www.elektor-magazine.com

.magazine
March 2013 | £ 4.90

elektor

The luxury of precision within everyone's reach

500 ppm LCR Meter

● **Raspberry Pi Prototyping Board** | The 7-uP Alarm Clock | USB-IO24 Cable | **FPGA Programming** | Simple Servo tester | **Battery Indicator** ● 1956 Audion Kit ● Frontline Breaking News | **Prototype Howlers** ● Open Data ● **'hue' Lighting**

03
9 770268 451173

Crystal-free 8-bit USB PIC® microcontrollers cut system costs and power consumption

0.25% clock accuracy enables USB connectivity, eliminating the need for external crystal

Microchip's lowest-cost and smallest-form-factor USB microcontrollers (MCUs), feature pin counts of 14 to 100 pins and are the first 8-bit MCUs to integrate LCD control, battery-backed RTCC, and USB on a single chip.

Microchip's latest USB PIC® MCUs feature internal clock sources with 0.25% clock accuracy to enable USB connectivity with no external crystal. They are also the first USB MCUs to combine pin-counts ranging from 14 to 100, with high peripheral integration and up to 128 KB of Flash. The eXtreme Low Power (XLP) technology also keeps power consumption down to 35 µA/MHz in active mode and 20 nA in sleep mode.

Lowest-cost and smallest-form-factor

The PIC16F145X MCUs give you USB connectivity and capacitive touch sensing, in addition to a wide range of integrated peripherals with footprints down to 4x4 mm.

High-performance touch-sensing with USB

With an integrated Charge Time Measurement Unit (CTMU) and 1.8 V to 5 V operation, PIC18F2X/4XK50 MCUs are pin-compatible with legacy PIC18 MCUs, giving an easy migration to higher-performance.

USB plus LCD control and a RTCC with Vbat

The PIC18F97J94 family gives you USB connectivity with LCD control, and a battery-backed real-time clock calendar (RTCC), all on a single 8-bit PIC® microcontroller.

GET STARTED IN 3 EASY STEPS:

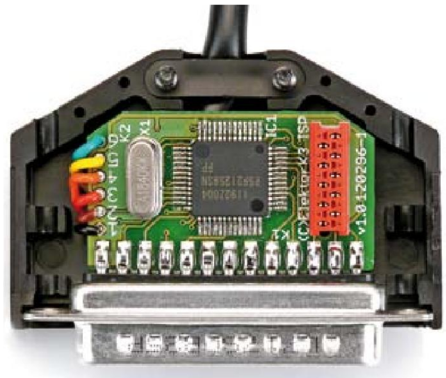
1. Choose a peripheral mix and pin count to suit your application
2. Use the free USB stacks and software drivers for faster design
3. Start developing with low-cost development kits

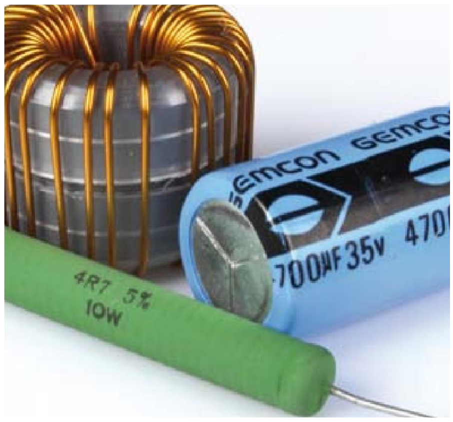
For more information, go to: www.microchip.com/get/eu8bitUSB

