays.
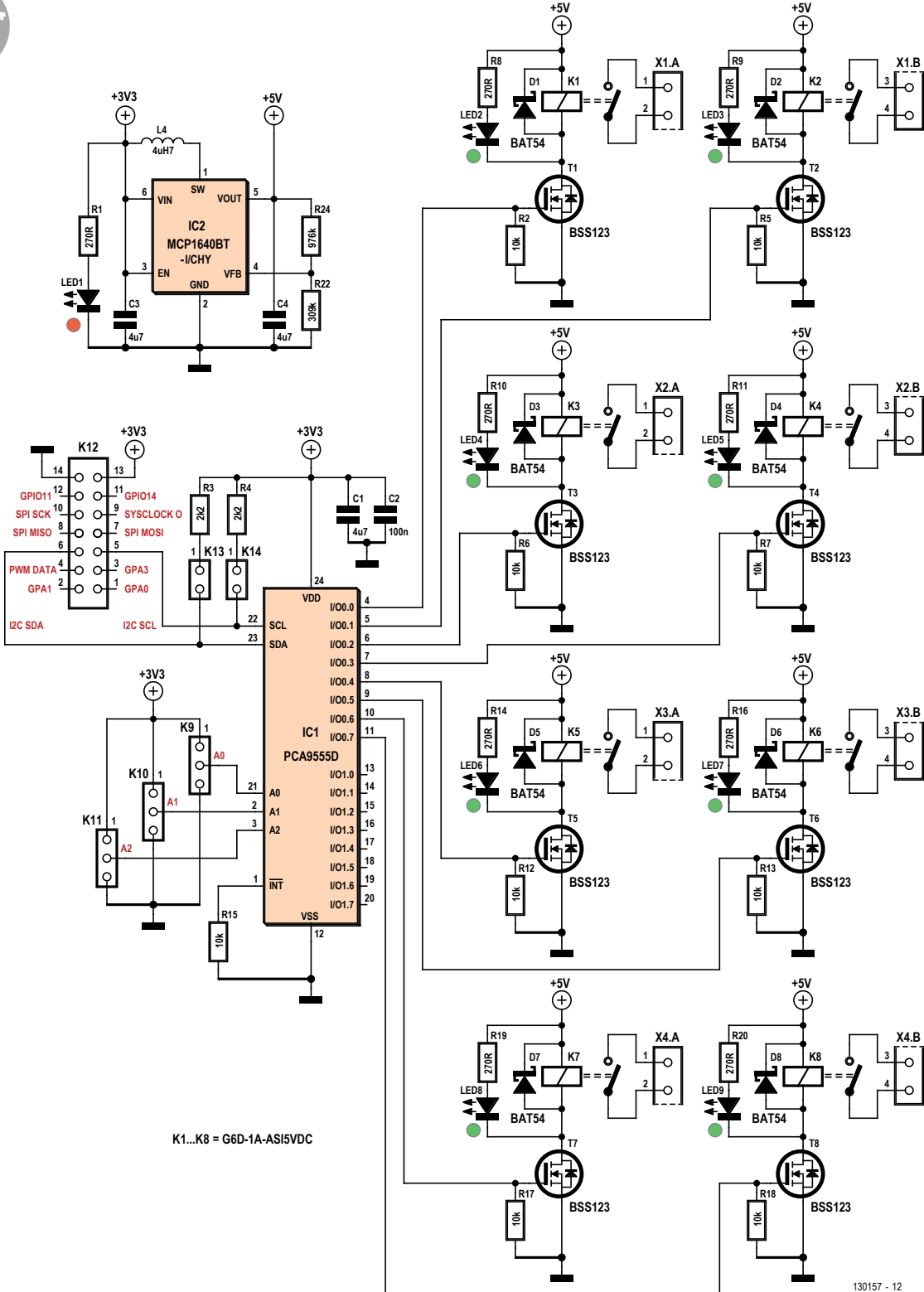

Figure 1.
Relay module.

The relay module (**Figure 1**) is driven using the 14-pin *Gnublin Connector*, just like the Linux Extension Board featured back in April. As an *Embedded Extension Connector* it is also used on the Xmega Webserver Board from Elektor (see next issue) and we are planning further controller boards equipped with this connector. The expansion board is ideal for newbies for whom Linux is (still) too complicated and equally suited to 'power users' who prefer to develop their programs using 'bare metal', in other words without the aid of an operating system.

The relay board, like the Elektor Linux board, originated from the firm Embedded Projects, run by Benedikt Sauter [1]. It's one of a whole series of expansion boards (see boxout) that all mate with the EEC connector mentioned above, pro-

viding as it does pins for SPI, I^2C, PWM, analog inputs and digital inputs/outputs. On the other hand, Elektor Labs have been working on modules using a 10-pin expansion connector (*Embedded Communication Connector*) for UART/TTL connections with one another (see boxout). A small zoo is growing up of controller boards and expansion boards that can be combined flexibly—microcontroller fans can look forward to an interesting time over the coming six months!

Relay card
The circuit diagram of the relay card can be seen in **Figure 2**. As with the Linux Extension Board, a port expander PCA9555 (IC1) addressed over the I^2C bus increases the number of digital outputs available to 16, of which eight are used here. The address of the I^2C module can be preset using

Figure 2.
Circuit diagram of the relay module.

130157 - 12

Expansion modules for the Gnublin/ Embedded Extension Connector (selection)

8x relays (130212-91)
4x20 text display (130212-92)
Stepping motor driver (130212-93)
I/O expander (130212-94)
Temperature sensor (130212-95)
Distributor board 'Bridge Module' (130212-71)
Raspberry Pi adapter 'GnuPi' (130212-72)

These and other boards are available from Elektor [2]. In each case the boards come with SMD components preinstalled, with leaded components supplied in the kit for DIY assembly.

Raspberry Pi Adapter

The Raspberry Pi adapter board 'GnuPi' increases the system's flexibility even further for using expansion boards. It plugs directly into the Raspberry Pi and in turn enables the use of a range of Gnublin/ EEC plug-in connectors [2]. In this way all the expansion boards shown can also be used with the new computer platform that's all the rage. This is neat: the C/C++ API from Embedded Projects can also be used with the Raspberry Pi. To convert a Gnublin/Elektor Linux board application to work on the Raspberry Pi all you need do is alter one single line of code:
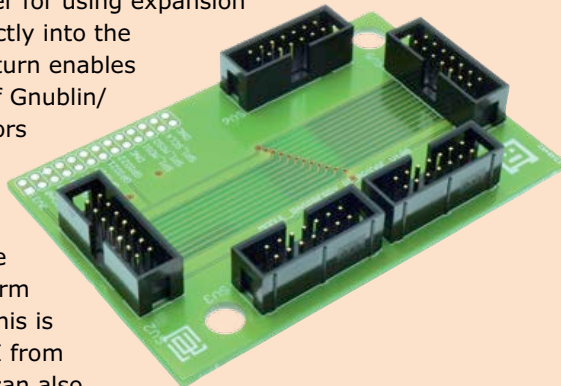
```
#define BOARD_GNUBLIN  →  #define BOARD_RASPBERRYPI
```

Expansion modules for the Embedded Communication Connector

RS485 interface (under development)
RS232 interface (planned)
433 MHz radio module (under development)
Bluetooth using the BTM-222 (planned)
WLAN using the WizFi220 (planned)
USB using BOB (planned)

More on this subject at the Elektor.Labs website [8].

jumpers K9 to K11. Pull-up resistors for the I²C bus can be implemented using K13 and K14. The digital outputs IO 0.0 to IO 0.7 of the port expanders each drive a FET, with in turn operates a relay. An LED for each serves as status display. The *Gnublin/Embedded Extension Connector* is equipped with a 3.3 V pin, which enables the expansion boards to be powered from the controller board. A step-up converter (IC2) is built in to provide the 5 V coil voltage for the relay.

A length of flat ribbon cable serves to connect the controller board and the relay board. The Embedded Projects development gang have already thought about the option of hooking up several expansion boards at the same time; a distributor board (**Figure 3**) is available from Elektor too [2].

C/C++ API
We have already shown in [3] and [4] how you switch the outputs of the port expander ICs in Linux. But there is now an even simpler option. Benedikt Sauter and his comrades-in-arms have written a complete C/C++ API for controlling the expansion cards with great ease. You can incorporate the functions in programs that you write yourself but a number of short command line tools are also available. More on this in the next edition, in which we'll introduce the other extension boards.

A first taster of the C/C++ API is given in the listings. Listing 1 shows how you can access the digital inputs and outputs of the Elektor Linux board easily. Listing 2 demonstrates how you can read in values via the analog input. And Listing 3 shows how you operate the relay card.

The new API [5] also clarifies the route into the world of Embedded Linux for beginners, managing without the more complex features of C like, for example, the Pointer. For the names of functions the developers have in part borrowed from the corresponding Arduino functions. If you're interested, you are of course welcome to cast a glance at the source code [6].

Debian for the Elektor Linux board
Not only has the Gnublin Linux system been enhanced with new hardware but the software side has also been updated. If you feel inclined,
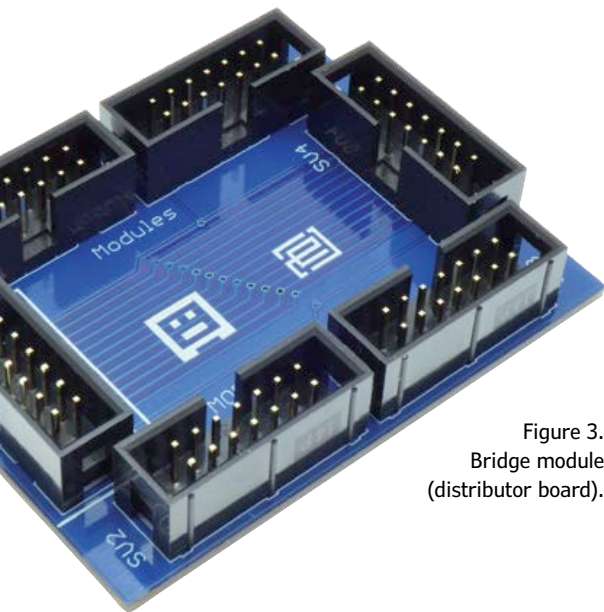
Figure 3.
Bridge module
(distributor board).

**Listing 1: Digital output control
on the Elektor Linux board.**

```
#define BOARD_GNUBLIN
#include "gnublin.h"

int main()
{
    gnublin_gpio gpio;

    gpio.pinMode(3,OUTPUT);

    while(1){
      gpio.digitalWrite(3,HIGH);
      sleep(2);
      gpio.digitalWrite(3,LOW);
      sleep(2);
    }
}
```

you can now equip the Elektor Linux board with
a Debian system (in place of the ELDK file sys-
tem). Debian is very easy to install on the SD
card (it doesn't matter whether you are using the
8 MB or 32 MB version of the board); you can
find instructions on the Internet [7].

(130157)

Internet Links

[1] sauter@embedded-projects.net

[2] www.elektor.com/gnublin

[3] www.elektor.com/120596

[4] www.elektor.com/120518

[5] http://wiki.gnublin.org/index.php/API

[6] https://github.com/embeddedprojects/
gnublin-api

[7] http://wiki.gnublin.org/index.php/
GNUBLIN-Elektor

[8] www.elektor-labs.com/ECC

Listing 2: Reading a value on the analog input.

```
#define BOARD_GNUBLIN
#include "gnublin.h"

int main()
{
    gnublin_adc ad;

    while(1){
        printf("AD value %i \n",ad.getValue(1));
    }
}
```
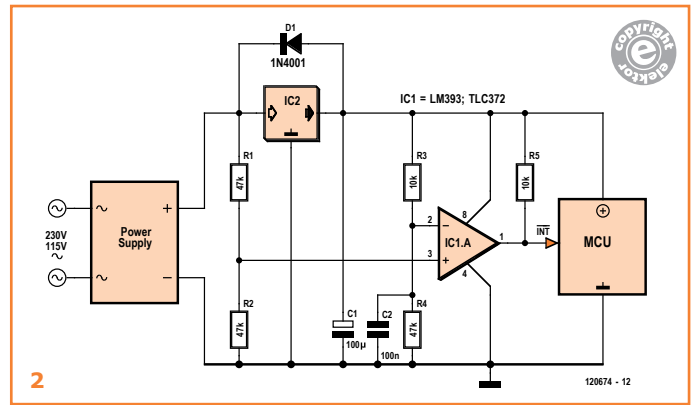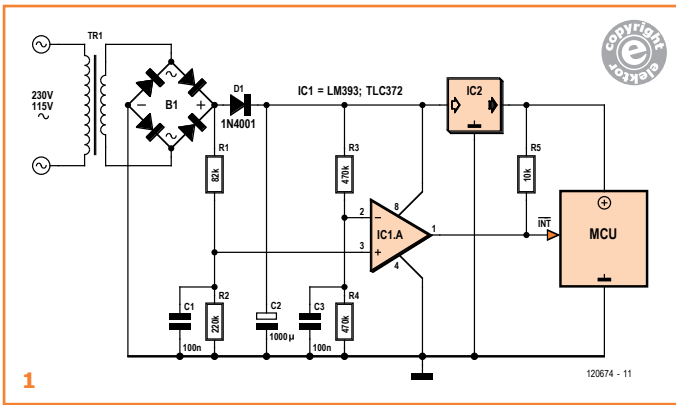
Listing 3: Driving the relay card.

```
#define BOARD_GNUBLIN
#include "gnublin.h"

int main() {
  gnublin_module_relay relay;

  relay.setAddress(0x24);
  relay.switchPin(4, ON);
  sleep(2);
  relay.switchPin(4, OFF);
}
```

Store it Quickly 2.0

By **Jürgen Werner**
(Germany)

Some microcontroller applications require status information or other important data to be stored to EEPROM immediately as power to the equipment is turned off or fails. When power is resumed this information will then be available for use as required. To solve this problem Rainer Reusch developed a circuit (**Figure 1**) and it appeared in Elektor magazine as a Design Tip [1]. The principle behind this original circuit is that the voltage at the anode of D1 falls sooner than voltage across reservoir capacitor C2. A comparator evaluates these levels and outputs a signal to the microcontroller indicating that the input voltage has fallen. Thanks to D1 and C2 voltage at the non-inverting input to IC1.A falls faster than the voltage at the inverting input. This produces a Low level at the comparator output, triggering an interrupt. As long as there is sufficient energy stored in the reservoir capacitor, the microcontroller now has time to store all important data to EEPROM before the supply rail sinks too low.

The circuit functions effectively, at least in simple situations. One problem is that it takes a few milliseconds to write data to EEPROM cells. The value of C2 must therefore be larger than is strictly necessary since it must also act as a reservoir to supply the regulator when input voltage falls. Apart from that the calculation of C1, for the ripple voltage is not so easy. Even more of a problem is if the power is supplied from a wall wart type adapter which includes built-in voltage regulation or switch-mode supply. In this case the circuit cannot work because the voltage at the input to R1 does not fall fast enough thanks to the reservoir capacitors integrated into the adapter. These shortcomings led the author to set about tweaking the original design; the result can be seen in **Figure 2** which is both a better and simpler solution. The comparator has now been moved to after the voltage regulator. With this configuration we are comparing the input voltage with the voltage output from the regulator. We no longer need the diode in series with the voltage regulator. The reservoir capacitor C1 does not need to be so big now. The biggest improvement however is that now the circuit is not dependant on how quickly the input voltage falls. When the voltage from the power adapter sinks the level on the output of the regulator is held constant by regulator action. When dimensioned correctly the voltage divider at the non inverting input of the comparator produces an input voltage lower than the level at the inverting input, generating a low output to trigger an interrupt in the microcomputer.

The circuit values have been calculated assuming the mains adapter has a 9 V output and the voltage regulator produces 5 V. D1 protects the regulator from current flowing in the reverse direction. With C1 equal to 100 µF and a load current of 5 mA the microcontroller has at least 17 ms in which to store data to EEPROM. An edge triggered interrupt is used here. When it is possible to disable power-hungry features of the microcontroller such as any A/D converters, that'll give extra time to store data.

(120674)

[1] Store it Quickly! Elektor January 2009, www.elektor.com/080379

Another Look at Some Specific Points of the
500 ppm LCR Meter
The luxury of precision within everyone's reach

Jean-Jacques Aubry,
Ollioules

The success of the Elektor 500 ppm LCR Meter

The enthusiastic, attentive reactions from a great many readers to the publication of this project show that high-quality measuring instruments are still among their favorite constructional projects. The author, Jean-Jacques Aubry, has issued a correction in the French forum [4] — the only significant error so far noted — for the USB connector and SW1 cut-outs drawing. For the SW1 cut-out, the 1.16 mm dimension should be 0.96 mm. For the J19 cut-out, the 8.03 mm dimension should be 5.39 mm and the 3.97 mm dimension should be 6.61 mm.

A corrected drawing has been included in the download on the website, as has a new version of the schematic which corrects a few points of detail. In the components list for the main circuit, J17, listed along with J7, J8, … should in fact be J9. R81 appears twice; the correct value is 7.5 kΩ, R81 = 10 kΩ should be deleted. On the main circuit diagram, R31 = 750 Ω and on the extension circuit, the resistor in series with LED D5 is R8 = 1 kΩ. And lastly, C30 = 1.5nF 5% NP0.

These corrections do not affect the operation of the circuit. The PCB and the ready-to-use module are unchanged, and as everything is working well, to date (May 23, 2013) there have been no updates since the 3rd article came out. And while we're talking details, at the start of the "First time Setup" document downloadable from our site, the firmware file is wrongly called LCR3A_update_Vxxx.hex. The correct name is LCR3A_firmware_Vxxx.hex.

The author will be happy to answer any other questions directly on Elektor's English language forum too, and will gladly discuss points with users of his precision LCR meter.

In the three articles describing the LCR meter [1], certain aspects were glossed over, to avoid making the articles excessively long. We propose here to go back over some details that may also be interesting outside the context of the device described.

This is the case for a special routine for displaying icons on a graphics screen, which takes a bit of gymnastics.

We're also going to discuss the accuracy of the measurements, the factors affecting accuracy, and errors. Reading this article will only be of any use if you have already also read the three preceding articles – particularly the description of the circuit and measuring principles.

Graphics display library

Where necessary, the bootloader and firmware use the graphics display (GLCD) to display messages and images. The display's RAM is organized into 128 columns and 8 lines of bytes (64 bits or pixels). Now the height of the elements to be displayed (icons) exceeds 8 bits and they are described using several bytes: the height of the two fonts used is e.g. 11 and 16 bits. All the icons are defined in the file glcd_bitmaps.c (or bootloader_glcd_bitmaps.c for the minimalist version, minus some of the icons and fonts used by the

firmware). If we want to be able to write in any position, e.g. spanning byte boundaries, and if writing into the GLCD's RAM is done in whole bytes at well-defined addresses, we need to: know the contents of the RAM before writing, and perform some clever calculations between the old and new bytes so as to only modify the required pixels (bits).