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

Contents

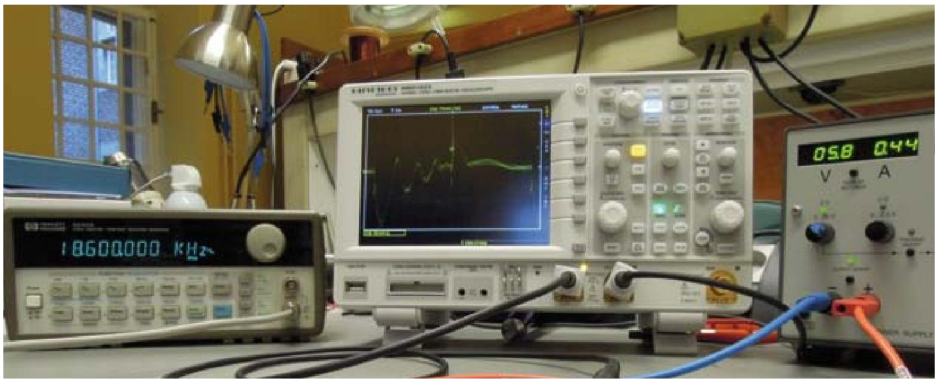


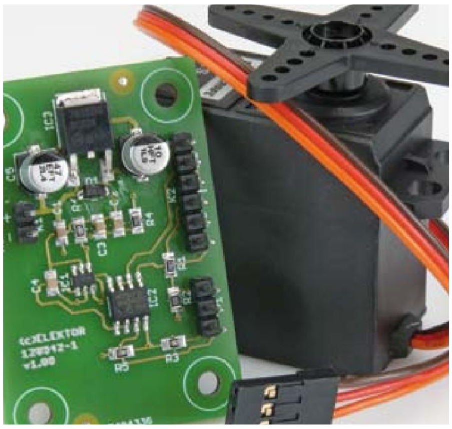


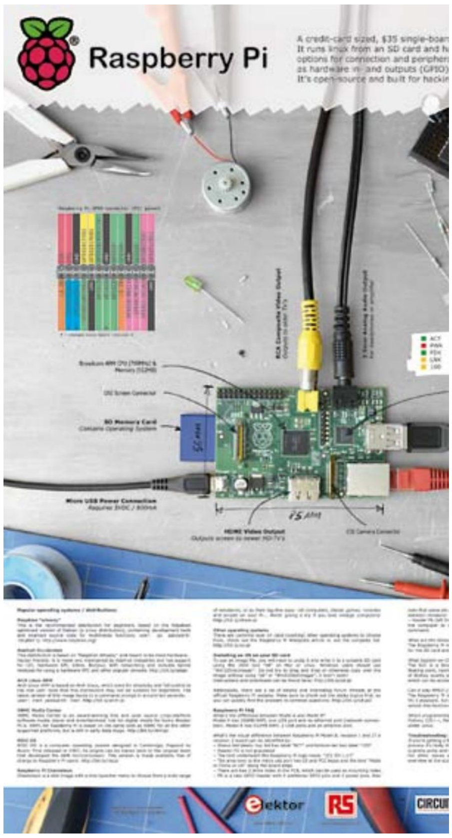
Community

Projects

Volume 39 – No. 435 March 2013







Labs

Industry

Tech the Future

Magazine

Volume 39, Number 435,
March 2013
ISSN 1757-0875

Publishers:
Elektor International Media,
78 York Street, London W1H 1DP,
United Kingdom.
Tel. +44 (0)20 7692 8344
www.elektor.com

The magazine is available from newsagents, bookshops and electronics retail outlets, or by membership.
Elektor is published 10 times a year with a double issue for January & February and July & August.

Memberships:
Elektor International Media,
78 York Street, London W1H 1DP,
United Kingdom.
Tel. +44 (0)20 7692 8344
Internet: www.elektor.com/member
Email: service@elektor.com

Head Office:
Elektor International Media b.v.
P.O. Box 11 6114 ZG Susteren
The Netherlands.
Telephone: +31 (0)46 4389444
Fax: +31 (0)46 4370161

Distribution:
Seymour, 2 East Poultry Street,
London EC1A, England.
Telephone:+44 (0)20 7429 4073

UK Advertising:
Elektor International Media b.v.
P.O. Box 11 6114 ZG Susteren
The Netherlands.
Telephone: +31 (0)46 43 89 444
Fax: +31 (0)46 43 70 161
Email: j.dijk@elektor.com
Internet: www.elektor.com/advertising
Advertising rates and terms available on request.