The number of port lines available for the display means we are obliged to use the serial mode. Unfortunately, this mode does not allow us to read the display's RAM. So we have to create a mirror of the GLCD's RAM in the MCU RAM:

```
uchar xdata GLCD_Array[LCD_COLS]
[LCD_ROWS];
```

These calculations between old and new bytes quickly seem inextricable, as there are so many different cases to resolve. The solution I've adopted consists in representing a column as 64 bits instead of as 8 bytes!

This solution requires:

1. Creating an image in the MCU RAM of the columns where each byte represents a single pixel in the GLCD RAM column, hence a value of 0 or 1.

```
uchar xdata Column_Array[(LCD_ROWS + 1) * 8]; // + 1 for 2nd byte of char in last
line
```

2. Creating a routine to read the display column (in fact the mirror) and write, after converting to equivalent bits (0 or 1), into Column_Array [].

```
void GLCD_read_column(uchar col)
{
    uchar  rows, pix, i, j;
    i = 0;
    // read column col, byte after byte
    for (rows = 0; rows < LCD_ROWS; rows++)
    {
            pix = GLCD_Array[col][rows];
            // write pix, bit after bit
            for (j = 0; j < 8; j++)
            {
                    Column_Array[i] = pix & 0x01;
                    pix >>= 1;
                    i++;
            }
    }
}
```

3. Creating a routine to read the contents of Column_Array[], write to the GLCD RAM, and update the mirror:

```
void GLCD_write_column(uchar col)
{
    uchar  rows, pix, i, j;
    i = 0;
    // write column col, byte after byte
    for (rows = 0; rows < LCD_ROWS; rows++)
    {
            pix = 0;
            // read pix, bit after bit
            for (j = 0; j < 8; j++)
            {
                    pix += Column_Array[i] << j;
                    i++;
            }
            if (GLCD_Array[col][rows] != pix)    // only if GLCD RAM byte modified
            {
```

```
                              GLCD_set_pos(rows, col);
                              GLCD_WriteData(pix);           // draw in GLCD RAM
                              GLCD_Array[col][rows] = pix;  // update RAM mirror
                    }
              }
       }
```

4. Creating a routine to display a defined icon in bitmap. This can be done anywhere, whatever its size (within the limits of the display!)

```
void GLCD_show_icon(uchar code *bitmap, uchar width, uchar height, uchar x, uchar
y, uchar mode)
{
     uchar  tx, ty, pix, hb, i, j, k;
     hb = (height - 1) / 8 + 1; // character height in bytes
     for (tx = 0; tx < width; tx++) // loop for width columns
     {
              GLCD_read_column(tx + x);
              i = y;
              k = 0;
              for (ty = 0; ty < hb; ty++) // read hb bytes of icon
              {
                       pix = *(bitmap + ty * width + tx); //  read one byte
                       if (mode == GLCD_PIXEL_OFF)
                              pix = ~pix;
                       for (j = 0; j < 8; j++) // write 8 bits of pix to Column_Array
                       {
                              if (mode != GLCD_PIXEL_INV)
                                     Column_Array[i] = pix & 0x01;
                              else
                                     Column_Array[i] ^= pix & 0x01;
                              pix >>= 1;
                              if (k == height)
                                     break;
                              i++;
                              k++;
                       }
              }
              GLCD_write_column(tx + x);
     }
}
```

5. And lastly, creating a write routine for text, which just means writing a succession of icons that are defined in a font.

```
void GLCD_draw_text( uchar x, uchar y, uchar *text , uchar mode )
{
     uchar i, posx, posy;
     uchar *pt;
     posy = y - font_height + 1;
     for( pt = text, i = 0; *pt; i++, pt++ )
     {
              posx = x + i * font_width;
              if( posx + font_width > LCD_COLS )
              {
                       i = 0;
                       posx = x;
                       posy += font_height;
              }
              GLCD_show_icon( font + (*pt - font_offset) * font_charsize, font_
width, font_height, posx, posy, mode );
     }

}
```

Measurement accuracy

The impedance to be measured can be written:

$$Z_x = \frac{V_p + jV_q}{I_p + jI_q} \times \frac{G_i\ R_{sense}}{G_v} = \left[\frac{V_p\ I_p + V_q\ I_q}{I_{p^2} + I_{q^2}} + j\ \frac{V_q\ I_p - V_p\ I_q}{I_{p^2} + I_{q^2}} \right] \times \frac{G_i\ R_{sense}}{G_v}$$

where G_i and G_v are the current and voltage gains of the amplifier chain, and R_{sense} is the *IU_converter* resistor.

$$G_i = G_{INA128} \times G_{BUFFER} \times G_{PGA}i \times G_{DAC}i \qquad\qquad G_v = G_{INA128} \times G_{BUFFER} \times G_{PGA}v \times G_{DAC}v$$

Hence again, using the series representation of an impedance $Z = R_S + j\ X_S$

$$R_S = \frac{V_p\ I_p + V_q\ I_q}{I_{p^2} + I_{q^2}} \times \frac{G_{PGA}i\ G_{DAC}i\ R_{sense}}{G_{PGA}v\ G_{DAC}v} \qquad\qquad X_S = \frac{V_q\ I_p + V_p\ I_q}{I_{p^2} + I_{q^2}} \times \frac{G_{PGA}i\ G_{DAC}i\ R_{sense}}{G_{PGA}v\ G_{DAC}v}$$

If we ignore the digitizing errors, as explained in the "We need to keep an eye on the gain" paragraph in the first article [1], then:

$$\frac{\Delta R_S}{R_S} = \frac{\Delta X_S}{X_S} = \left[\frac{\Delta G_{PGA}i}{G_{PGA}i} + \frac{\Delta G_{PGA}v}{G_{PGA}v} \right]_{i \neq v} + \left[\frac{\Delta G_{DAC}i}{G_{DAC}i} + \frac{\Delta G_{DAC}v}{G_{DAC}v} \right]_{i \neq v} + \frac{\Delta R_{sense}}{R_{sense}}$$

We can split the overall error into two main parts:
- the error due to the inaccuracy in the PGA103 and DAC8811 gains
- the error in the true value of the resistors in the *IU_converter* (calibration error).

To this will be added an error due to:
- the display (± 1 bit in the last digit)
- the residual phase error (after phase error compensation)
- the fluctuation caused by amplification noise and stray signals picked up by the measuring leads (power line, etc.)

And then the digitizing error will appear in ranges 1 and 8, when the amplitudes of the voltage or current signals become too different through lack of gain.

Factors affecting accuracy

Gain error

According to the BURR-BROWN (TI) documentation for the PGA103

Gain	1	10	100
Typical gain error	±0.005%	±0.02%	±0.05%
Max gain error	± 0.02%	±0.05%	±0.2%

For ranges 3–6, the PGA103 programmable amplifier always has unity gain and the term

$$\left[\frac{\Delta G_{PGA}i}{G_{PGA}i} + \frac{\Delta G_{PGA}v}{G_{PGA}v} \right] \quad \text{is zero.}$$

For ranges 2 and 7, this term introduces a maximum error of ±0.07 % (typically ±0.025 %).
For ranges 1 and 8, this term introduces a maximum error of ±0.07 % (typically ±0.025 %).

According to the *TI* documentation for the **DAC8811C**, its max. relative precision is ± 1 LSB Hence the maximum gain error will be ± 1/N, where N is the code defining the final amplifier gain.

post_Ampli. step	0	1	2	3	4	5	6	7
N	**7 500**	**8 700**	**10 000**	**11 600**	**13 500**	**15 500**	**18 000**	**20 700**
post_Ampli. step	8	9	A	B	C	D	E	F
N	**24 000**	**27 600**	**32 000**	**36 900**	**42 600**	**49 100**	**56 700**	**65 500**

When the *post_amplification_U* and *post_amplification_I* steps are equal, the term

$$\left[\frac{\Delta G_{DAC}i}{G_{DAC}i} + \frac{\Delta G_{DAC}v}{G_{DAC}v} \right]$$ is zero.

Otherwise, it is maximum when one is 0 and the other 1; it then has a value of $\frac{1}{7500} + \frac{1}{8700}$ i.e. 0.025 %.

Note: Using the B version (**DAC8811B**) doubles this error.

The *IU_converter*, whose open-loop gain is not infinite, also introduces a measurement error. As the closed-loop gain is ≤1 in ranges 3–6, its open-loop gain of around 80 dB (10,000) @ 10 kHz introduces an additional error of around 0.01 %. This will be negligible at the lower frequencies.

Note: On the computer, the *AU2011* program lets you display the gain error value.

Phase error
The use of a very wide bandwidth (50 MHz) amplifier in the *IU_converter* and the various phase compensations (described in the downloadable document "First time Setup" [3]) allow us to reduce spurious phase shifts to a minimum. However, they are not completely eliminated. What's more, the phase of the final amplifier is assumed to be independent of the gain, which is only true to a first approximation, as with data between 0x2000 and 0xFFFF, its bandwidth is fairly constant (around 8 MHz) within the operating range of the DAC8811.
This residual error will have an effect on the value of the secondary parameter, which will be all the more pronounced the closer the component under test's phase is to 0 ° or ±90 °.

Calibration error
The initial error is that of the precision resistors fitted on the PCB, i.e. ±0.05 %. It is possible to improve this by following the indications given in paragraph 7 *Calibrating the resistors* in the downloadable document "First Time Setup" [3].

Internet Links & References

[1]   500 ppm LCR meter, Part 1 www.elektor. com/110758

[2]   500 ppm LCR meter, Part 2 www.elektor. com/130022

[3]   500 ppm LCR meter, Part 3 www.elektor. com/130093

[4]   www.elektor.fr/forumLCR

(130174)

HOT CHIPS **25**

CHIPS

ADVANCE PROGRAM
August 25-27, 2013
A Symposium on High-Performance Chips
Memorial Auditorium, Stanford University, CA
http://www.hotchips.org

HOT CHIPS brings together designers and architects of high-performance chips, software, and systems. The tutorial and presentation sessions focus on up-to-the-minute developments in leading-edge industrial designs and research projects. The expanded poster session features industry as well as student posters.

Sun. Aug. 25

Morning Tutorial
Heterogeneous System Architecture (HSA): Overview and Implementation
AMD, ARM, Qualcomm

Afternoon Tutorial
Fast Storage for Big Data   **Bay Storage, Fusion IO**

Monday August 26

SoC1
- AMD's Kabini SoC   **AMD**
- Intel's Bay Trail SoC for Tablets, Netbooks, and Entry Desktops   **Intel**
- Clover Trail+ – Intel's Next Atom SoC Smartphone Platform   **Intel**

Keynote 1   The Chip Design Game at the End of Moore's Law
Dr. Robert Colwell, Director of Microsystems Technology Office   DARPA

Processors
- Next Generation POWER Microprocessor   **IBM**
- IBM zEC12 Mainframe Symmetric Multiprocessor System   **IBM**

Keynote 2   Google Glass
Babak Parviz   Google

SoC2
- AMD's Richland Client APU SoC   **AMD**
- A 50% Lower Power ARM Cortex CPU using DDC Transistors with Body Bias **Suvolta**

Interconnects
- Silicon Photonics Platform for Integration of Optical IOs with ASICs   **Luxtera**
- NetSpeed eNoC – a Quantum Leap in On-Chip Interconnect Design   **Netspeed**
- Integrating Rack Level Connectivity into a PCI Express Switch   **Pixtech**

Tuesday August 27

FPGA Based Dataflow
- Dataflow Architectures for 10Gbps Line-Rate Key-Value-Stores   **Xilinx**
- Trading and Risk in a Chip with Multi-Scale Dataflow Computing   **Maxeler**
- An FPGA-Based In-line Accelerator for Memcached   **U. Texas**

Networking
- Serial Networking Memory Bandwidth Engine Breaks 4.5 B Accesses/sec **Mosys**
- HARRIER– A 22nm 200Gb/s High-Performance Network Flow Processor **Netronome**

Keynote 3   Intellectual Property Issues in the Computer and Electronics Industries
Michael Brody, Vice-Chair, Intellectual Property   Winston and Strawn

Mobility
- Qualcomm Hexagon DSP: An Architecture for Mobile Multimedia   **Qualcomm**
- Power Management Challenges in Wireless WAN SoCs   **Intel**
- 5th Generation Touchscreen Controller for Mobile Phones and Tablets   **Cypress**

Processors 2
- Hardware-level Thread Migration in a 110-core Shared-Memory Processor   **MIT**
- Haswell Client Product Family   **Intel**
- Microprocessors for Roots of Trust   **Bluerisc**

Processors 3
- SPARC64 X+ : Fujitsu's Next-Generation Processor for UNIX Servers   **Fujitsu**
- SPARC M6: Oracle's Next-Generation Processor for Enterprise Systems   **Oracle**
- Bixby: the Scalability and Coherency ASIC in Oracle's Enterprise Systems   **Oracle**
- XBOX One Silicon   **Microsoft**

Organizing Committee
Chair
Jan Willem Van De Waerdt   Cypress
Vice Chair
Krste Asanovic   UC Berkeley
Member at Large
Ralph Wittig   Xilinx
Finance
Lily Jow   HP
Advertising
Don Draper   Oracle
Sponsorship
Amr Zaky   Broadcom
Publications
Randall Neff
Press
Ralph Wittig   Xilinx
Registration
Charlie Neuhauser   Neuhauser Associates
Larry Lewis
Location Services
John Sell   Microsoft
Allen Baum
Volunteer Coordinator
Gary Brown   Tensilica
Webmasters, IT
Kevin Broch
Alice Erickson   Alice Erickson Consulting
Production
Lance Hammond
Mike Albaugh
Keith Diefendorff

Steering Committee
Chair
Alan Jay Smith
Committee Members
Allen Baum
Don Draper   Oracle
Pradeep Dubey   Intel
Lily Jow   HP
John Mashey   Techviser
John Sell   Microsoft
Keith Diefendorff

Program Committee
Program Co-Chairs
Don Newell   Qualcomm
Mike Flynn   Maxeler Tech.
Committee Members
Forest Baskett   NEA
Pradeep Dubey   Intel
Fred Weber
Alan Jay Smith   UC Berkeley
Samuel Naffziger   AMD
Subhasish Mitra   Stanford
Israel Koren   U. Massachusetts
Thomas McWilliams BayStorage
Rumi Zahir   Intel
Ralph Wittig   Xilinx
Oliver Pell   Maxeler Tech.
Patrick Hung   CPO Tech.
Behnam Robatmili   Qualcomm
Bill Dally   NVIDIA
Founder Bob Stewart   SRE

ORACLE®   AMD   **Warthman Associates** Technical Writers www.warthman.com   elektor   IEEE computer society

CIRCUIT CELLAR®   MOUSER ELECTRONICS   SSCS

A Symposium of the Technical Committee on Microprocessors and Microcomputers of the IEEE Computer Society and the Solid-State Circuits Society

SPOT THE DIFFERENCE

By **Wisse Hettinga**

In line with a strong trend in the industry, Renesas underscore that their latest GR Sakura board is Arduino compatible. This overview allows you to check for yourself just how far that compatibility goes.

Difference #1
Okay, the GR Sakura board is pink! But then, look at the specs—Pink is Power!

Difference #2
It's 8 bit, 16 MHz and limited memory of the AVR controller against full 32 bit, 96 MHz and massive memory capacity of the Renesas processor.
The real question is, which applications will actually unleash GR Sakura's full potential.

Difference #3
The USB Host functionality on the Sakura board shows potential. It is implemented with a Mini-B connector, while a Type-A connector can be fitted on the back side of the board.
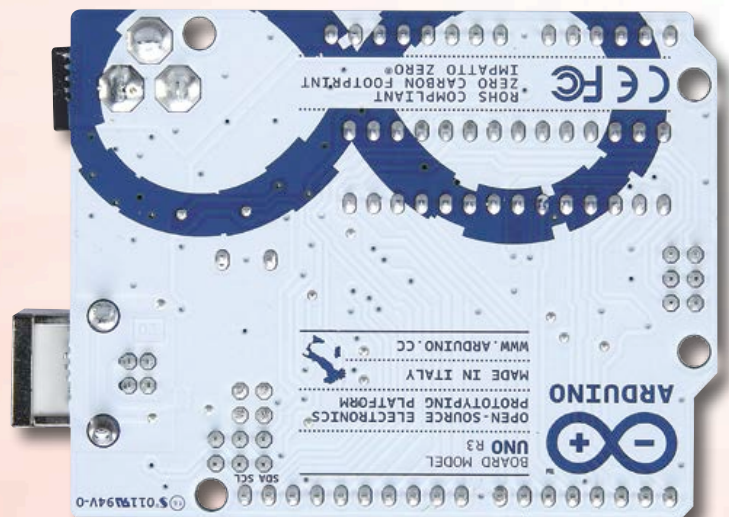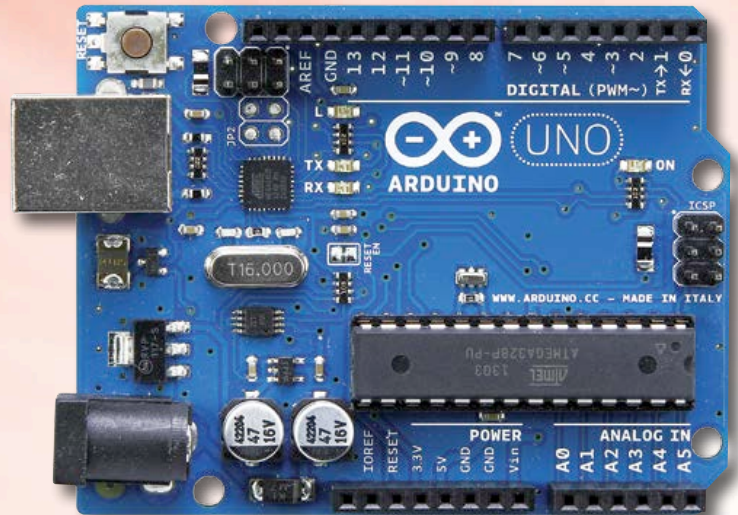
	Arduino Uno	GR Sakura
Supply Voltage(s)	5 V processor operating voltage	3.3 V processor operating voltage 5 V board supply voltage
USB	Type B connector Board runs off USB voltage by default.	Mini B connector Type A connector provides Host support Board runs off USB voltage by default
Network	None	Ethernet RJ45

ARDUINO UNO *VS.* GR SAKURA FULL
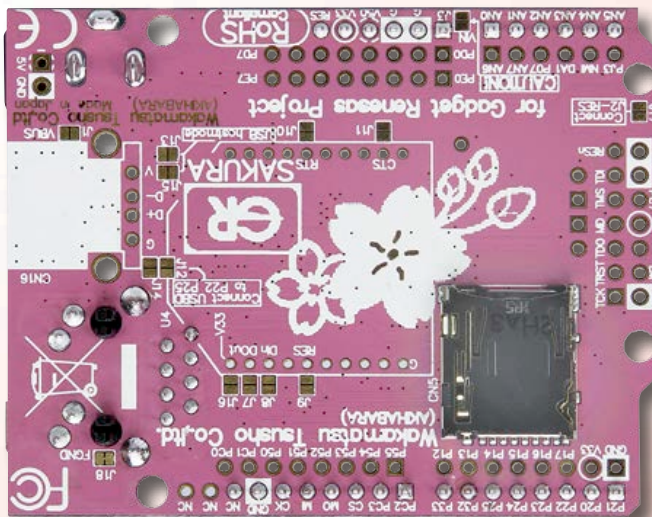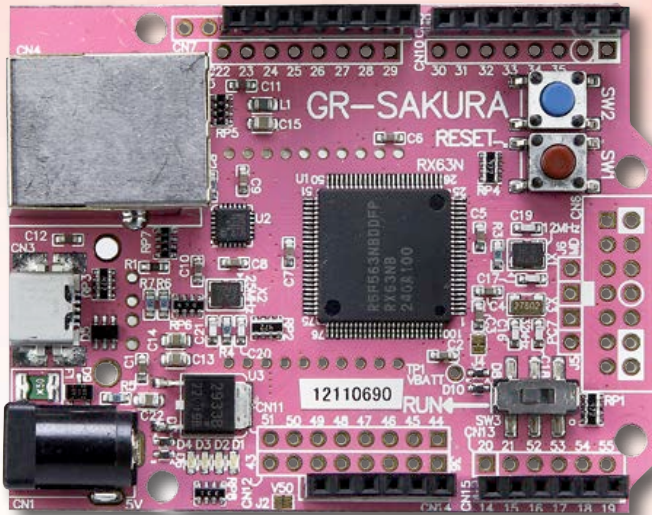
Difference #4

You can't deny that the true power of the Arduino concept is due to the huge and easily accessible program library (at www.arduino.cc), rather than the hardware. However, don't underestimate the programming options available for the Sakura board either. It has lots of features that come to life once you start using the Cloud Base Compiler. Just hook up the board to your PC, push the right buttons—all well documented— and watch the board appear as a new drive connected to your PC. With the new drive a link is provided that takes you straight to the website. If you have an Android phone, have a look at Gadget Director—an easy, 'icon' based programming language.

Find all the references at www.designspark.com and go to the Design Centers.

(130177)

	Arduino Uno	GR Sakura
Processor	ATmega328 8 bit 16 MHz operating frequency	RX63N 32 bit 96 MHz operating frequency
Memory	32 KB Flash of which 0.5 KB used by bootloader SRAM: 2 KB EEPROM: 1 KB	1 MB Flash RAM: 128KB Data Flash: 32 KB MicroSD socket

Wideband Wien Oscillator
with Single-Gang Pot

By **Merlin Blencowe**
(United Kingdom)

This Wien bridge oscillator (after Max Wien, 1866–1938) produces a low-distortion sine wave of constant amplitude, from about 15 Hz to 150 kHz. It requires just four opamps and will work off a single 9-volt battery. Also, unlike most Wien bridge oscillators, it does not require a dual-gang potentiometer for tuning.

Op amp IC2b provides an artificial ground so that the circuit will operate from a unipolar supply (9 V battery or power pack). IC2a is the main amplifier for the oscillator. The frequency range is divided into four decades by 2-pole, 4-way rotary switch SW1.

Only one arm of the Wien network is varied, but the change in positive feedback that would normally result is compensated for by IC1b, which works to bootstrap R2, thereby changing the negative feedback enough to maintain oscillation. A *linear* change in the resistance of the tuning pot results in a roughly *logarithmic* change in frequency. To get a more conventional linear change a log-taper pot is used wired so that rotating the knob anticlockwise causes frequency to increase. You could use an anti-log pot the other way around if you prefer, but these things are notoriously hard to find.
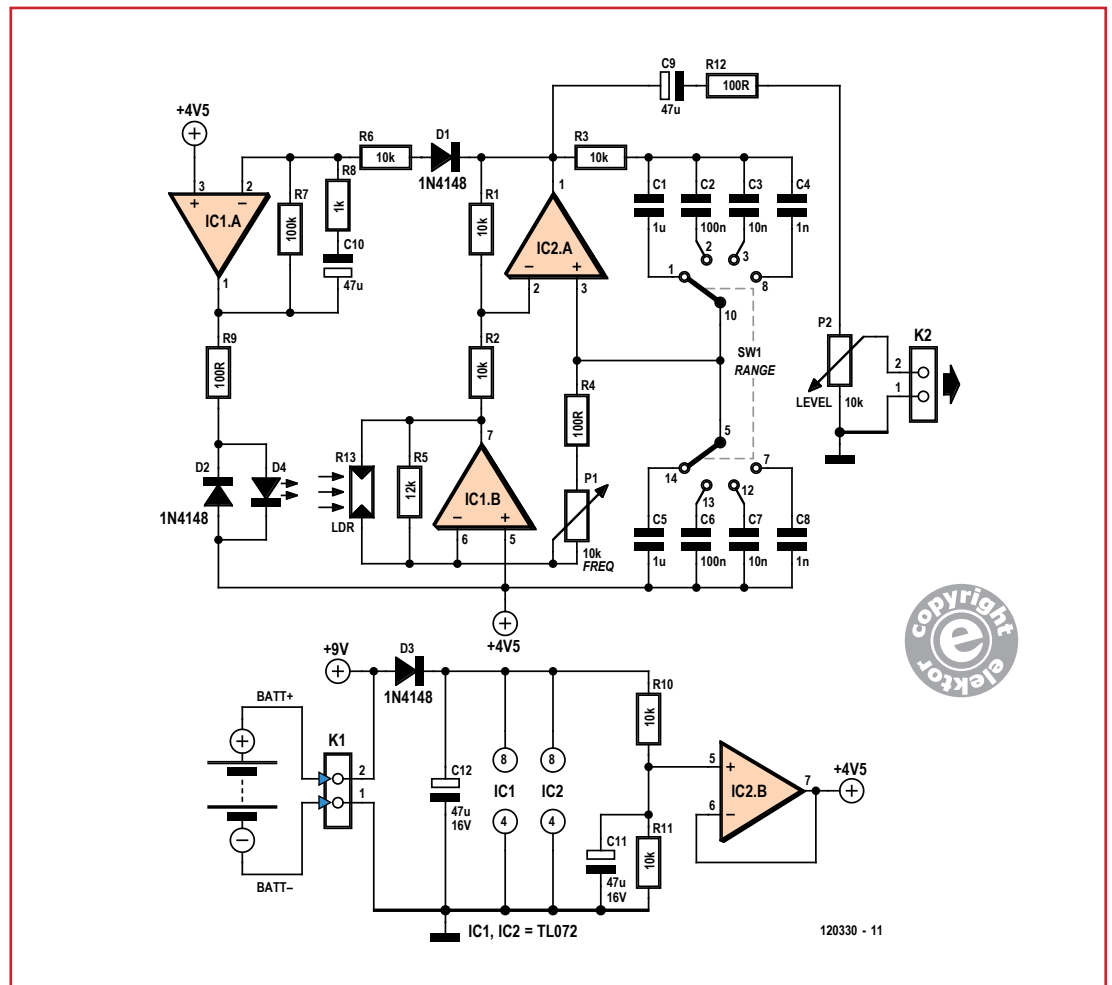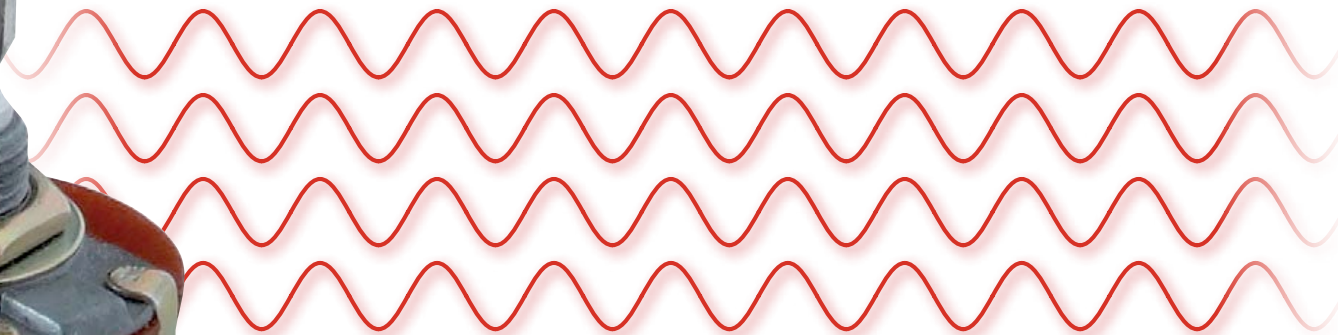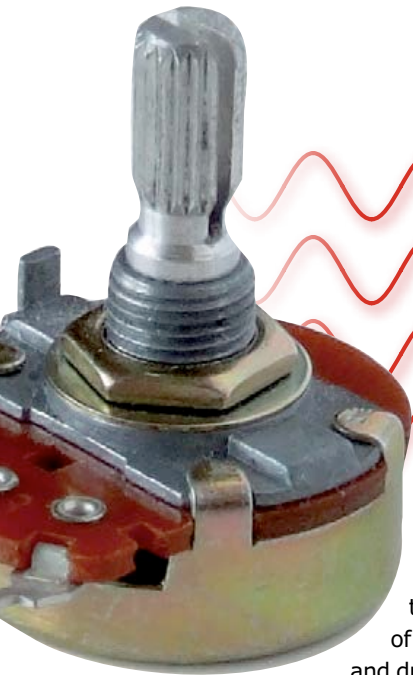
IC1A is an integrator that monitors the amplitude of the output signal and drives an LED (D2). This must be mounted facing the LDR (light dependent resistor) and shielded from ambient light (for example, with a piece of heat-shrink tubing). IC1a is then able to control the gain of IC2a so that oscillation is maintained with minimum distortion.

The maximum output amplitude of the generator is about 2 V_{p-p} when the LED and LDR are mounted as close as possible. Distortion is less than 0.5 % in the lowest range, and too low for the author to measure in the higher ranges. Any LDR should work, provided its dark resistance is greater than 100 kΩ. If you do not have an LDR with such high resistance, try increasing R5 until oscillation starts. Breadboarded prototypes of the circuit were built by the author using dual and quad opamp packages, and both work equally well.

The DesignSpark schematic and circuit board design files for this project are available for downloading from www.elektor.com/120330.

(120330)

COMPONENT LIST

Resistors
R1,R2,R3,R6,R10,R11 = 10kΩ
R7 = 100kΩ
R4,R9,R12 = 100Ω
R5 = 12kΩ
R8 = 1kΩ
P1,P2 = 10kΩ potentiometer, logarithmic law
R13 = LDR, R(dark) >100kΩ, e.g. Excelitas Tech type
   VT90N1 (Newark/Farnell # 2568243)

Capacitors
C1,C5 = 1µF solid
C2,C6 = 100nF
C3,C7 = 10nF
C4,C8 = 1nF
C9-C12 = 47µF 16V, electrolytic, radial

Semiconductors
D1,D2,D3 = 1N4148
D4 = LED, red, 5mm
IC1,IC2 = TL072ACP

Miscellaneous
SW1 = 2-pole 4-position rotary switch, C&K Components type RTAP42S04WFLSS
K1,K2 = PCB terminal block, 5mm pitch
PCB # 120330-1
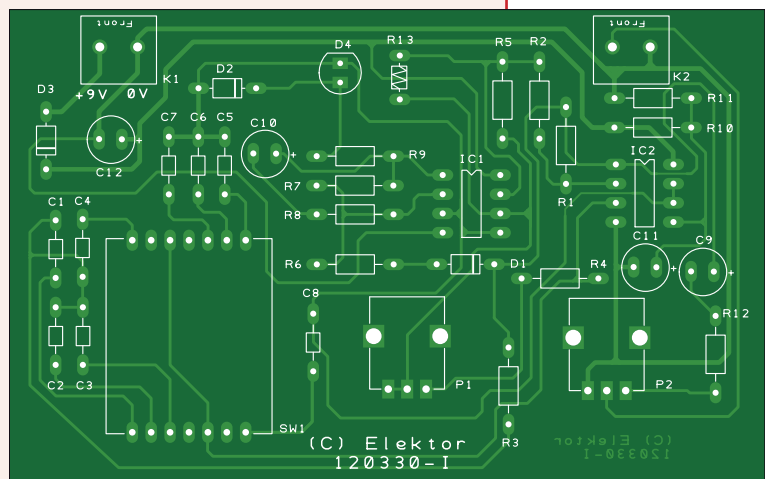
4 Amps Photovoltaic Charge Controller

The use of solar photovoltaic (PV) energy sources is increasing due to global warming concerns on the one hand, and cost effectiveness on the other. Many engineers involved in power electronics find solar power tempting and then addictive due to the 'green' energy concept. The circuit discussed here handles up to 4 amps of current from a solar panel, which equates to about 75 watts of power. A charging algorithm called 'pulse time modulation' is introduced in this design.
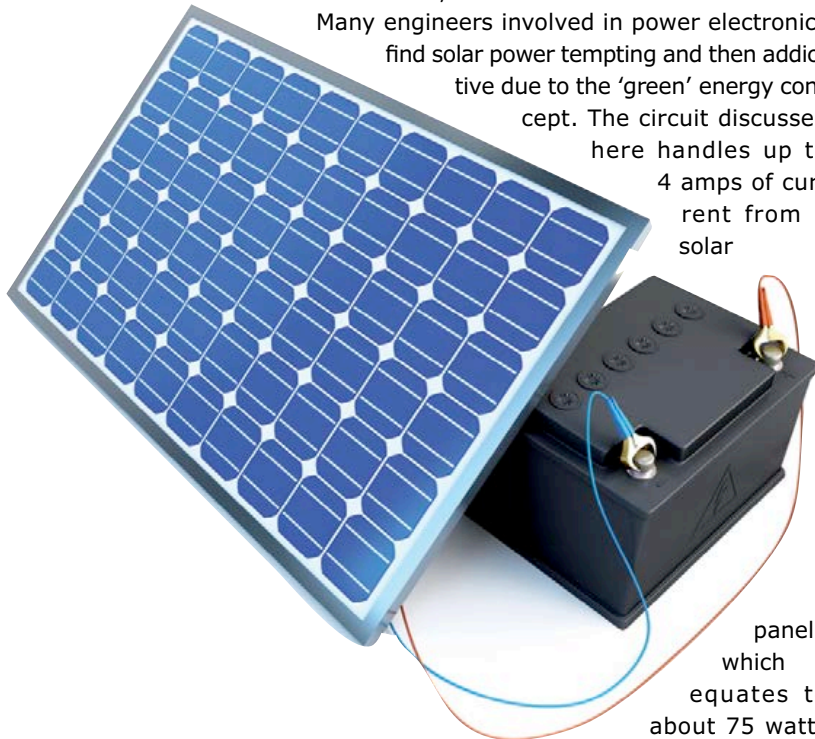
By **T. A. Babu** (India)

The current flow from the solar panel to the battery is controlled by an N-channel MOSFET, T1. This MOSFET does not require any heat sink to get rid of its heat, as its $R_{D-S(on)}$ rating is just 0.024 Ω. Schottky diode D1 prevents the battery discharging into the solar panel at night, and also provides reverse polarity protection to the battery. In the schematic, the lines with a sort-of-red highlight indicate potentially higher current paths.

The charge controller never draws current from the battery—it is fully powered by the solar panel. At night, the charge controller effectively goes to sleep. In daytime use, as soon as the solar panel produces enough current and voltage, it starts charging the battery.

The battery terminal potential is divided by resistor R1 and trimpot P1. The resulting voltage sets the charge state for the controller. The heart of the charge controller is IC1, a type TL431ACZ voltage reference device with an open-collector error amplifier. Here the battery sense voltage is constantly compared to the TL431's internal reference voltage. As long as the level set on P1 is below the internal reference voltage, IC1 causes the MOSFET to conduct. As the battery begins to take up the charge, its terminal voltage will increase. When the battery reaches the charge-state set point, the output of IC1 drops low to less than 2 volts and effectively turns off the MOSFET, stopping all current flow into the battery. With T1 off, LED D2 also goes dark. There is no hysteresis path provided in the regulator IC. Consequently, as soon as the current to the battery stops, the output of IC1 remains low, preventing the MOSFET to conduct further even if the battery voltage drops. Lead-acid battery chemistry demands float charging, so a very simple oscillator is implemented here to take care of this. Our oscillator exploits the negative resistance in transistors—first discovered by Leo Esaki and part of his studies into electron tunneling in solids, awarded with the Nobel Prize for Physics in 1973.

In this implementation, a commonplace NPN transistor type 2SC1815 is used. When the LED goes out, R4 charges a 22-µF capacitor (C1) until the voltage is high enough to cause the emitter-base junction of T2 to avalanche. At that point, the transistor turns on quickly and discharges the capacitor through R5. The voltage drop across R5 is sufficient to actuate T3, which in turn alters the reference voltage setting. Now the MOSFET again tries to charge the battery. As soon as the battery voltage reaches the charged level once more, the process repeats. A 2SC1815 transistor proved to work reliably in this circuit. Other transistors may be more temperamental—we suggest studying Esaki's laureate work to find out why, but be cautioned that there are *Heavy Mathematics Ahead*.

110751 - 11
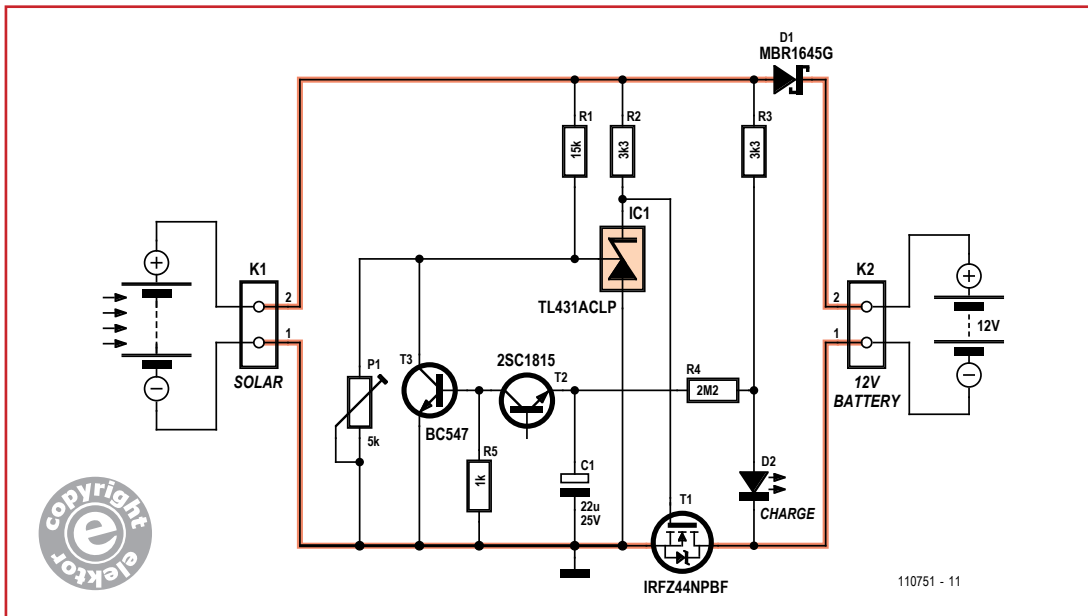
As the battery becomes fully charged, the oscillator's 'on' time shortens while the 'off' time remains long as determined by the timing components, R4 and C1. In effect, a pulse of current gets sent to the battery that will shorten over time. This charging algorithm may be dubbed Pulse Time Modulation.