Bridging the gaps bit by bit

Here's an assumption up for consideration if not for revision: microcontrollers and microvolts (and -amps, -henries, -farads…) are separated by a big gap; they are two separate worlds, each with their own fan club, its members easily discernible by activity: the digital pundits 4ver eliminating bugs, the analogue guys, always chasing and combatting noise of the electric variety, and corporate too at times. Jean-Jacques Aubry's 500 ppm (that's 0.05%) LCR Meter in this edition should be a feast for LSB manipulators and microhenry addicts alike, as it successfully combines some of the best digital and analogue design techniques I have seen so far, at least in the DIY scene. Even if you do not intend to build this sophisticated instrument, there's a lot to be learned from the descriptions on how it got designed, both in terms of hardware and software.
Another apparent gap in the current electronics / embedded scene is not just revealed but also eliminated by the Raspberry Pi Extension Board article on pages 30–35. While I was easily able to find a mass of RPi software and ready-made hardware out there, I saw preciously little in the way of e-prototyping from the ground up with that nifty little computer. Tony Dixon's article I am convinced will redress the balance, literally challenging you to develop your own hardware for the Rpi, starting on a prototyping area with all bus signals and supply rails easily available on connector pins. Along the way, hopefully you'll have the stirring experience of reducing component count by clever software. If you do it bit by bit, in a low noise environment, success is assured. Please tell me and the Elektor community how you get along LCR'ing, RPi'ing and Elektor'ing.

Jan Buiting, Managing Editor

The Team

United Kingdom
Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com

USA
Hugo Vanhaecke
+1 860-875-2199
h.vanhaecke@elektor.com

Germany
Ferdinand te Walvaart
+31 46 4389417
f.tewalvaart@elektor.de

France
Denis Meyer
+31 46 4389435
d.meyer@elektor.fr

Netherlands
Harry Baggen
+31 46 4389429
h.baggen@elektor.nl

Spain
Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es

Italy
Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it

Sweden
Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

Brazil
João Martins
+55 11 4195 0363
joao.martins@editorialbolina.com

Portugal
João Martins
+351 21413-1600
joao.martins@editorialbolina.com

India
Sunil D. Malekar
+91 9833168815
ts@elektor.in

Russia
Nataliya Melnikova
+7 (965) 395 33 36
Elektor.Russia@gmail.com

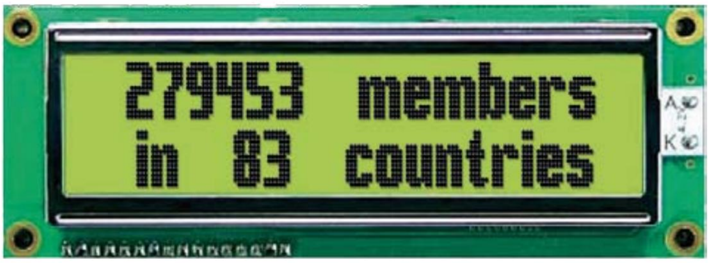
Turkey
Zeynep Köksal
+90 532 277 48 26
zkoksal@beti.com.tr

South Africa
Johan Dijk
+27 78 2330 694
j.dijk@elektor.com

China
Cees Baay
+86 21 6445 2811
CeesBaay@gmail.com

Our network

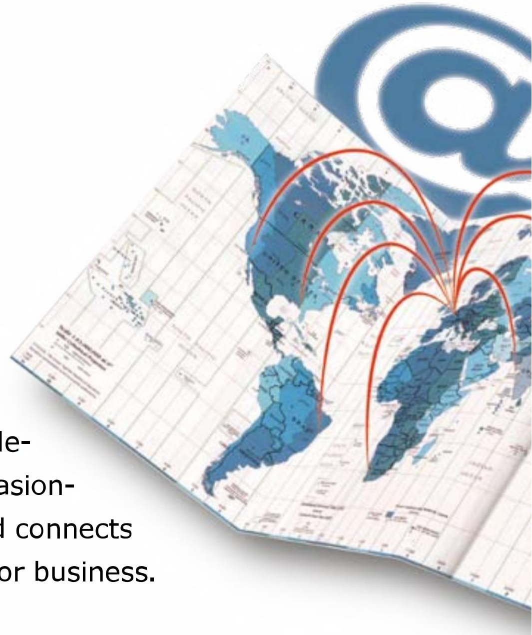
Connects you to



Supporting Companies

Not a supporting company yet?

Contact Johan Dijk (j.dijk@elektor.com, +27 78 2330 694)
to reserve your own space for the next edition of our members' magazine