To adjust the circuit you'll need a good digital voltmeter and a variable power supply. Adjust the supply to 14.9 V, that's the 14.3 volts battery setting plus approximately 0.6 volts across the Schottky diode. Turn the trimpot until at a certain point the LED goes dark, this is the switch point, and the LED will start to flicker. You may have to try this adjustment more than once, as the closer you get the comparator to switch at

exactly 14.3 V, the more accurate the charger will be. Disconnect the power supply from the charge controller and you are ready for the solar panel. The 14.3 V setting mentioned here should apply to most sealed and flooded-cell lead-acid batteries, but please check and verify the value with the manufacturer. Select the solar panel in such a way that its amps capability is within the safe charging limit of the battery you intend to use. The DesignSpark schematic and circuit board design files for this project are available for downloading from www.elektor.com/110751.

(110751)

COMPONENT LIST

Resistors
R1 = 15kΩ
R2,R3 = 3.3kΩ 1%
R4 = 2.2MΩ
R5 = 1kΩ
P1 = 5kΩ preset

Capacitors
C1 = 22µF 25V, radial

Semiconductors
D1 = MBR1645G (ON Semiconductor)
D2 = LED, 5mm
IC1 = TL431ACLP (Texas instruments)

T1 = IRFZ44NPBF
  (International Rectifier)
T2 = 2SC1815 (Toshiba)
  (device is marked: C1815)
T3 = BC547

Miscellaneous
K1,K2 = 2-way PCB terminal block,
  lead pitch 5mm
PCB # 110751-1
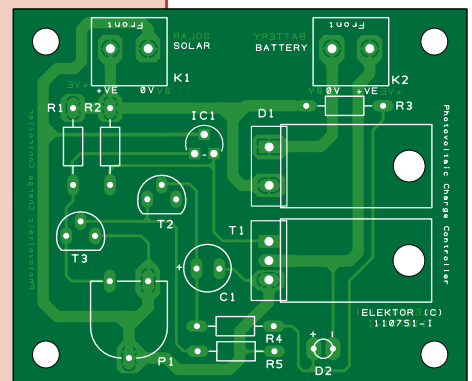
Starting a Schematic Design

By **Neil Gruending**
(Canada)

Last time I talked about how DesignSpark uses technology files to store configurations settings. In this article we'll start a new project and start a schematic design. We'll start by configuring the DesignSpark libraries and then we'll set up a schematic title block so we can create a nice looking schematic.

Figure 1.
Inspecting DesignSpark's library paths.

Configuring the libraries
DesignSpark uses library files to organize all of your design information. Schematic symbols are one library type and PCB footprints are another. They are then combined to make a component library that you can use to place parts and documentation symbols into your design. The only difference between a documentation symbol and a regular component is that a documentation symbol contains either a schematic symbol or a PCB symbol, but not both. If you would like more information about DesignSpark's library system and how it works, there is a good tutorial at [1]. In this article we will make a schematic documentation symbol to use a title block in a schematic but before we do that we need to double check

DesignSpark's library search paths. You do that by going into the 'Files -> Libraries...' menu and selecting the 'Folders' tab. Here you will see a screen that looks something like **Figure 1**.

You want to make sure that the directory where you are storing your library files is listed as the first item in the 'Folders and Search Order' box which in my case is 'C:\Users\Neil Gruending\ Documents\dspcb2lib\library'. You can reorder the directory list by selecting the directory that you want to move and using the Up/Down buttons. I would recommend that you don't save any changes or new files into the DesignSpark system default library folders just in case the libraries get overwritten in a future upgrade.

Now that the library paths are set up, you can create a new schematic symbol library to store our schematic title block by selecting the 'New Lib…' button on the 'Schematic Symbols' tab. Then select the 'New Item…' button to open up a blank schematic symbol page. For more information you can review the symbol creation tutorial on the DesignSpark website [1].

Creating a schematic title block
I personally always use title blocks on a schematic to make them look much more professional and to help to document a design. DesignSpark is different from other packages because schematic title blocks are stored in a schematic component library instead of a template file or technology file which means that DesignSpark will ignore any drawing elements in a schematic technology file. DesignSpark comes with several title block templates in its Schema library in several different sizes like A4 and Letter, but I prefer to use Tabloid (11 in. × 17 in.) for my designs. I also prefer to use a more traditional documentation area that takes up less of the drawing area. In my last article I showed you how to use truetype fonts in a schematic technology file, but there is a downside to using them in a schematic title block. That's because DesignSpark shifts truetype fonts slightly downwards when printing

a schematic to a PDF file. That normally doesn't matter for things like reference designators but in title blocks where text alignment is more important you'll definitely notice. Therefore I chose to use stroke fonts for my title block, which is shown in **Figure 2**.

I recommend that you name the various text styles so that they're easy to modify later. In my case I ended up with the styles shown in **Figure 3**.

The numbers and letters around the drawing area frame use the text style 'Frame' and the field descriptions use the text style 'Title-small'. Field items use the style 'Title'. Since DesignSpark doesn't support project variables you have to add the text strings to the title block manually, which is why all of the title block fields are blank in the schematic component. Also, you don't have to add these text styles to the schematic technology file because they will be copied into the schematic when you add the title block.
Once you've finished editing the title block, save it to the schematic symbol library you created earlier so that we can create a schematic document symbol. The first step is to open the Library Manager ('File->Libraries…') and go to the Components tab. You can create a new component library by selecting the 'New Lib…' button and then create the document symbol by clicking on the 'New Item…' button which will open the 'New Component' window. Here you can give the component a name and select your title block symbol. Unchecking the 'PCB Symbol' check box will make the component a schematic document symbol like we need. Save your changes and let's start a new DesignSpark project.

Creating a new project
DesignSpark uses projects to collect all of the relevant information about a design like schematics and PCB documents in one place. The main reason for using a project is to allow a set of schematic sheets to be linked to a PCB design. The linked schematic sheets can then act as one large project where global net information is shared and all component designators are unique.

Creating a new DesignSpark project is simple. Go into the 'File -> New' menu to open the 'New Design' window, select 'Project' and then press 'OK'. You will then be prompted where to save

the new project and then a blank project will be created for you. Now you can add existing files to the project by using the 'Project -> Add Files to Project…' menu. Adding new items to a project is done by opening the 'New Design' window, but before clicking on OK make sure you check the 'Add to Open Project' box.

Conclusion
Now that we can create projects and create nice looking schematic templates, the next step is to start drawing your design using components from the DesignSpark libraries. You can also create and use your own libraries with some extra component attributes that will make it easier to generate bill of material (BOM) listings later. The title block I've drawn here (**Figure 4)** is available from my dsppcb2lib project on Bitbucket at [3].

(130181)

Figure 2.
Structure and layout of the schematic title block.
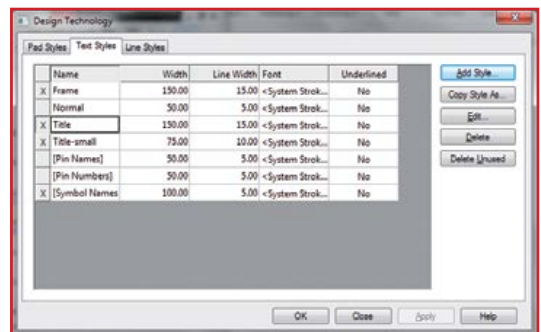
Figure 3. Text formatting in the title block.
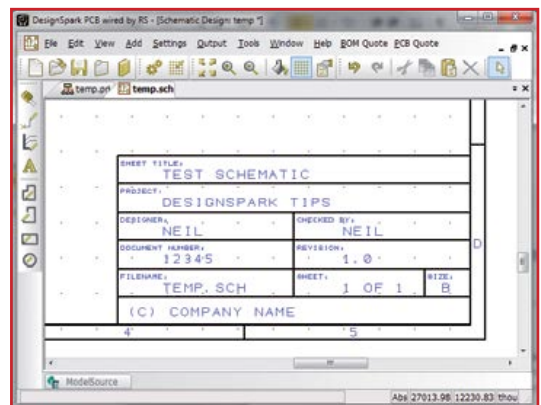
Figure 4. A completed schematic title block.

Internet References
[1] www.designspark.com/tutorial/components-library-structure-library-manager

[2] www.designspark.com/tutorial/components-creation-with-symbol-footprint-wizards

[3] https://bitbucket.org/neilg/dspcb2lib

Celebrate!

By **Clemens Valens**
(Elektor.Labs)

This month we party because Elektor**.**LABS has passed the 5,000 registered participants mark. Thank you! Also we are happy to start awarding Gold Memberships for outstanding **.**LABS performers. Another excellent reason to *Join Us...* *

EDITOR'S CHOICE

D Return of D Formant

Most people posting a project on **.**LABS limit themselves to a few lines of text and maybe a photograph or a schematic. But not Greek contributor "AChorevas", hell no! Not being an Elektor member this OP contacted us to post his **d-Formant** project on Elektor**.**LABS, a **d**igital version of the legendary Elektor Formant modular analog music synthesizer from the late seventies. After reading the project summary he kindly provided, we were more than happy to grant him free access to the **.**labs website. Not even in our wildest dreams did we expect that this would result in fifteen posts containing detailed descriptions, sound samples, schematics and source code files.
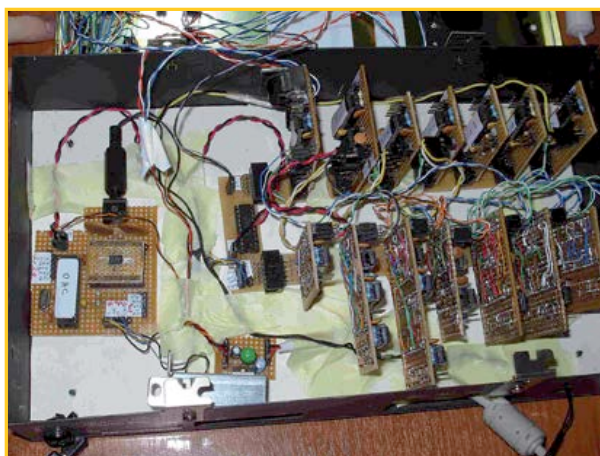
The OP's d-Formant is an all digital sound synthesizer. In the instrument all the analog signals of the original Formant got replaced by 16-bit digital streams, while PIC24 microcontrollers have taken the place of the transistors and operational amplifiers. The design is fully modular and offers the patching capabilities of the original Formant, allowing the user to rewire sounds on the fly. All the original controls can be implemented, although the prototype has a much simpler and cheaper user interface: an LCD, some pushbuttons and a rotary encoder (that alone is enough for me to like it ;-). To play the synthesizer you can use a standard MIDI keyboard or any other MIDI device capable of sending 'note on' and 'note off' commands.

We are definitely going to write about this excellent project in the printed magazine. Meanwhile we have awarded AChorevas a free Elektor Gold Membership. Congratulations! www.elektor-labs.com/node/3124

(130098-I)

* Quote from *The Evil Dead* (1981).

---

www.elektor-labs.com

Note: OP stands for Original Poster, the person who started an online project or discussion. OPs who want to have a chance of appearing in the printed edition of Elektor must (regularly) check the email address they use to access Elektor.Labs. This is our only means of contact.

SMD Desoldering Tip

If an engineering problem appears daunting and complex that does not always equate to the solution being expensive and/or high tech. Elektor Labs resident Luc Lemmens supplied a valuable tip on a method of desoldering surface mount ICs he found on the Internet. All you need are precision pliers, a short piece of solid copper wire—like electrical installation wire—a decent solder iron and some solder tin, and a pair of tweezers. When desoldering ICs with leads at two opposite sides of the case, it is suggested you start by folding the copper wire as shown in the first photo. Make sure the wire contacting the pins is as straight as possible, making sure it touches as many IC pins as possible. Now apply some solder to the wire where heat needs to be transferred (second photo). This improves the heat transfer to the IC pins and the PCB pads significantly.
Press the wire on the chip pins as shown in the third photo and heat it with a decent soldering iron, all the while making sure the pins at both sides of the IC are in full contact with the wire. When the solder has melted, quickly remove the IC from the PCB using tweezers.

With a little custom folding of the copper wire, even ICs with pins at all four sides of the package can be desoldered using the above method. Shape the copper wire in a similar way, as pictured in the fourth image.

Watch out for damage to the IC as well as the PCB by overheating, and keep the time you apply heat limited to just melt the solder and be able to securely remove the IC. An IC that's been subjected to overheating, meaning it's been soldered on either too long or at an excessive temperature will obviously be DBR (damaged beyond repair). As is the precious circuit board, where too much heat will result in copper pads becoming detached from the PCB surface. Once you get the hang of it and perform this little trick correctly wielding your solder, solder iron and copper wire, you will damage neither IC nor PCB.

(130099)

Found on http://youtu.be/dCUSwADP6DE.

By **Thijs Beckers,**
Associate Editor
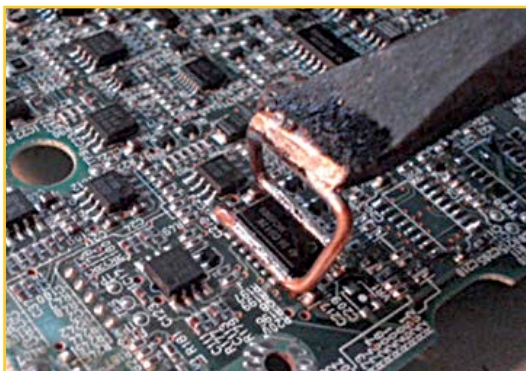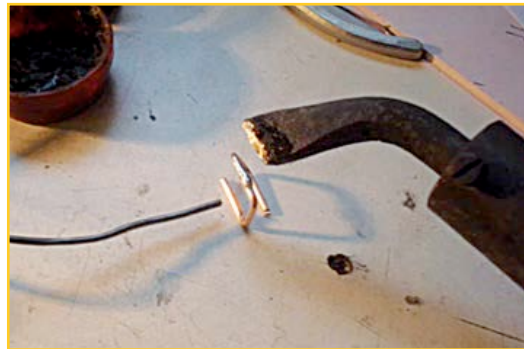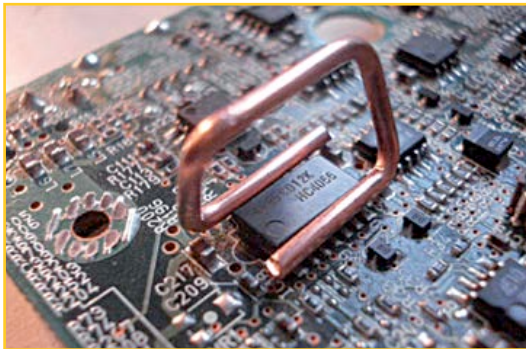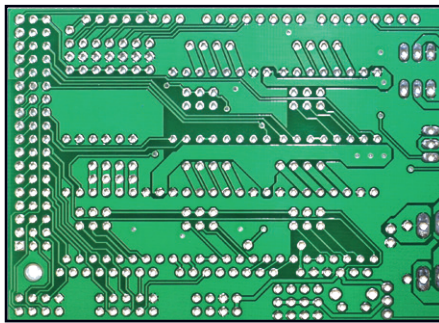
Professional PCB Manufacturing

How your four-layer PCB gets produced at Elektor PCB Service

The technology of manufacturing printed circuit boards professionally has evolved immensely since Elektor published its first "panels" some 50 years ago. In this article we venture into the kitchen of our Elektor PCB Service, powered by our production partner Eurocircuits, to check out how we bake our —multi-layer— cakes.
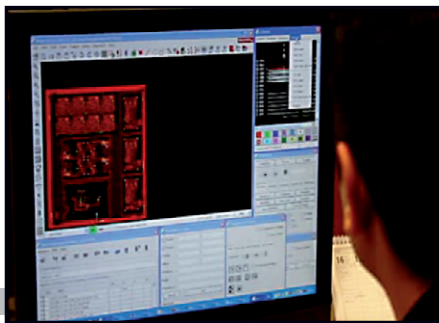
Understanding the manufacturing process is sure to enable you to design PCBs that can be produced easier and cheaper while also improving long-term reliability, so that your customers will keep coming back to you. So let's have a look at the manufacturing process of a 4-layer PCB.

Professional PCB manufacturers usually don't manufacture PCBs as one-offs. Instead they combine several circuits onto one large production panel, which is much more efficient to produce and handle throughout the production process. This is often called 'order pooling'. Eurocircuits also produce their PCBs this way. In the illustrations you should be able to discern four individual designs combined in a single panel.

(1) From Gerber to production data
The board designer has prepared his/her track layout on a CAD system (Computer Aided Design). In response to each system using a proprietary data format, the PCB industry developed a standard output (file) structure to ensure a uniform format is used to describe the physical properties of a PCB. This is format is called "Extended Gerber" or "RS274X". The Gerber files define the layers of copper tracks as well as the solder masks and component positions and designations.
The first job in the whole process is to check whether the data submitted by a client meets the manufacturing requirements. This is mostly done automatically. Track widths, space between tracks, pads around holes, smallest hole size and the likes are pitched against manufacturability. Once the design is approved, an engineer will output all tool files needed to run the machines that produce and test the PCB.

(2) Photo tools for PCB image transfer
A laser photo plotter prints the films needed for the production. These are automatically developed and held ready for the PCB fabrication process. For every PCB layer one film or photo tool is generated. The films are aligned with each other by punching precisely located alignment holes in each sheet. These holes will align with the alignment pins on the imaging equipment so the different layers of the board will be perfectly aligned.

(3) Inner layer imaging
To produce the inner layers of a multilayer PCB, manufacturers like Eurocircuits typically start with a panel of laminate, which is an epoxy resin and glass-fiber core with
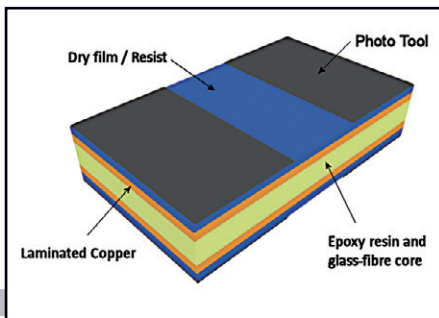
copper foil pre-bonded onto each side. First the copper is cleaned and the panel is transferred to a clean room to ensure no dust gets onto the surface where it could cause a short or open circuit. The cleaned panel is given a coat of photosensitive film, the photoresist.

Next the image on the film is transferred to the PCB by a 'printer' using powerful UV lamps to cure the photoresist through the clear film, thus defining the copper pattern. The bed of the printer has alignment pins matching the holes in the photo tools and panel. The operator positions the first film using the pins, then the coated panel, and then the second film. The pins ensure that the top and bottom layers are precisely aligned.

When 'printed', the panel is sprayed with a powerful alkali solution to remove any resist that failed to cure completely. The panel is then pressure-washed and dried. The copper pattern is now covered by the cured resist. The operator checks the panel to make sure that the copper surface is clean and all the unwanted resist has been removed.

(4) Etching the inner layers
The copper layout is now etched using a powerful alkaline solution to dissolve the exposed copper. The process is carefully monitored to ensure that the final conductor widths are exactly as designed.
Next, the blue photoresist protecting the copper image is stripped off. The operator checks that all the photo-resist has been removed.

(5) Alignment and inspection of inner layers
The inner core of our multilayer board is now complete. The operator punches alignment holes in the panel to align the inner and outer layers. Since there is no way to correct any mistakes on the inner layers once the outer layers are attached, a full inspection is given to the panel. An automatic optical inspection system scans the board and compares it with the digital image generated from the original design data.

(6) Bonding the inner and outer layers
The outer layers consist of sheets of glass cloth pre-impregnated with uncured epoxy resin ('prepreg') and a thin copper foil. First a copper foil and two sheets of prepreg are placed on the heavy steel base plate. Then the pre-treated core is carefully positioned using the alignment pins. Lastly two more sheets of prepreg, another copper foil and an aluminum press plate are put on top. This stack is loaded into the bonding press, which uses heated press plates and pressure to bond the layers of the PCB together. The heat melts and cures the epoxy resin in the prepreg, while the pressure bonds the layers together. The bonding process is computer controlled, so a permanent bond that will last the lifetime of the PCB is ensured.

Copper foil
Prepreg
Core
Prepreg
Copper foil

(7) Drilling
X-ray drilling of reference holes.
Before etching the outer copper layers, all holes for leaded components and vias are drilled. First an X-ray equipped drill is used to locate the drill positions in the copper of the inner layers. The machine drills alignment holes to ensure precise drilling through the center of the inner layer pads.

Prepare the stacks for drilling
To set up the drill the operator first puts a panel of disposable material on the drill bed to keep the drill from tearing the copper foil as it exits at the bottom side. Then he loads the panel and a sheet of aluminum entry foil.

Drilled hole
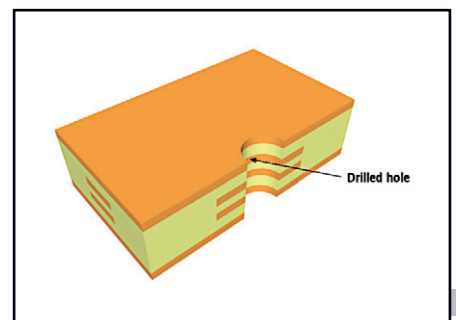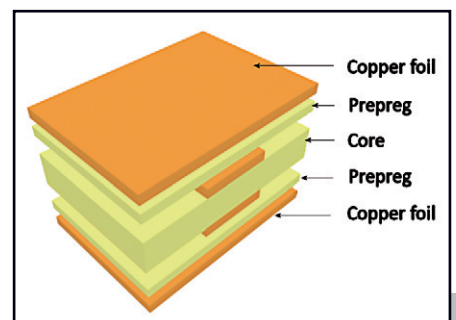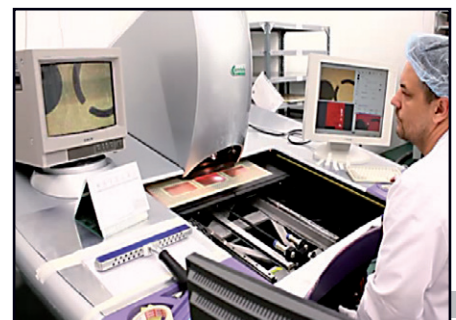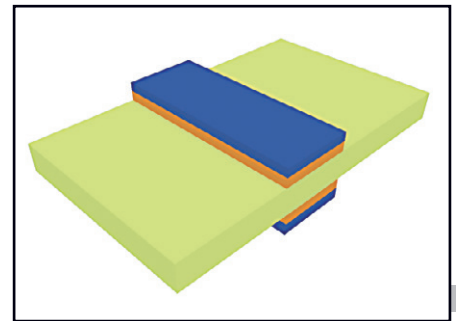
Drilling the holes

The drilling machine is computer-controlled. The operator selects the correct drilling program. This tells the machine which drill to use at what XY coordinates. The drill uses air-driven spindles which can rotate up to 150,000 rpm. High speed drilling ensures clean hole walls to provide a secure base for plating.

Drilling is a slow process as each hole must be drilled individually. So depending on the drill size up to three PCB panels are drilled in one run. The machine selects the drill to use from the drill rack, checks that it is the correct size and loads it into the drill head.
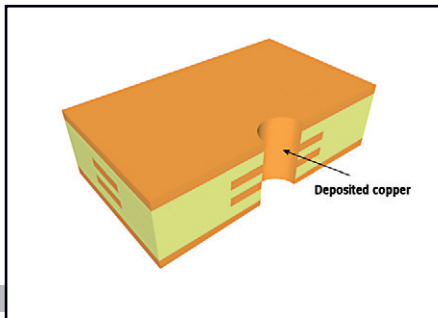
Cut-off excess resin

During the bonding process excess resin from the prepreg is squeezed to the edge of the panel outside the image area. This is cut off on a computer controlled profiling machine. The drilled panel is now ready for plating.

(8) Plating – first part

First a conductive layer is deposited over the hole walls. The operator clamps the panel into a jig so the panels can be taken through a series of chemical and rinsing baths, where the hole walls are seeded with micro particles of palladium, and a layer of copper about 1 micron thick can be deposited. The remaining copper is up for "electroplating", a process to deposit a layer of metal onto an electrically conductive surface. But first...

(9) Outer layer imaging

... the panel is transferred to the clean room again and coated with a layer of photoresist, which is hot-rolled onto the copper using a cut-sheet laminator. The operator loads the first film onto the alignment pins, followed by the laminated panel, and finally the second film. After removing the Mylar film protecting the photoresist, the uncured resist is removed in a developer. The operator checks the panels again to make sure that the copper surface is clean and all the unwanted resist has been removed.
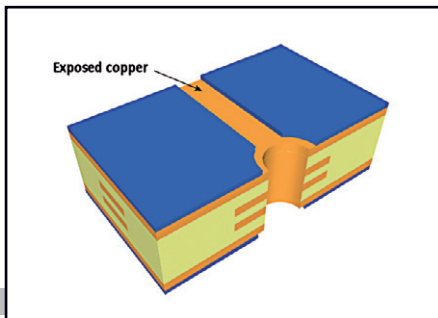
(10) Plating – second part

Now the boards are electroplated with copper. The operator starts the automated plating line, where the copper surface is cleaned and activated in a number of baths and then electroplated. The whole process is computer controlled to ensure that each panel stays in each bath exactly the right amount of time.

To ensure good conductivity through the holes about 25 microns of copper are needed on the hole walls. Due to the way electroplating works, 25–30 microns gets effectively plated onto the rest of the surface tracks as well. Thus starting off with a 17.5-micron copper foil will result in a 40–42 micron copper layer after processing.
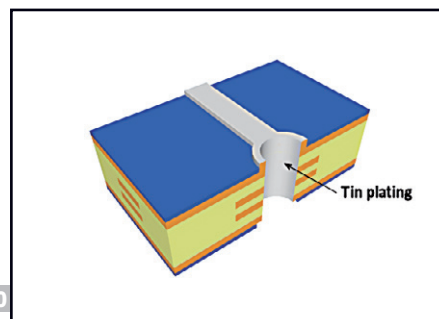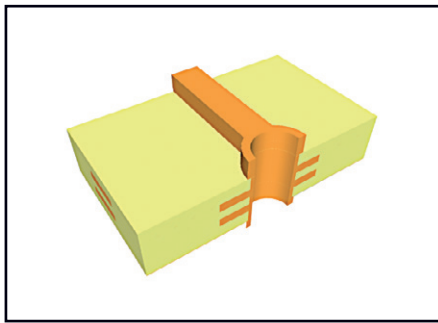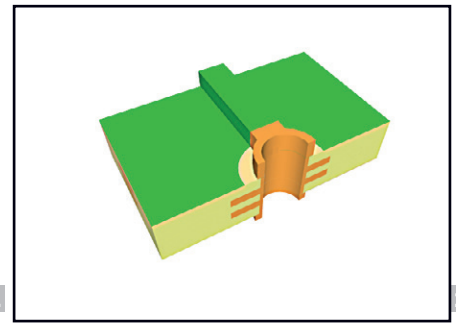
For the next step—etching off unwanted copper foil—a thin layer of tin is plated onto the copper. When the plating is completed the operator uses a non-destructive test to check the copper and tin plating for correct thickness.

(11) Etching the outer layers

Now the outer layers are etched. First the resist covering the unwanted copper is dissolved and washed off. Then a powerful alkaline solution etches away any exposed copper. The process is carefully controlled to ensure sideways etching is prevented, so the finished track widths are exactly as designed. Finally the thin tin coating which used to protect the copper image, is stripped off.

(12) Solder mask coating

Before applying the solder mask, the panels are first cleaned and brushed to remove any surface tarnish. Then they are loaded into the vertical coating machine which simultaneously covers both sides of the panel with the epoxy solder mask ink. The panel is racked and put through a conveyorized drier which cures the resist just enough to allow it to be printed ("tack-dried"). The operator checks for a complete and even coating.

Next, the coated panels are imaged using a UV printer. The operator mounts the photo tool films on the machine and places the panel onto the alignment pins. As with the etching and plating resists used earlier in the process, the UV lamps in the machine cure the ink where the film is clear. This is where solder mask will be on the finished board.

The imaged panels are put on a conveyor moving them out of the clean room and into the developer, which strips off the uncured and unwanted resist. The operator checks the alignment of the solder mask on the panel and makes sure there are no traces of ink on the pads or through the holes. To provide a robust and permanent coat the resist gets cured once more in a conveyorized oven.

(13) Pad and hole finishing

The copper pads and holes for the component wires or terminals do not have solder mask material on them. A solderable surface finish is now applied to protect the copper until the components are soldered onto the board. The image shows a gold finish, which is achieved by chemically depositing 5 microns of nickel onto the copper followed by 0.1 microns of gold over the nickel.

Under the EU Reduction of Hazardous Substances (RoHS) legislation, lead cannot be used in the finishes, so gold over nickel is offered, as well as a sterling silver finish or lead-free hot-air leveling. For the last option the panel is lowered into a bath of molten tin. As it is lifted from the bath, hot air jets blast the surplus molten metal from the panel to leave a coating of tin of about 2 microns thick.
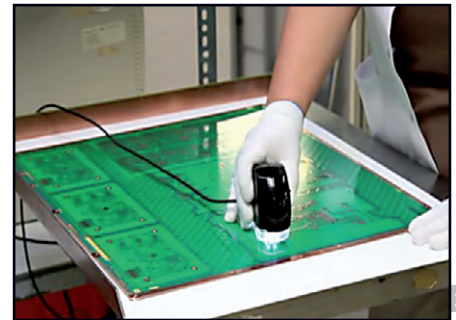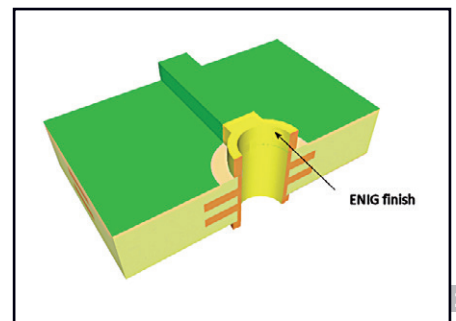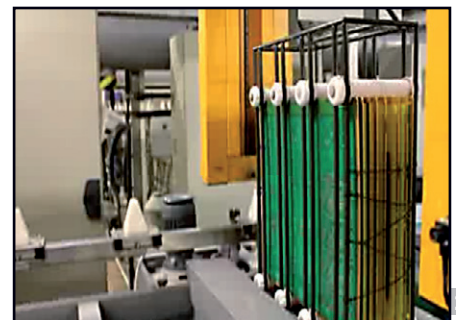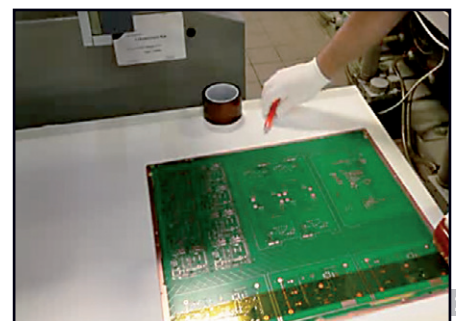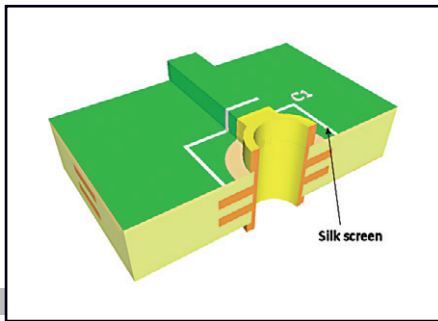
(14) Hard gold electroplating

Electroplated gold is needed for edge connectors getting repeatedly inserted in, and removed from, a bus connector. First, the operator puts protective tape on the board above the connectors. Then he mounts the panel on a horizontal electroplating bath. Between 1 and 1.5 microns of gold are then electroplated over 4–5 microns of plated nickel.

(15) Silk screen printing

A special inkjet printer is used to print the silk screen directly onto the board. This printer works just like a conventional inkjet printer where minute droplets of ink are sprayed onto the panel to generate the image. Now both the epoxy ink solder mask and the silk screen are finally cured. This takes about 10 minutes using a five-stage conveyorized oven.
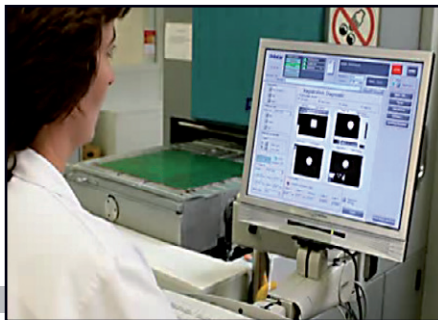
(16) Routing out the panels

The panel is now ready for separating the different PCBs by routing them out. A computer controlled milling machine mills out any small slots or internal cutouts first. Then the milling head automatically picks up a 2 mm cutter, checks the diameter and mills around each PCB.

(17) Electrical testing

At the end of the PCB production process every multilayer PCB is electrically tested against the original board data. A flying probe tester checks each net to ensure that it is complete (no open circuits) and does not short to any other net.
A faster method using an Acceler8 machine is optional. This uses 4000 tiny probes like a brush. It builds an electronic map of the PCB from a pre-tested board. Then it compares each board with its map. This cuts test times by 90%.
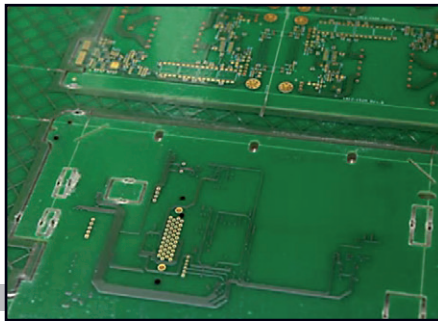
(18) Final inspection

In the last step of the process a team of eagle-eyed inspectors carefully check each PCB. If everything is OK, a release note is printed. The PCBs are vacuum-sealed to keep out dirt and moisture. Next they get bubble-wrapped, securely boxed, sealed and shipped out to customers.
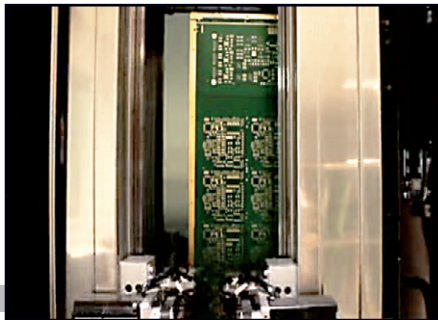
Now you know how we produce your PCB at elektor PCB Service and what happens after you order it at www.elektorpcbservice.com. In the next installment we will focus on design requirements you have to comply with considering several physical properties of the production process described here.

(130061)
Illustrations courtesy of Eurocircuits.
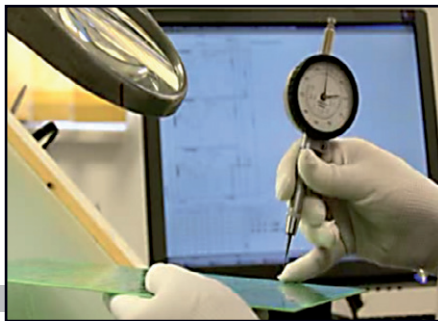
Internet Link

www.elektorpcbservice.com

Eurocircuits are a European based manufacturer of standard technology printed circuit boards. Its headquarters are located in the picturesque Belgian town of Mechelen, while production units are near Aachen in Germany, and in Eger, Hungary. Eurocircuits specialize in providing prototypes and small batch PCBs for designers, product development departments, niche market electronics companies, universities and research institutions.

ByPic Dev Board is Text Oriented

The ByPic development board from ByVac is a unique concept in microcontroller development that allows you to "program with text". The 'intelligence' is within the IC. This means that no IDE is necessary, no complex C code, no compiling , and no programmer is required, just a serial interface. Build, connect and it's ready to go; simply download your text and the IC will take care of the rest.

The concept is ideal for beginners and rapid prototyping as there is no need to learn any complex IDE systems, a free terminal emulator is available. Perfect for 'what if' scenarios and to test out new ideas.

The IC is a PIC32MX150F128 that has 128 K of flash, 32 K RAM and operates at 40 MHz. Loaded into the flash is the unique ByPic development system that gives you access to all of the internals of the microcontroller without having to know how it all works. As a great confidence booster, once connected to a serial port it gives an 'ok' prompt to let you know everything is working okay. The language will interpret commands typed directly into the IC but will also compile any functions that you write and so you have the interactivity of an interpreted language and the speed of a compiled language. As an example, typing:

adc_init(0) will set up ADC channel 0 and

print adc_get(0) will get the value of the channel

The IC can be purchased on its own to built into your own projects, the BP1 is an Arduino™ shaped board with a large prototyping area that is supplied in kit form. The kit costs just £9.95 and £12.95 with a USB to serial adapter.
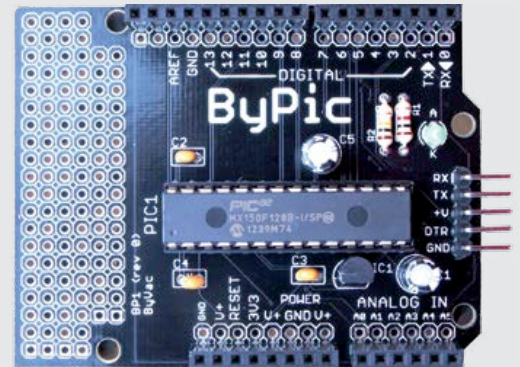
www.byvac.com   (130223-III)

USB 3.0/WiFi Mixed Signal Oscilloscope with Protocol Analyzer

USBee.com, the website for CWAV, has introduced the first PC-based mixed signal oscilloscope (MSO) integrated with a protocol analyzer utilizing USB 3.0 and WiFi technology. The USBee QX is a 600 MHz MSO with 24 digital channels and 4 analog channels, resulting in the highest integrated MSO in the PC-based test instrument category. While competitive MSOs provide protocol decoders that display data in complex HEX format, the USBee QX utilizes a protocol analyzer to display serial or parallel protocols in human readable format using a packet presentation layout. By eliminating the tedious tasks of constantly converting HEX data to meaningful interpretations, firmware developers and verification engineers are more productive in their debug process, saving man-days or man-weeks of effort.