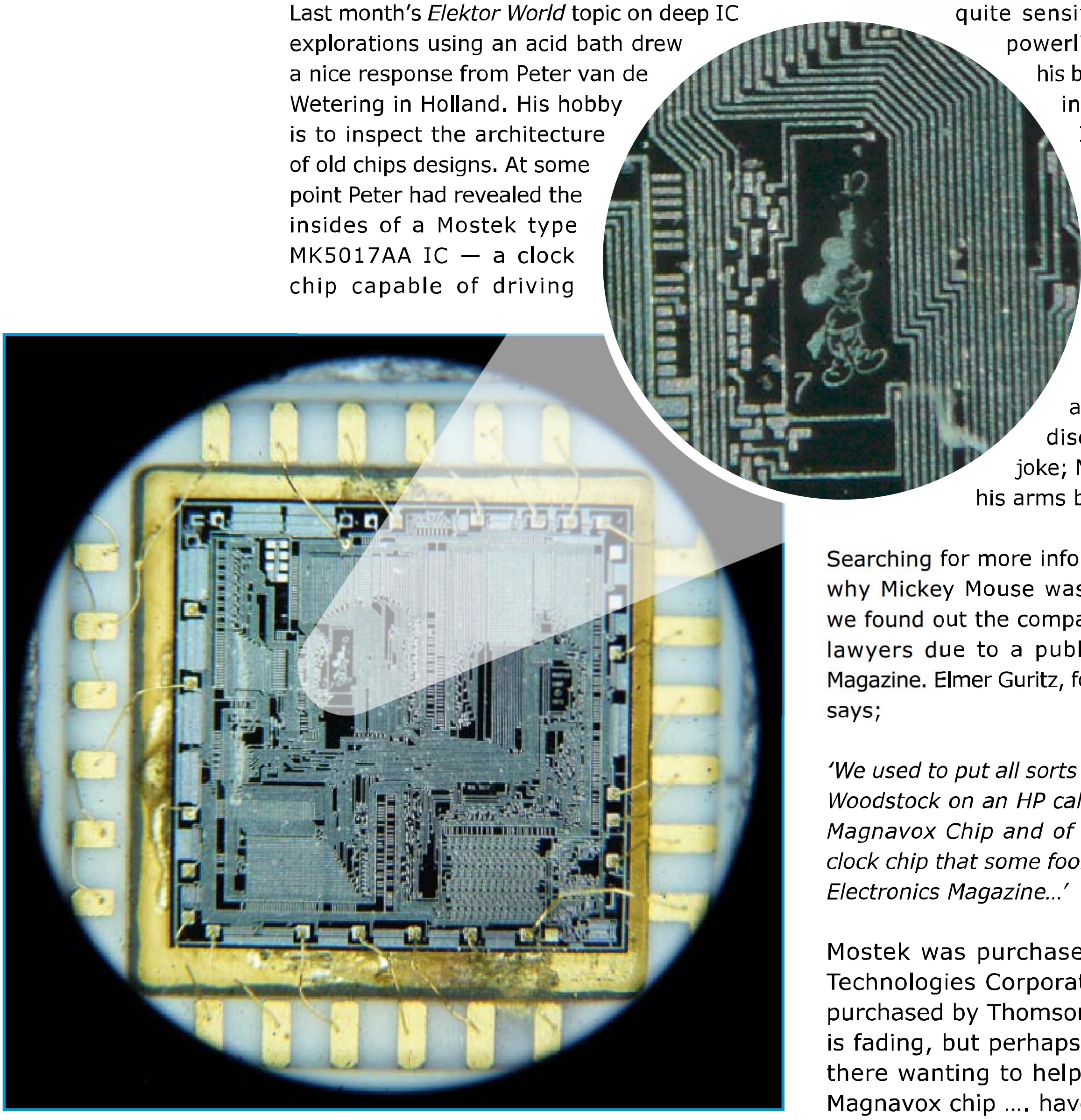
Elektor World

Compiled by
Wisse Hettinga

Every day, every hour, every minute, at every given moment designers and enthusiasts are thinking up, tweaking, reverse-engineering and developing new electronics. Chiefly for fun, but occasionally fun turns into serious business. Elektor World connects some of the events and activities — for fun and for business.

What is Mickey Mouse doing on a chip?

Last month's *Elektor World* topic on deep IC explorations using an acid bath drew a nice response from Peter van de Wetering in Holland. His hobby is to inspect the architecture of old chips designs. At some point Peter had revealed the insides of a Mostek type MK5017AA IC — a clock chip capable of driving 7-segment displays. Peter remembers: "a very nice design, but not terribly accurate, and also quite sensitive to all sorts of AC powerline distortions". When his brother was busy welding in the shed outside, the IC suddenly shortened 3 days to about 3 minutes!

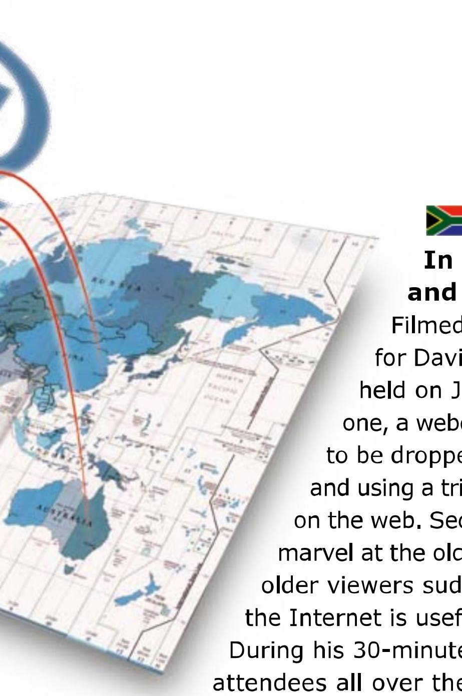


When inspecting the 4x4 mm design of the actual silicon chip, Peter noticed a small spot. Put under a magnifier glass he discovered a developer's joke; Mickey Mouse stretching his arms between '7' and '12'!

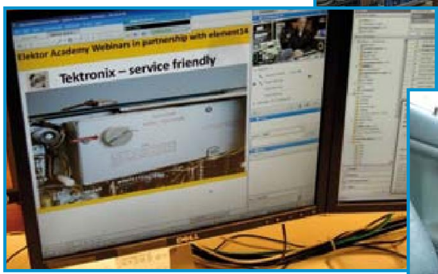
Searching for more information on Mostek and why Mickey Mouse was etched in the design we found out the company got sued by Disney lawyers due to a publication in Electronics Magazine. Elmer Guritz, former Mostek employee, says;

'We used to put all sorts of designs on chips, i.e. Woodstock on an HP calculator chip, a rat on a Magnavox Chip and of course Mickey on that clock chip that some foolish person published in Electronics Magazine...'

Mostek was purchased in 1979 by United Technologies Corporation, which ended up purchased by Thomson France, so the story is fading, but perhaps there are people out there wanting to help us find this rat on a Magnavox chip …. have a close look!



In front of the camera, behind it, and behind the wheel



Filmed at Elektor HQ in Holland in a studio not unfit for David Letterman, the *Retronics — Best Of* webinar held on January 24 was remarkable in a few ways. For one, a webcast like this allows antique electronic equipment to be dropped in the App Age at the flick of a few switches and using a trifling amount of bits on the web. Second, young people marvel at the old gear shown, while older viewers suddenly confirm that the Internet is useful.

During his 30-minute talk to registered attendees all over the globe (check the picture of the notebook in de passenger seat, it was taken in South Africa by Brian Tristam Williams on his way home from work), Elektor's resident e-vintage & boat anchor expert Jan Buiting showcased some of his favourite items discussed since 2004 in his '*Retronics*' column — you know, those nostalgic pages towards the back of the magazine. Jan's 80+ instalments are now in a book, too! The webinar was canned and may be viewed at your best time: start at element14.com/webinars.
For the next webinar on February 21, sign up at www.elektor.com/webinar

CC25

Over a year ago during a lunchtime conversation in San Jose, CA, *Circuit Cellar* staffers began preparing the *Circuit Cellar 25th Anniversary Issue*. Today, the team is happy to announce that the issue is ready for the world to read (http://circuitcellar.com/25th-anniversary/home/).
Circuit Cellar magazine was founded in 1988 by Steve Ciarcia, who had previously authored the "Ciarcia's Circuit Cellar" column in *BYTE* magazine. Today, Circuit Cellar is an Elektor International Media publication supplying articles, tutorials, and design projects covering professional electrical engineering and state of the art embedded systems design.
The purpose of the Circuit Cellar Anniversary Issue is to commemorate 25 years of *Circuit Cellar* magazine, as well as document the history of embedded technology since the late 1980s.
The issue is divided into three main sections:

- *The Past*: articles about the last 25 years of Circuit Cellar magazine (memorable design projects from past issues), as well as essays and interviews about early embedded technologies.
- *The Present*: essays on present-day electrical engineering topics (essential embedded design principles, user interface design, embedded security, and more).
- *The Future*: prognostications about the future of embedded electrical engineering, embedded technology, and the microcontroller industry.