With a large buffer memory of 896 Msamples coupled with data compression capability, the USBeeQX can capture up to 32 days of traces, allowing developers to find and resolve the most obscure and difficult defect. The USBee QX includes popular serial protocols such as RS232/UARTs, SPI, I2C, CAN, SDIO, Async, 1-Wire, and I2S that are typically costly add-ons for benchtop oscilloscopes. In addition, the USBee QX has the unique capability to support any custom protocol, utilizing APIs and Tool Builders that are integrated into the USBee QX software.

With protocol packets to wire behavior on a single time correlated screen; external triggering with multiple test instruments are no longer needed, enabling the capture of symptoms and root cause in a single trace. The new WiFi capability in the USBee QX allows the test set-up to be in the lab while the developer or engineer is at their desk. WiFi also creates electrical isolation of the device under test to the host computer.

With a price of $2495, the USBee QX is 85% lower price than equivalent benchtop MSOs. Being extremely small, portable, and affordable, every firmware and electronic hardware engineer or firmware developer can have a debug system at their disposal, eliminating wasted time for scheduling lab time or accessing shared test instruments.

www.usbee.com   (130223-IV)

FabStream:
Three Steps to DIY and Small Volume PCBs

DownStream Technologies, LLC recently launched FabStream™, an integrated printed circuit board (PCB) design and manufacturing solution targeted at the DIY electronics market (small businesses, start-ups, engineers, inventors, hobbyists, and other electronic enthusiasts).

"FabStream offers the DIY user free, powerful and easy-to-use PCB design tools integrated with a network of PCB manufacturers," said Rick Almeida, Founder of DownStream and Managing Director of FabStream. "FabStream supports those individuals who are less experienced in PCB design and manufacturing by eliminating design pitfalls that can lead to problems getting a PCB produced."

FabStream works in three easy steps. First, the user logs onto Fabstream.com and selects a manufacturing partner and downloads the free design software (SoloPCB™). The software has been customized with the specific manufacturers' production capabilities allowing the user to work within the manufacturers' constraints right from the start. Next, they create PCB libraries, schematics and board layouts. Finally, the software leads the user through the process of ordering PCBs on-line with the manufacturer. The user only pays for the PCBs they purchase.

Almeida explains, "The success of FabStream is
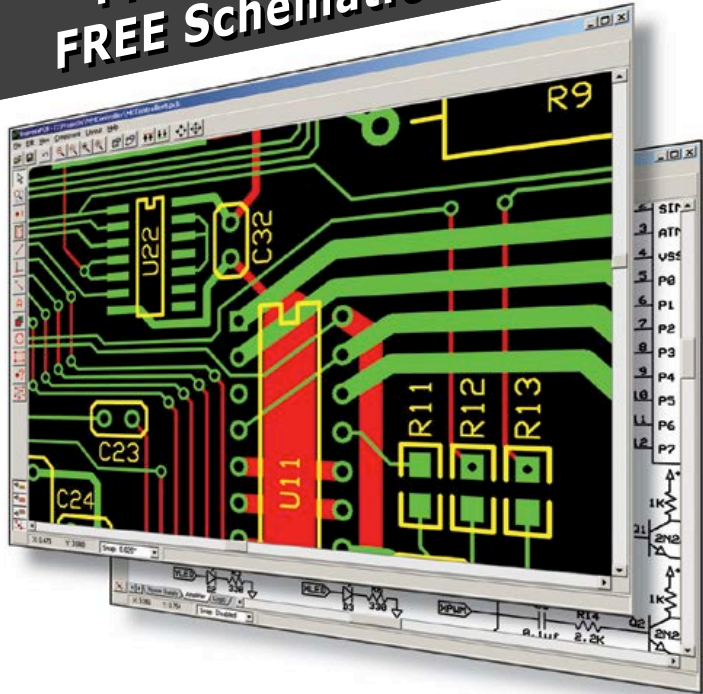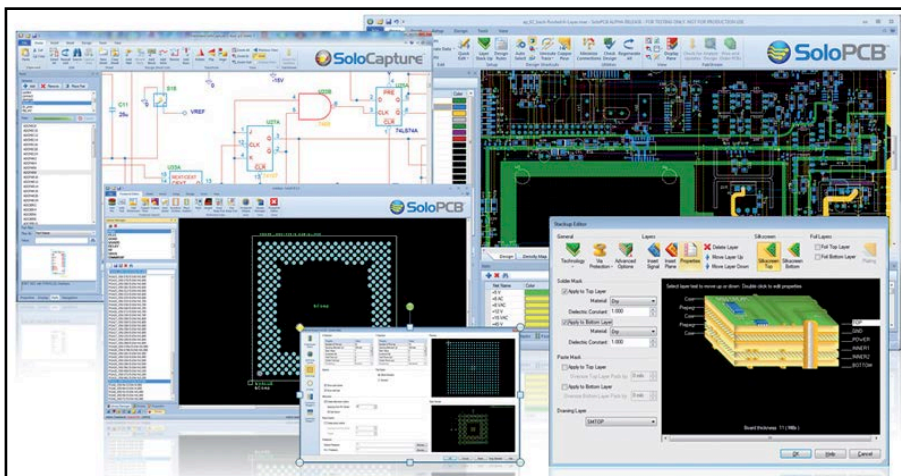
tied directly to our customers' success. Unless a user can get a PCB designed and built, we do not receive any income. This gives us strong incentive to make sure our users not only have all the technology they need, but that it's presented in a way that doesn't impede their workflow".

www.fabstream.com
(130223-II)

Myriad Open Source RF Project Gets US Backing

Richardson RFPD has committed to the Myriad open source RF initiative. The US-based distributor will begin stocking and selling the Myriad-RF-1 board to customers around the world via its website's online store immediately. Myriad was launched in March 2013 as an open source, non-profit initiative to increase access for easy-to-use, low-cost RF hardware and to drive innovation in the sector.

Myriad-RF boards use field programmable RF (FP-RF) transceivers to operate on all mobile broadband standards—LTE, HSPA+, CDMA, 2G—including all regional variants; and any wireless communications frequency between 0.3 and 3.8 GHz, which includes the regulated, licensed bands and unlicensed / whitespace spectra.

Lime is seeking to increase involvement and design contribution from the general RF design community — including both hobbyists and professional system designers.

Myriad-RF-1 measures approximately 5x5 cm, uses a 5 V power supply and is software configurable to operate from 300 MHz to 3.8 GHz and on 2G, 3G and 4G communication networks. Pre-built boards will initially retail for $299 or less.

"We've had a lot of interest in the project since its launch, and Richardson RFPD's involvement is a great endorsement," said Ebrahim Busherhi, Lime CEO and Myriad-RF creator. "RF has needed an open source model for a long time and as people from outside the RF sector begin to work with such technologies new inventions will come to the fore that overcomes the problems they face, but we had no idea existed."

http://myriadrf.org   www.limemicro.com   www.richardsonrfpd.com   (130167-V)
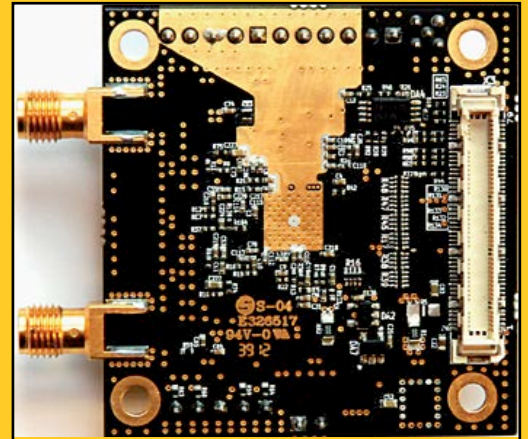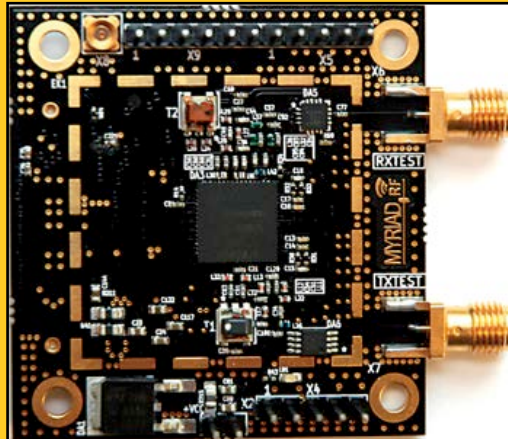
R&S High-Performance Vector Signal Generator

The R&S SMW200A high-performance vector signal generator from Rohde & Schwarz enables faster time-to-market, improves end-device quality, and exceeds important 2G, 3G and 4G digital standards and applications. Featuring versatile configuration options, the range of applications extends from single-path vector signal generation to multichannel MIMO receiver testing. The R&S SMW200A vector signal generator is the only product on the market that provides a baseband generator, RF generator and real-time MIMO fading simulator in a single instrument.

The vector signal generator covers the frequency range from 100 kHz to 3 GHz or 6 GHz, and features an I/Q modulation bandwidth of 160 MHz with internal baseband. Exceptional modulation and RF characteristics make it ideal for developing high-end components, modules and complete products for wideband communications

systems such as LTE-Advanced and WLAN IEEE 802.11ac. The generator performs especially well in verification of 3G and 4G base stations, as well as aerospace and defense applications.

The R&S SMW200A can be equipped with an optional second RF path for frequencies up to 6 GHz and with a maximum of two baseband and four fading simulator modules, giving users two full-featured vector signal generators in a single unit. Fading scenarios, such as 2x2 MIMO, 8x2 MIMO for TD-LTE and 2x2 MIMO for LTE-Advanced carrier aggregation, can be easily simulated. Previously, this had required complex setups consisting of multiple instruments.

Higher order MIMO applications such as 3X3 MIMO for WLAN or 4X4 MIMO for LTE-FDD are easily supported by connecting a third and fourth source to the R&S SMW200A. The R&S SGS100A are highly compact RF sources that are controlled directly from the front panel of the R&S SMW200A. Overall this solution takes up considerably less space with only a total of five height units for 4x4 MIMO receiver tests and provides correctly encoded baseband signals, real-time channel simulation, AWGN generation, and phase-locked coupling of multiple RF paths, if required.

Options for every important digital communications standard are available from the start: LTE, LTE-Advanced, 3GPP FDD/HSPA/HSPA+, GSM/EDGE/EDGE Evolution, TD-SCDMA, CDMA2000®/1xEV-DO and WLAN IEEE 802.11a/b/g/n/ac. The standards run directly on the R&S SMW200A—without having to connect an external PC—making it possible to vary signals or specific parameters quickly and easily, as required to test multi-standard radio base stations, for example.

www.rohde-schwarz.us

(130223-V)

Manchester and NRZ Configurable Protocol Decoders

Teledyne LeCroy Corporation Manchester and NRZ (non-return-to-zero) configurable protocol decoders enable users to specify a broad range of physical layer characteristics for Manchester- or NRZ-encoded signals. The decoders define the grouping of bits into words, and words into frames, which makes short work of analysis for custom and/or proprietary protocols based on those generic encoding schemes. Decoded information is then shown in a color-coded overlay directly on top of the physical layer waveform.

Many of today's data-communication protocols are built on Manchester or NRZ encoding. Protocols like this range from specialized buses such as Digital Addressable Lighting Interface (DALI) for control of building lighting and the Peripheral Sensor Interface 5 (PSI5) used to connect sensors to controllers in automotive applications, to proprietary, custom buses used for non-standardized applications. In all of these cases, basic Manchester and NRZ schemes are modified to create the more complex, specialized protocols. Designers around the globe are developing and debugging systems with these protocols and looking for bus analysis tools to simplify the process.

The new protocol decoders aid in the process of designing and debugging such custom protocols by providing broad flexibility in terms of physical layer characteristics, protocol word and frame structure as well as other parameters. Users may specify bit rates from 10 bits/s to 10 Gbits/s. Idle states, sync bits, header and footer information can all be configured to decode custom preambles or CRC details. Decoding is highly flexible: data mode can be in bits or words; viewing in hex, ASCII, or decimal; and bit order may be either LSB or MSB. Decoded information is displayed with a color-coded overlay which expands or contracts as the user adjusts the oscilloscope timebase or zooms in on the waveform for more details.

Powerful search capabilities allow users to quickly search long captures of decoded Manchester and NRZ waveforms for specific bus details such as data, sync or interframe gap. Decoded data is conveniently displayed in an interactive table. Clicking on any line in the table opens a zoomed view of that instance in the waveform.
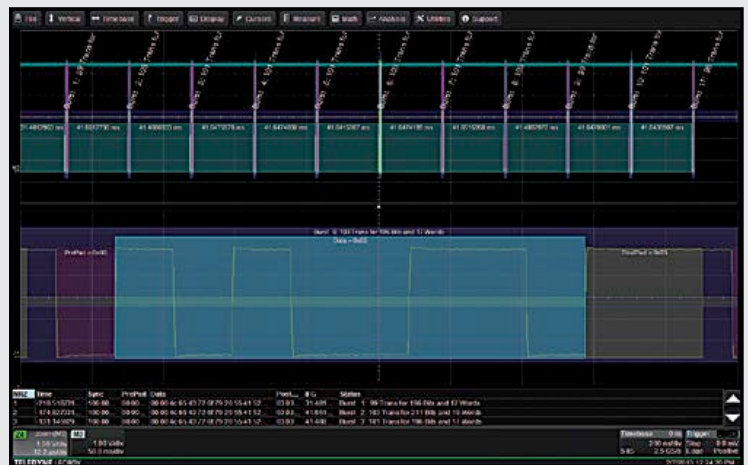
teledynelecroy.com/europe   (130167-I)

Care Robots
The future of health care

Johan Hoorn
(photo: Waag Society CC BY 2.0)

By **Tessel Renzenbrink** (Elektor Editorial TTF)

The second time I met Alice, she was already able to stand up. When she smiled, I involuntarily smiled back. It took a while for me to realize that I was sending nonverbal signals to an entity that was not able to receive them. That says something about the Alice robot, and it says something about me. The facial expressions of the robot are so refined that I responded to them automatically as a social being.

The meeting with Alice took place in the lab of the Services of Electromechanical Care Agencies (SELEMCA) project, which has its quarters at the Free University of Amsterdam [1]. The project team is studying how intelligent systems, such as robots, can interact with their users in a more human manner. The social issue underlying this project is the growing demand for care services. As a consequence of the ageing of the population, the number of people needing care will keep rising faster than the available number of care professionals. To be able to offer people adequate care in future, work is currently underway on technological solutions that could take over some of the care tasks. To make dealing with a technological care system pleasant for users, SELEMCA is developing the human-friendly I-Care system for care services.

Johan F. Hoorn, who holds doctorates in literature and science, is the principal investigator and project manager of SELEMCA. He was enthusiastic when talking about the goals, achievements and obstacles of the project: "The core of SELEMCA is the scientific investigation of intelligence, emotion and creativity. This is surrounded by a shell of machine code and machine behavior, consisting of a number of programs that can simulate these

capabilities. There are also specific functionalities – things that can be meaningful to someone or actions that someone can perform. They collectively form the I-Care system, which runs in the background. Finally, there is the interface that makes the I-Care system visible to the outside world."

Machines with human capabilities

An example of how this layered structure is elaborated in practice is provided by research into the emotional component of moral reasoning. A robot that acts strictly according to an ethical code will be experienced by humans as coldly rational and therefore threatening. In a scientific article about Moral Coppélia, to which Johan contributed as a co-author, this was illustrated using the cart and footbridge dilemmas [2].

A cart traveling at dangerously high speed is heading along a railway track towards a group of five people. By throwing a switch, the cart can be directed onto a track where just one person is standing. The choice for the moral agent is to take action to save the lives of five people at the expense of the life of one person, or to allow the cart to continue on its course, resulting in five fatalities. In another scenario, the moral agent is

standing on a footbridge next to another person. Here again the cart is threatening five people, and this time the choice is whether to throw the one person off the bridge in order to stop the cart. Although taking action results in a five to one ratio of living and dead persons in both cases, people will generally choose to throw the switch but will draw the line at actively throwing someone from a bridge. This is because they do not reason on the basis of purely ethical principles, but also let emotions play a part in their moral decisions. By contrast, a robot with purely rational moral reasoning will always sacrifice the one person for the benefit of the larger number.

People would not like a robot that throws people off bridges, so Johan and his colleagues are developing a system that integrates emotional intelligence into moral reasoning. Systems of this sort, which simulate human capabilities such as affection, moral reasoning and creativity, are built into I-Care and are expressed in the functions offered to care recipients. If a patient with a broken leg does not want to eat, the robot respects the patient's autonomy and leaves the decision to the patient, but in the case of an Alzheimer patient with reduced autonomy, the robot would offer the food again. Creativity is also a significant aspect. Instead of repeatedly putting the plate in front of the patient, which would probably just provoke more and more resistance, the robot can try an alternative approach, such as taking a spoonful of food and pretending it's an airplane.

Alice en DARwIn
The interface that makes the I-Care system visible to the outside world is an important element. According to Johan, the interface can take virtually any imaginable form. It can be a robot, a toy, a doll or a virtual agent on a screen, but what's behind it is always the same system. It does not have to look like a person, but it does act like a person. A coffee machine, for example, could act as an avatar of the I-Care system. Users may think that they are working with three different devices, but in fact they are simply interacting with the I-care system in three different manifestations. After all, the real meaning of "avatar" is a god in human form, such as Vishnu.

The Alice robot is one of the avatars in which the I-Care system can manifest itself. Thanks to the human facial expressions of the robot, many

users find it a nice way to communicate with the system. However, in terms of physical development Alice is still at a relatively rudimentary stage. It can stand up, but it can hardly perform any actions. Alice's companion DARwIn-OP, whose

Alice en DARwIn-OP
(photo: Waag Society CC BY 2.0)

name is short for "Dynamic Anthropomorphic Robot with Intelligence – Open Platform", is a lot more agile and can perform physical tasks.