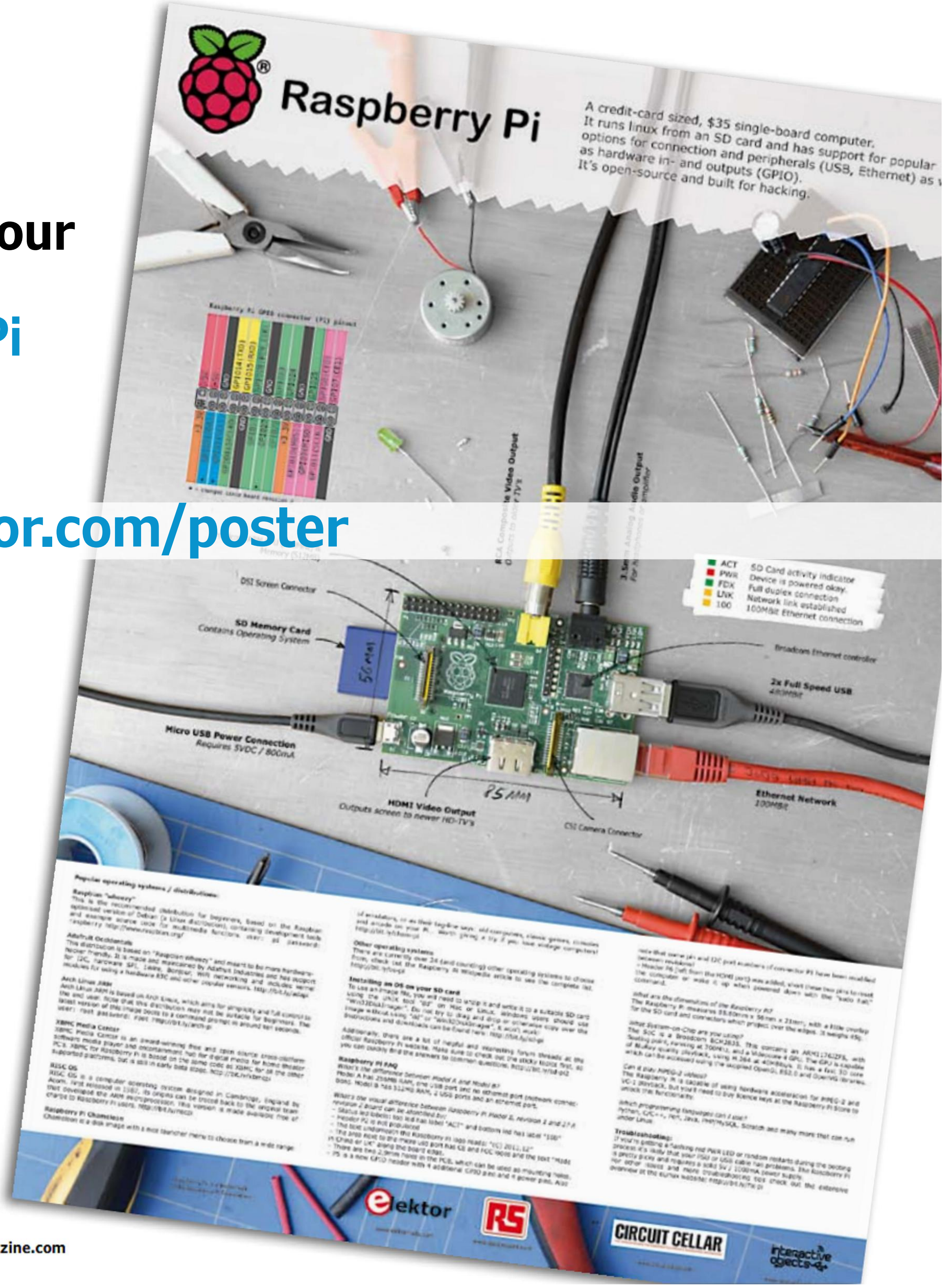
The issue features the following articles, essays, and interviews:

- Steve Ciarcia (Founder, *Circuit Cellar*) on the magazine's history;
- Dave Tweed (Engineer/Editor, *Circuit Cellar*) on 25 years of embedded design projects;

- John Regehr (Professor, University of Utah) on the future of small RAM microcontrollers;
- Limor Fried (Founder, Adafruit Industries) on the future of the DIY revolution;
- Simon Ford (Director of Online Tools, ARM) on the future of rapid prototyping;
- and many more.