However – as Johan mentioned – robots are not the only type of interface. In the SELEMCA lab they are also working on an interactive bicycle. Alzheimer patients are not good at following therapy programs. After they sit down on a home trainer to get the required exercise, they quickly become distracted and get off the trainer. Johan and his team are developing a virtual environment that gives the patient the feeling of cycling through the city and increases their attention span. The team would like to extend this to allow the patient to cycle virtually alongside a close friend or relative, by establishing an online link to another person, such as the patient's son, who in reality is biking to work. This gives the patient physical exercise together with human contact, without the risk of ending up under a bus. The companion cyclist is visualized on a handlebar screen as an avatar. Having the companion act as an interface for I-Care gives the system very human traits. During the day, the I-Care system in its various manifestations cares for the patient without the conscious awareness of the patient.
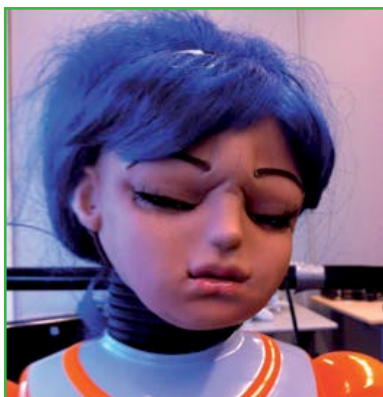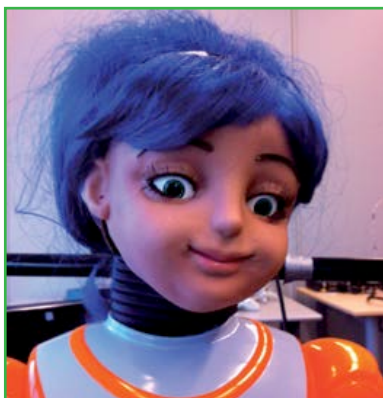
Robot Alice

DARwIn-OP
(photo: Waag Society CC BY 2.0)

Fashioning the future today

It's essential to build I-Care as an open, modular platform, according to Johan: "Everything we develop is open and available to the entire world. What we offer is a structure or abstraction, and what you attach to this structure is up to you." That applies equally well to users and developers. If a company in the industry wants to offer its own module and wishes to screen off part of it in order to make a profit with it, that is possible. "I like to describe the lab as a sort of cathedral with lots of little shops clustered around it, like the ones you see around old cathedrals where you can buy things that communicate the religious message. In this case, we would like to see interface designers, robotic sensing companies and electromechanical companies set up shop around the lab. Almost literally, so that there is face-to-face contact every day and the knowledge about I-Care that we have here can be put into practice by the companies and industrial organizations."

"That's the sticking point right now, because things are very quiet on the commercial side. It's a bit strange, because we are certain that there will be market for this in ten years. You hardly need to do any market research, because we worked with the end users in the development process. Care providers and people in need of care have personally contributed to the concept that we have created here. For the government, this offers a solution to a growing problem, and for companies it offers a business opportunity, so I don't understand why people are so reluctant to run with it. What we do here arouses more interest in Hong Kong and South Korea than here in Europe. Here everybody says, "Very interesting, really special, good job", but that's it. We lack a real innovation climate. People talk about innovation all the time and there are a thousand committees, but all the committees get in the way of innovation. I don't want committees; I want effective action."

"In terms of technology, a lot is already possible with robotics, but there is a lack of cooperation. Alice has well developed facial expressions, but the body of the robot is fairly limited. If you look at DARwIn, the body motion is quite good but it doesn't have any facial expression. The machines developed by the DARPA projects in the USA can kick you without falling over – they recover their balance and keep on walking – but they are totally lacking in creativity. There are all sorts of bits and pieces that work well on their own, but we still don't have an integrated platform. What we need is for all these people to get together and integrate everything that is already possible. You would be amazed at the results – it's unbelievable what you could do then."

(130039-I)

Internet References

[1] http://crispplatform.nl/selemca/selemca

[2] http://dare.ubvu.vu.nl/bitstream/handle/1871/38598/Moral%20Coppelia%20IBERAMIA%20Proof%2076370442.pdf?sequence=1

[3] www.crispplatform.nl

[4] http://waag.org/en

Philips PR9150/PR9151
Surface Roughness Testers

Just scratching the surface of 1950s metalworking

By
Jan Buiting,
Editor-in-Chief

Most electronics people I know hate machining and mechanical work in general, and most metalworking addicts I know take a dim view of electronics. Historically it's at CAD & CAM where the two interests were seen to get along somehow. Recently e and mech are friends in that hot area called 3-D Printing.
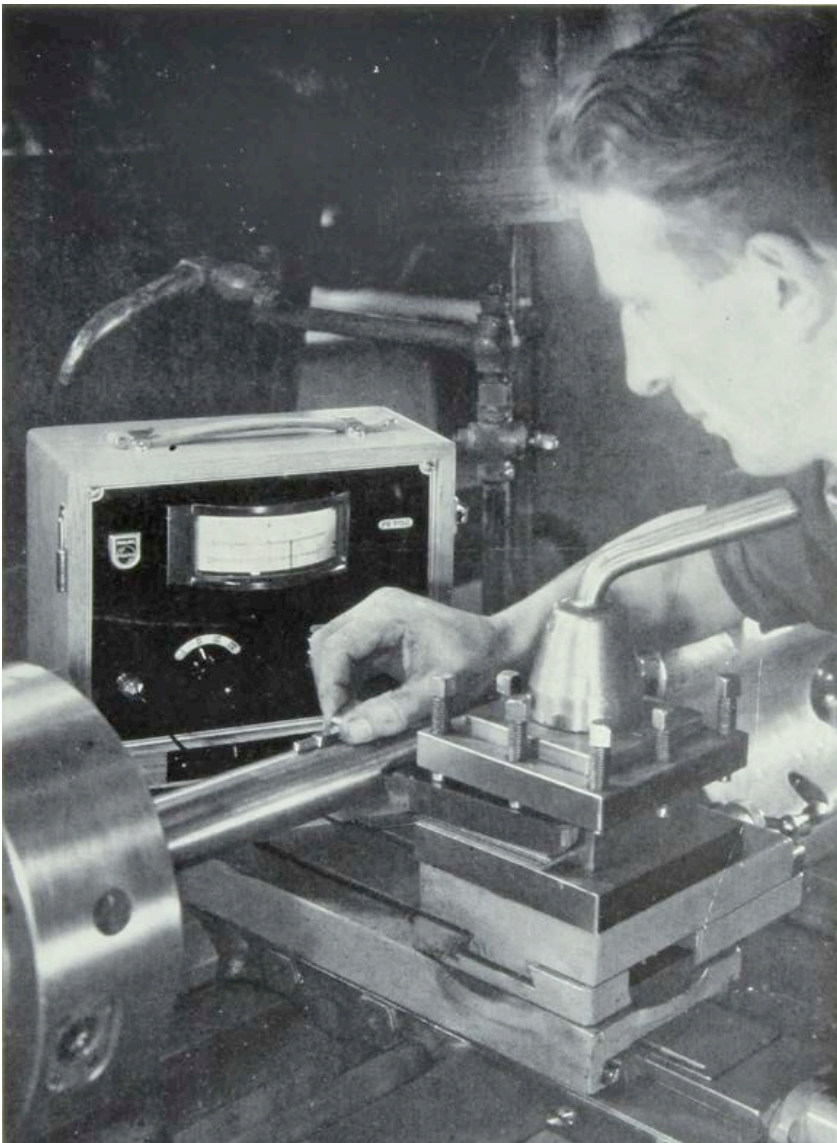
In the mid-1950s, the Philips Netherlands company was renowned for the outstanding quality of all mechanical assemblies and parts in their test & measurement and communications equipment. This must have been due largely to the scientific foundations laid by the renowned Philips' Physics Labs (Natuurkundig Laboratorium) where some of the most brilliant physicists and mechanical engineers were given total freedom to perfect their art—with no lack of funding either.

Again in the mid 1950s Philips was a strong player in the sub-scientific class of test and measurement equipment—the stuff you'd find in laboratories and workshops of the non-hobbyist type. Philips' 'PR' series of equipment was electronic for sure, but not limited to electronics—serving the chemical, medical and mechanical industries and research institutions, which represented a large and lucrative market at the time. Examples include acidity testers and meters for fluids conductivity, sound pressure, megavolts, all sorts of gases, vibration, Xrays… you name it.

Back in 1952, Philips Physics Labs worker G. W. van Santen got all worked up by the muddle of standards the 'mechies' threw at each other when talking about surface roughness (SR) of finished metal products, let alone expressing SR in values everyone could understand. At one end of the spectrum you had extremely expensive instruments operated by a few happy scientists, while at the other end, old hands at metalworking, eyes closed, used their nails to gauge the smoothness of the finish after milling, honing and polishing their workpieces. Remarkably, many of us can perceive surface roughness down to about 40 µm or about the thickness of human hair.
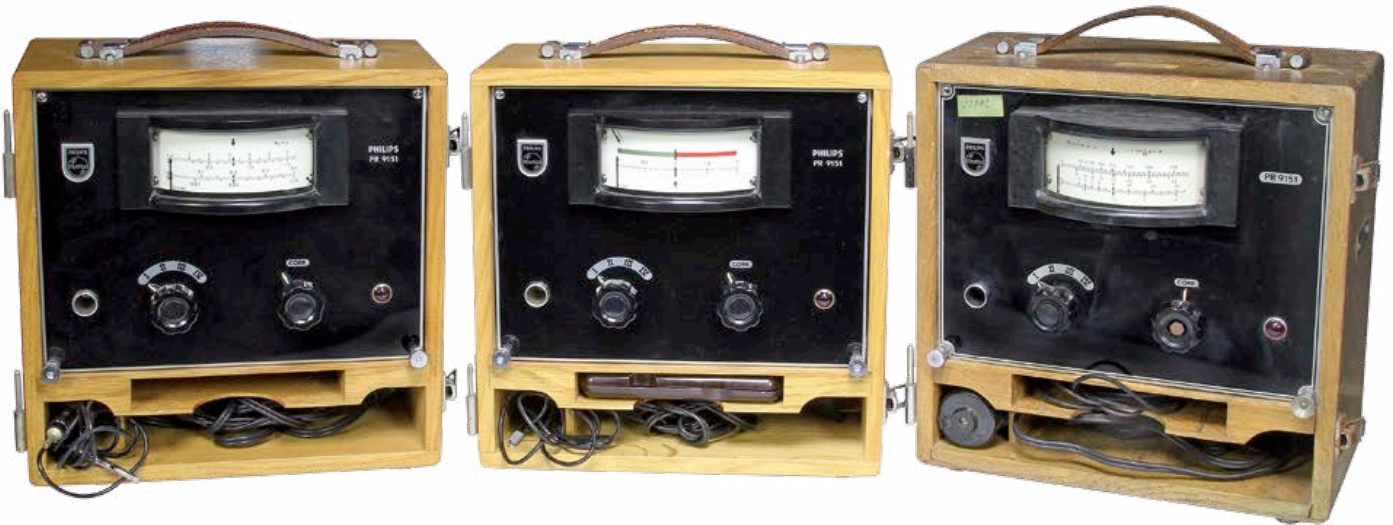
Roughness confused & defined

Here now follows a tale of horror on standards not unlike the Tower of Bable. Try to do a time warp to 1952, okay?

Figure 1 shows the profile of a machined surface. The drawing is vertically expanded with respect to the length of the workpiece. For the surface roughness to be recorded unequivocally the quantity should be normalized internationally. Regrettably, that is not the case yet.

Surface roughness is defined differently in countries, as follows:

- **maximum roughness height**, H_{max}, i.e. the height difference between the highest peak and the lowest valley. This forms the basis of German standard DIN 4762. H_{max} is expressed in microns.

- **effective roughness height**, h_{eff} or h_{rms}, i.e. the square root of the mathematic average of the squared deviations h from the baseline (L, average level), measured over a defined length:

$$h_{eff} = \sqrt{\frac{h_1^2 + h_2^2 + h_3^2 + ... h_n^2}{n}}$$

American standard ASA B46 mentions h_{rms} throughout, with the value given in micro-inches.

- **average roughness height**. h_{avg}, i.e. the mathematical average of the absolute values of the deviations h from the baseline measured over a defined length:

$$h_{avg} = \frac{h_1 + h_2 + h_3 + ... h_n}{n}$$
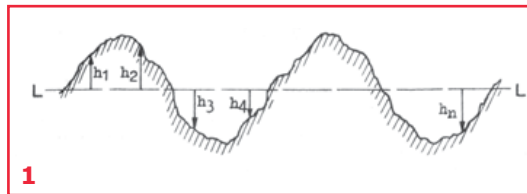
British standard BS 1134 defines average roughness height (center-line average; C.L.A.) as the standard, with the actual values expressed in micro-inches just as with h_{rms}.

The Dutch High Commission for Standardization also recommended average roughness height as the standard, arguing that (a) the definition of h_{avg} is beyond discussion; (b) the term average value is easier to clarify in a workshop; and (c) the quantity can be measured directly with electronic means. Additionally, the difference between h_{avg} and h_{eff} is negligible in practice.

Unit of roughness

For smooth (metal) surfaces the micron (micro meter; μm; 10^{-6} m) is a fairly coarse unit that's likely to pester users with small numbers. And then the milli-micron sadly is too small giving impractically large numbers. The micro-inch is a good unit in between large and small, as most finished surfaces will be within 1 and 500 micro-inch in terms of surface roughness. For example:
smooth surface   0.05 μ = 2 μ'' = 50 mμ
rough surface   6.3 μ = 250 μ'' = 6300 mμ
Alas, in countries like Holland where the metric system dominates, a reference standard based

2

1. stylus
2. barium titanate element
3. adjustment screws
4. handle
5. connexion cable

3
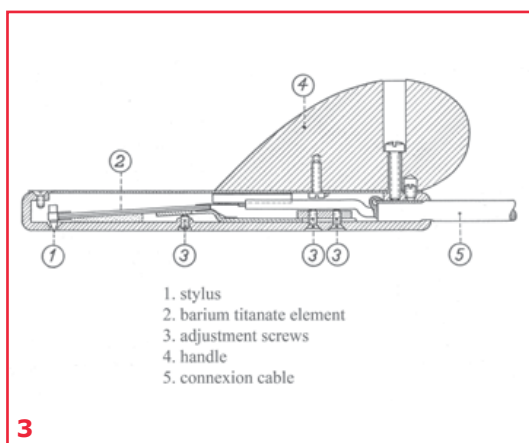
on inches (i.e. a non-metric unit) is unlikely to be adopted. As a workaround the 'ru' got proposed as the unit of surface roughness, with these linguistic ploys applied: **ru**gosité; **ru**wheid; '**r**oughness **u**nit' to convince French, Dutch and English speaking users respectively.

A surface has a roughness of 1 ru if the average roughness height R_{avg} = 1/40 micron ≈ 1 micro-inch.

With this problem out of the way (*and a chuckle on the persistent recurrence of the inch*) roughness classes R_1 through R_6 were set up along with a set of symbols the people at the lathes and milling machines were supposed to recognize from construction drawings. It was an extensive set of solid and open triangles, and solid and open circles to play with and I suppose you could learn them just like we learn our kilo-ohms and milliamps.

The PR9150/PR9151

Hey the only difference between the two is the use of American tubes in the PR9151 (like a a 6X4 for an AZ41).

Regrettably I've no schematic of the PR9150 or 9151 to share with you. But then I always open up equipment—see **Figure 2**. Pristine interior after 60 years. Let's guesstimate. The electronics probably amounts to an adjustable input attenuator (for roughness classes R2 though R5), an

Table 1. PR9150 / 9151 versions			
Type	**Meter Scale**	**Measurement Ranges**	**Roughness Samples**
PR9150/01 PR9151/01	Scale /01 (Fig. 5a) for • comparison; Red= approved, Green = rejected • multiplication (with indicated factor, for SR w.r.t. reference workpiece)	1 – 300 ru, divided in 4 sub ranges	Reference piece or PR9180/00 (R_a calibration values: 125 – 32 – 8 and 2 ru; 3.2 – 0.8 – 0.2 and 0.05 µ)
PR9150/02 PR9151/02	Scale /02 (Fig. 5b) calibrated for R_a in ru (=C.L.A. in µ''); scale shows SR values recommended as standard	I: 50-280 ru II: 10-70 ru III: 3-16 ru IV: 1-4 ru	PR9180/00 (see above)
PR9150/03 PR9151/03	Scale /03 (Fig. 5c) with decimal division, calibrated for R_a in ru.	I: 50-250 ru II: 10-60 ru III: 3-16 ru IV: 1-4 ru	PR9180/00 (see above)
PR9150/04 PR9151/04	Scale /04 (Fig. 5d) calibrated for R_t = H_{max} in µ	I: 5-25 µ II: 1-6 µ III: 0.3-1.6 µ IV: 0.1-0.4 µ	PR9180/02 (R_a calibration values: 12.5 – 3.2. – 0.8 and 0.2 µ
PR9150/05 PR9151/05	Scale /05 (Fig. 5e), calibrated for R_a in µ	I: 1-6 µ II: 0.3-1.6 µ III: 0.1-0.4 µ IV: 0.02-0.1 µ	PR9180/00 (see above)

amplifier, a rectifier and a moving coil meter. Can't be wrong much.

The electronics is housed in a beautifully made wooden case with a leather carrying handle and a hinged cover. The lot weighs approximately 15 lbs (6 kg). Whenever I open up the case with people around they think I am going to take radiation measurements.

The Probe

The crux of the instrument is the probe shown diagrammatically in **Figure 3**. It contains a piezo electric crystal made from barium titanate, secured to the probe case at one side, and terminated with a stylus at the other. The stylus is a synthetic sapphire needle with a hardness of 2,000 VPN and an end radius of about 60 μm. This allows the roughness profile of finished surfaces to be 'probed' down into the deepest valleys (the slopes nearly always exceed 150°). The tip of the stylus can easily be seen and felt. If the PR9150/9151 had a loudspeaker it could probably be used to play Joe Cocker or Janis Joplin records. In practice, the probe is so small it can be used for measurements in holes down to 8 mm diameter. It is connected to the instrument with a good length of screened cable terminated in a solid, very high quality plug. The probe (**Figure 4**) has an elegantly styled Bakelite handle.

A dial for each standard

Apparently Philips were not too confident in a 'one size fits all' instrument with a uniform **ru** readout as proposed by what must have been a lot of committees and bureaucrats. Clearly, a case of the industry finding ways to deal with "Washington and Brussels". I was amazed and intrigued to find five different types of PR9150/9151 mentioned in the manual that came with the first instrument I obtained about two years ago. I've summarized the differences in **Table 1**.

Each version has its own meter scale to suit different markets, Chief Inspection Officer (CIO) preferences and applications I suppose. **Figures 5a-d** are an attempt at reproducing the various meter scales from my only copy of the manual. I have PR9151 instrument versions /01, /02 and /05, also three probes (one with a broken stylus), one manual, and one…

Original Calibration Box!