There are also interviews on the future of technology with Steve Sanghi (CEO, Microchip Technology), Stefan Skarin (CEO, IAR Systems), and Jeff Kodosky (Co-Founder, National Instruments).
Go to **www.cc-webshop.com/CC25-Anniversary-Issue-FI-2013-CC25.htm**
for more information about the unique issue.

Download your free Raspberry Pi poster at www.elektor.com/poster



500 ppm* LCR Meter (1)

The luxury of precision within everyone's reach

By
Jean-Jacques Aubry
(France)

The remarkable precision of this device and its amazing ease of use are the result of careful design. It works so well behind its uncluttered front panel that one could almost forget the subtleties of the measurement techniques employed. A dream opportunity for our readers who are passionate about measurement to enjoy themselves. If, like us, you wonder at the marvels modern techniques bring within our reach, come along and feel the tiny fraction of a volt.

Apologies

It is impossible to tell the full story on this test instrument in one go, so we'll just have to accept right away the idea of breaking it up into manageable chunks. You will only have seen the whole thing once the series of two or three instalments have been published — please excuse this inconvenience.

* *see detailed specifications*

Technical specifications			
Display	Dominant parameter: value Secondary parameter: value Equivalent circuit: series or parallel (manual or automatic selection) *Q*-*D* (can be reversed with respect to the automatic choice) |*Z*| Φ either *R*s + *X*s, *V*x + *I*x, or ADCU + ADCI		
Sort function	On the main parameter of a control component (*R*, *L*, or *C*) after adjustment of the central value. Tolerances: 1 % 2 % 5 % 10 % 20 %		
Measurement scope	**Parameter**		**Value**
	L		0.1 nH – 100 H
	C		0.1 pF – 100 mF
	R, |*Z*|		0.1 mΩ – 1,000 MΩ (1GΩ)
	Q or *D*		0 – 10,000
	Φ		−90.00° to +90.00°
	R_s, X_s		0.1 mΩ – 1,000 MΩ (1 GΩ)
	U_x and I_x	U_x	0 – 500 mV
		I_x	0 – 5 mA
	ADC *U* and ADC *I*		0 – 5 V
Test frequencies	50 Hz supply		100 Hz, 1 kHz, 10 kHz
	60 Hz supply		120 Hz, 1 kHz, 10 kHz
Power consumption	Without display		5 V – 100 mA
	With backlit display		5 V – 180 mA
PC software	For Windows, Linux, MacOSX		

Test conditions	
Off-load test voltage	0.4 V_{rms} ± 5 %
Ranges	8, automatic
Measuring speed	Around 2 measurements per second. It is possible to take the average of several measurements (1–9), at the expense of speed. A green LED indicates the end of each sequence.

Accuracy of the main parameter (*R*, *L*, *C*)	
Conditions*	Warmed up for 10 minutes, 25°C ± 2°C Use of 0.01 % resistors (100 Ω, 1 kΩ, 10 kΩ, 100 kΩ) in the current/voltage converter.
Ranges 3, 4, 5, and 6	< ±500 ppm (0.05 %) (down to 200 ppm **) ±1 on the final digit
Ranges 2 and 7	< ± 1000 ppm (0.1 %) (down to 800 ppm) **) ±1 on the final digit
Ranges 1 and 8	< ± 3000 ppm (0.3 %) ±1 on the final digit
** The accuracy (±1 unit in the last figure on the right) is greatest when the post-amplification ranges are the same for *U* and *I*.	

Miscellaneous	
Measurement connections	4 Kelvin wires on BNC connectors
Compensations	"OPEN" / "SHORT"
"SHORT" limit	$\|Z_s\| < 10\ \Omega$
"OPEN" limit	$\|Z_s\| > 100\ k\Omega$
Supply voltage	5 V_{DC} ± 5 %, via the USB connector