While the PR9150 and 9151 aren't rare birds, the calibration box mentioned in the manual is.

6
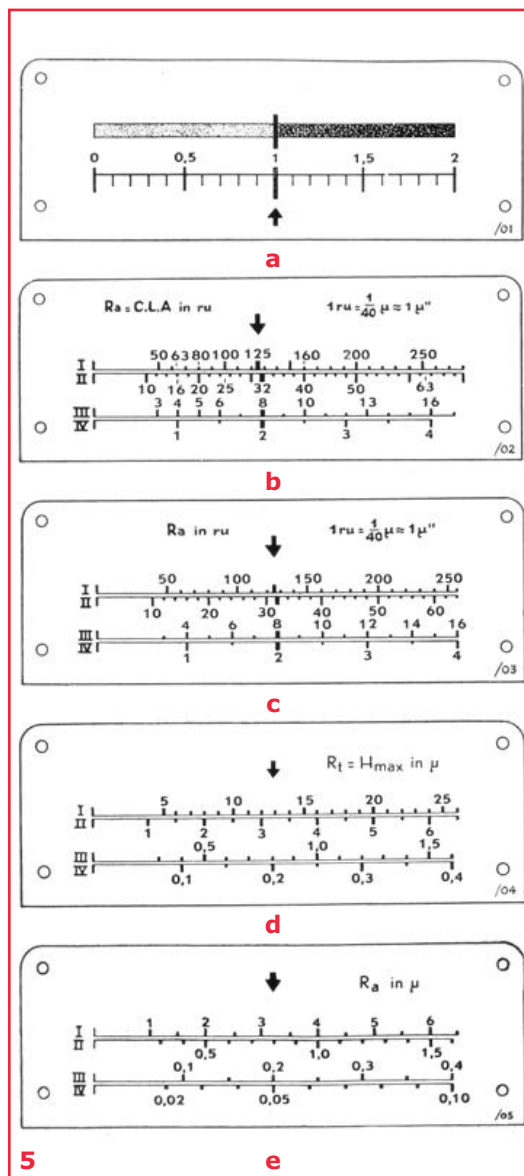
on the two 'roughest' samples. The other two pieces require a bit more effort to tell apart. All pieces have thin chrome plating which does not affect their surface roughness. Anyone have a PR9180/02 box?

In practice

Having calibrated the instrument using the samples in the box you are supposed to put the probe on the surface to be tested at an angle of 90 degrees with respect to the direction of finishing. You can move the probe up and down 2-3 times per second over a distance of 1.5 – 2 cm, or have the workpiece turn slowly (say, in a lathe or mill). A minimum probe or object speed of about 4 cm/s is required. The underside of the probe has to touch the surface across the full length. Although the PR9150/9151 has high-pass filters to eliminate the effects of uncontrolled hand movement, some practicing is required to avoid sudden changes in the meter deflection. And yes, the testing is damaging—the stylus leaves scratches.

In a rather lengthy chapter in the PR9150/9151 manual Philips conclude that their instruments can achieve an accuracy of about 20%, which is "outstanding, considering that extremely accurate and costly roughness meters achieve about 15% precision due to the non-homogeneous structure of ordinarily finished surfaces".

Today, surface roughness (SR) meters are aplenty on Ebay & Co. They use vastly improved technologies over recording bumps with a piezoelectric stylus and a tube or two like they did in 1955. Still, you do not see an awful lot of them around in metal workshops, not even where they do cylinder bore honing and precision polishing. I guess the good old 'thumbnail' method coupled with a solid amount of craftsmanship and experience still hold their own against K-dollars worth of electronics and the Internet.

(130057)

I was eventually able to get one included with the third PR9151 I came across, which was also the scruffiest.

While you can do comparative measurements with the PR9150/9151, Sales dept., CEO, CFO and CCO will insist on having absolute numbers and benchmarks! So you need to calibrate your instrument before taking readings on samples. The PR9128/00 calibration box shown in **Figure 6** is made from strong Bakelite. Also note the solid and open triangle symbols to denote four classes of surface roughness. The exact SR values are hand written. I was easily able to see the degree of finishing of the four metal pieces in the box, and feel the grooves due to milling

ELEKTOR Preferred Suppliers

Tel. 1-978-281-7708   **Fax 1-978-281-7706**   **Email ElektorUSA@smmarketing.us**

Hexadoku   Puzzle with an electronic touch

Summer in electronics land does not mean Hexadoku takes a vacation. Try to crack the puzzle on the beach, in an airport lounge—anywhere you think you need to take your mind off the daily grind of reporting, debugging and meeting deadlines. Enter the right numbers or letters A-F in the open boxes, find the solution in the gray boxes, submit online, and you automatically enter the prize draw for one of four vouchers.

The Hexadoku puzzle employs numbers in the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

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

Participate!

Before September 1, 2013, supply your personal details and the solution (the numbers in the gray boxes) to the web form at **www.elektor.com/hexadoku**

Prize winners

The solution of the May 2013 Hexadoku is: **3D1AE**.   The Eurocircuits $140.00 voucher has been awarded to Dirk Neerijse (Belgium). The Elektor $60.00 book vouchers have been awarded to József Nagy (Hungary), Sigrid Scheel (Germany), and Joe Young (Canada).

Congratulations everyone!

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

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

●Gerard's Columns

Penny Wise and Pound Foolish

By Gerard Fonte (USA)

What this means is that the larger concerns are neglected in favor of the small or immediate concerns. This is a common human trait. It's much easier to address small, urgent problems than big, important ones.

Up in Smoke

Hobbyists are perpetually short on cash (me too). So we try to do the most with the least. There's nothing wrong with this unless you end up spending more over time by trying to cut corners. A classic example of this is the humble power supply. I'll bet that the large majority of readers don't have a dedicated, adjustable voltage, current limited power supply for their bench. I didn't for a very long time. I used the typical cheap ones based on the ubiquitous three-terminal regulators. And I accepted the occasional fried component when I made a wiring error. But these smokers add to the cost of using a non-current regulated power supply.

Let's look at the numbers. You spend $20 for a cheap kit. There's no voltmeter so either you have to add one (another $10) or go through the bother of measuring it every time you use it. (How much is your time worth?) Suppose you destroy one $2 part every 3 months. Over 5 years you've blown $40 in parts. Total cost: $70 (including $10 for the meter you finally added because you got so annoyed at having to measure the voltage every time).

However, for about $60 you can get a nice current regulated power supply (from a number of on-line sources). It includes a voltmeter and ammeter. (Knowing how much current your project actually draws is very important and useful information.) And by limiting the current to a very low value, there are no more bad smells coming from your circuit when you make a mistake.

The point is to examine the future before you make a decision. It's a lot less expensive to buy something marginal now, but it will cost more in the long run. It makes sense to spend more for quality tools that you use a lot (soldering iron, meters, etc.). While you may be able to get by with penny-pinching equipment, the results will be better and faster with superior tools. That also makes your hobby much more fun.

Upscale

The same thing happens at the other end of the economic spectrum as well. Large companies consistently fail to examine the consequences of short-term decisions. Managers and VPs work hard to provide the largest profit for the year-end report to the share-holders. You can't argue with success, can you? However, the situation is a bit more complex in this situation. Here there is a clear conflict of interest.

Human beings are naturally selfish creatures. We all want as much as we can get. And executives often get a sizeable portion of their salaries in the form of performance bonuses. If the company does well that year, they get a big bonus and perhaps a promotion. If the company does poorly, there are smaller pay-outs and maybe it's golden parachute time. Obviously, it is in the best interest of the executive to provide the maximum performance for that year so he or she can get the biggest paycheck. But this may not be the best for the company.

For example, the Research and Development (R&D) department rarely shows a profit for any year. In terms of straight economics, R&D is a major drain on the company. Money goes in to support expensive employees (with PhD's) and costly equipment. But no money ever seems to come out. If you are interested in a quick boost to the bottom line, cutting R&D funds is the way to go. The yearly profits are increased in direct proportion to the reduction in R&D. Obviously, this is a very short-sighted approach to increasing the company's profits. However, this is done with regularity.

Down to Earth

Human nature is amplified with powerful people. Politicians, and the wealthy people that influence them, can manipulate matters on a world-wide scale. But politicians are people, too. They each have an agenda that they think is important. The only way that they can pursue their goals is to be in office. In order to remain in office, they must please the voters on a regular basis. So, naturally, they are under pressure to provide immediate results that will get them re-elected. Important long-term issues that benefit the country are neglected in the urgency to maintain popularity with the political base. Even the issues with global reach are postponed.

There are a growing number of academics who think that the human race may face a serious downward population adjustment or even extinction within a century or two (or less). The primary possible causes are: global population, climate change, energy availability, fresh water supplies, nuclear war, or disease (man-made or otherwise). There is no doubt that humans have the power to exterminate themselves by simply being short-sighted. This was not true even a hundred years ago. And while there is some political talk, there's not much action.

Square One

Curiously, individuals do have the capability to set and achieve long-term goals at the expense of short-term amusements. You spend four years to get your EE degree, eight for a PhD or twelve for an MD. So while human nature tends to be penny wise, we certainly don't have to be pound foolish.

(130224)

•Store
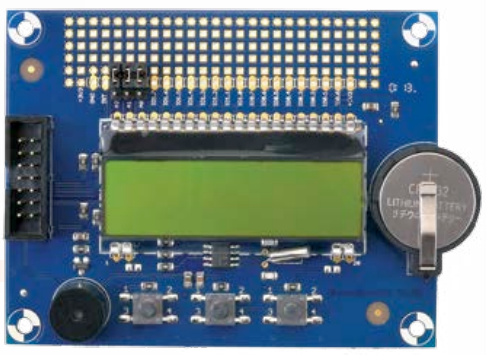

NEW

1

2

3

4

Limited Time Offer for GREEN and GOLD Members!
16% DISCOUNT + FREE SHIPPING
www.elektor.com/july

Ideal reading for students and engineers

1 Practical Digital Signal Processing using Microcontrollers

This book on Digital Signal Processing (DSP) reflects the growing importance of discrete time signals and their use in everyday microcontroller based systems. The author presents the basic theory of DSP with minimum mathematical treatment and teaches the reader how to design and implement DSP algorithms using popular PIC microcontrollers. The author's approach is practical and the book is backed with many worked examples and tested and working microcontroller programs. The book should be ideal reading for students at all levels and for the practicing engineers who may want to design and develop intelligent DSP based systems.
428 pages • ISBN 978-1-907920-21-9
£44.90 • € 49.90 • US $72.50

10 captivating lessons

2 PIC Microcontroller Programming

Using the lessons in this book you learn how to program a microcontroller. You'll be using JAL, a free but extremely powerful programming language for PIC microcontrollers. Assuming you have absorbed all lessons you should be confident to write PIC

microcontroller programs, as well as read and understand programs written by other people. You learn the function of JAL commands such as include, pin, delay, forever loop, while loop, case, exit loop, repeat until, if then, as well as the use of functions, procedures and timer- and port interrupts. You make an LED blink, build a time switch, measure a potentiometer's wiper position, produce sounds, suppress contact bounce, and control the brightness of an LED. And of course you learn to debug, meaning: how to spot and fix errors in your programs.
284 pages • ISBN 978-1-907920-17-2
£29.50 • € 34.50 • US $47.60

A whole year of Elektor magazine on a single disk

3 DVD Elektor 2012

The year volume DVD/CD-ROMs are among the most popular items in Elektor's product range. This DVD-ROM contains all editorial articles published in Volume 2012 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 favorite graphics program, zoom in / out

on selected PCB areas and export circuit diagrams and illustrations to other programs.
ISBN 978-90-5381-273-0
£23.50 • € 27.50 • US $37.90

Display, buttons, real time clock and more

4 Elektor Linux Board Extension

This extension board was developed to further propel our Embedded Linux series of articles and the matching GNUblin board. It has a display, buttons, a real time clock and 16 GPIOs. Linux devotees, switch on your solder irons. The Linux extension board includes everything needed to provide the user interface for a wide variety of projects!
Module, SMD-populated and tested board, incl.
LCD1, X1, K1-K4, BZ1, BT1 for home assembly
Art.# 120596-91
£31.10 • € 34.95 • US $50.20

LabWorX 2

5 Mastering Surface Mount Technology

This book takes you on a crash course in techniques, tips and know-how to successfully introduce surface mount technology in your
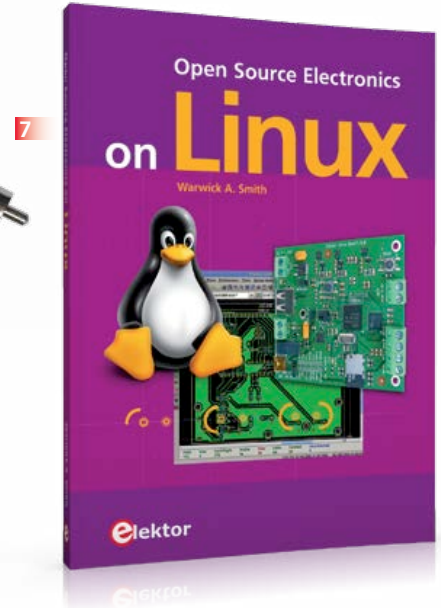
workflow. Even if you are on a budget you too can jumpstart your designs with advanced fine pitch parts. Besides explaining methodology and equipment, attention is given to SMT parts technologies and soldering methods. Many practical tips and tricks are disclosed that bring surface mount technology into everyone's reach without breaking the bank. A comprehensive kit of parts comprising all SMT components, circuit boards and solder stencils is available for readers wishing to replicate three projects described in this book.

282 pages • ISBN 978-1-907920-12-7
£29.50 • € 34.50 • US $47.60

Ultrasensitive wideband E-smog detector
6 TAPIR Sniffs it Out!

Attention boy scouts, professionals and grandfathers! This ultrasensitive wideband E-smog detector offers you two extra senses to track down noise that's normally inaudible. TAPIR — short for Totally Archaic but Practical Interceptor of Radiation — also makes a nice project to build: the kit comprises everything you need. Even the enclosure, ingeniously consisting of the PCB proper! Using the TAPIR is dead easy. Connect the headphones and an antenna and switch it on. Move it around any electrical device and you'll hear different

noises with each device, depending on the type and frequency of the emitted field.

Kit of parts, incl. PCB
Art.# 120354-71
£13.30 • € 14.95 • US $21.50

OS Hard- and Software for Electronics Applications
7 Open Source Electronics on Linux

If you have ever wanted to take advantage of the expanding field of open source software for electronics and everyday applications, this book is for you. Using the Linux OS, Warwick A. Smith guides you through the world of open source hardware and software, teaching

readers to use EDA tools and software that is readily available online, free to download. The hardware projects inside can be built using easily obtainable parts, in the comfort of your own home, on single sided PCBs, or professionally manufactured with output files generated by you. Open Source Electronics on Linux is about changing today's electronics enthusiast into empowered, savvy, discerning engineers capable of building and modifying their creations, be it solely on Linux or in tandem with your current operating system.

272 pages • ISBN 978-1-907920-19-6
£29.50 • € 34.50 • US $47.60

Further Information and Ordering: **www.elektor.com/store**
or contact customer service for your region

UK / ROW

Elektor International Media
78 York Street
London - W1H 1DP
United Kingdom
Phone: +44 20 7692 8344
E-mail: service@elektor.com

USA / CANADA

Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
E-mail: service@elektor.com

●Magazine

NEXT MONTH IN ELEKTOR MAGAZINE

Compact Audio Power Amp

Audio projects have been thin on the ground recently in Elektor. Next month we make amends with the publication of a cracker power amplifier built around a special driver IC from Texas Instruments. Operating at supply voltages up to ±100 V, a single pair of output transistors supplies more than 200 watts continuously into 4 ohms, still maintaining pretty low distortion figures. The compact amplifier board also contains switch-on delay and DC protection circuitry.
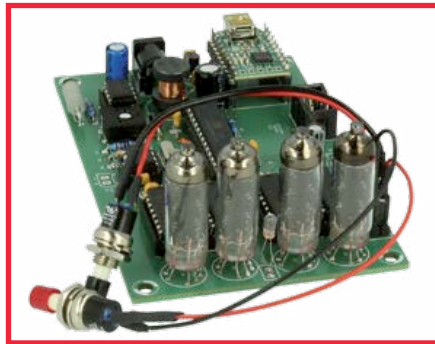
Numitron Clock

Elektor magazine is increasingly home to advanced projects built on the Arduino platform. In this case we strove to blend old and new technologies in a stylish way using an Arduino-compatible microcontroller system with some add-ons electronics to drive a couple of Numitron tubes for a digital clock/thermometer. A Numitron is a vintage electron tube that can be used as a seven-segment display. Remarkably, each segment consists of a filament. Numitrons are generally affordable, and available from several online resources.
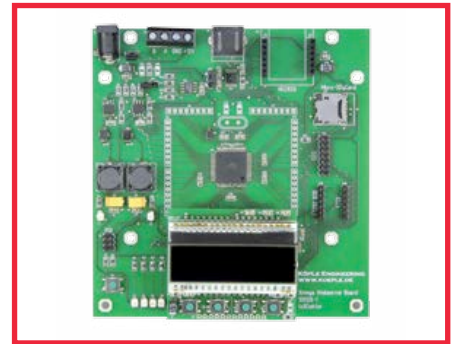
Xmega Webserver

In the next issue we present a versatile microcontroller board designed around a very powerful AVR microcontroller. In terms of I/O we have 4 LEDs, 4 pushbuttons and a (separately installed) display. For interfacing, you can choose between RS485 and various UART/TTL connectors, allowing our BOB USB-TTL converter to be connected, for example. The Embedded Extension Connector makes the board pretty versatile. The board also has a Micro SD connector, and there is room for a TCP/IP module that allows web server and other network applications to be realized.

Article titles and magazine contents subject to change; please check the Magazine tab at www.elektor.com.
Elektor September 2013 edition is published to members on August 13, 2013.

ORDERING INFORMATION

To order, contact customer service for your region:

USA / CANADA
Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
E-mail: service@elektor.com

Customer service hours:
Monday-Friday 8:30 AM-4:30 PM EST.

UK / ROW
Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344
E-mail: service@elektor.com

Customer service hours:
Monday-Thursday 9:00 AM-5:00 PM CET.

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 for your region.*

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.

TERMS OF BUSINESS

Shipping Note:
All orders will be shipped from Europe. Please allow 2–4 weeks for delivery.

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, magazines, online publications and presentations. Elektor accepts no responsibility or liability for failing to identify such patent or other protection.

Copyright
All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, discs, 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 educational 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.

MEMBERSHIPS

Membership renewals and change of address should be sent to the Elektor Membership Department for your region:

USA / CANADA

Elektor USA
P.O. Box 462228
Escondido, CA 92046
Phone: 800-269-6301
E-mail: elektor@pcspublink.com

UK / ROW

Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344
E-mail: service@elektor.com

Do you want to become an Elektor GREEN or GOLD Member or does your current Membership expire soon?

Go to www.elektor.com/member.