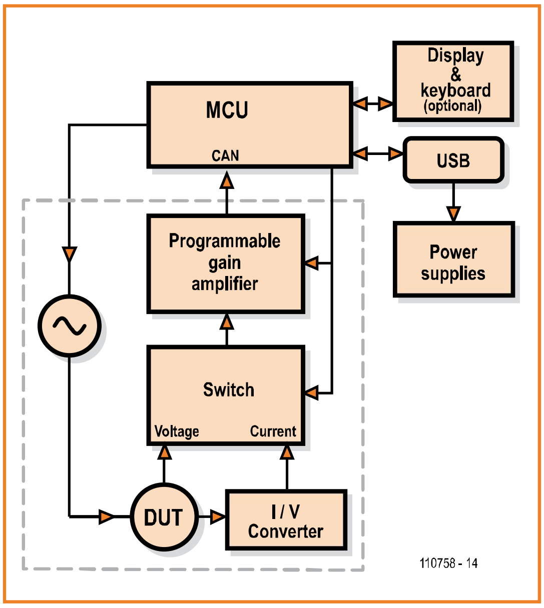


Figure 1.
Analogue and digital techniques are closely intertwined within the sub-assemblies of the 500 ppm LCR Meter.

An LCR meter isn't usually seen as indispensable in an amateur electronic enthusiast's workshop. However, with the proliferation of SMD components with no markings, like chip capacitors, or the increasingly common use of chokes in switch-mode PSUs, the use of an LCR meter is becoming more and more frequent. Let's not forget too that this device doesn't stop at just giving a passive component's 'value' in the usual sense — inductance *L*, capacitance *C*, or resistance *R* — but also its secondary component, defining its 'quality' (see specifications box), which can be defined in several ways:

- Φ – phase angle between voltage and current: tan Φ = $|X_s| / R_s$
- *Q* – quality factor = tan Φ, used to characterize an inductor.
- *D* – dissipation factor = 1/*Q*, used to characterize a capacitor.

This duplicity in our components, innocent enough as to be most usually ignored in low-frequency circuits, does need to be taken into account in high-frequency work, and more generally in precision circuits.

We have to go back more than 15 years in the Elektor archives to find a precision LCR meter [1]. My project, the fruit of which you're going to be discovering here, has itself been in gestation for four years; the first version, battery-operated, with a modest 2×16 LC display, has become a bench model, mains powered, with a 128×64 graphic display. My exchanges with Elektor Labs have led to the version published here, which benefits from all the accumulated experience. Did we have to choose between two concepts, either a computer used as a display and control peripheral (USB link) associated with an LCR meter that was reduced to being just a measuring interface, or a true self-contained LCR meter? No. To keep everyone happy, I'm proposing a no-compromise variable configuration, designed to combine accuracy and convenience at the highest level:

- A measuring head that dialogues with a PC (for displaying the results and sending commands), but is also present on an extension connector;
- An optional extension comprising a display and a mini keyboard, which when connected to the previous element converts it into a **standalone** unit;

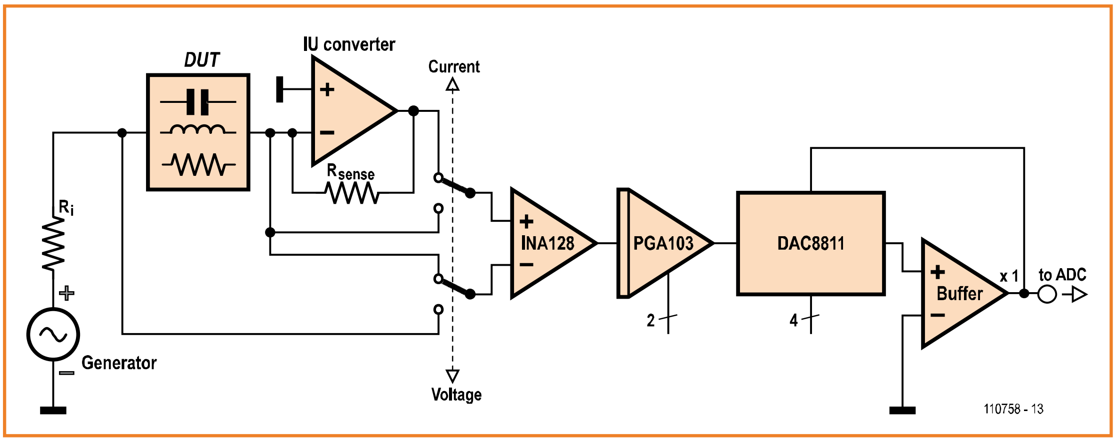


Figure 2.
An enlargement of the section within the dotted line in Figure 1: this is the measuring head, which most of this article is about.

A little bit of theory

The complex impedance to be measured is equal to the ratio between the vectorial dimensions $\overrightarrow{U_X}$ and $\overrightarrow{I_X}$, representing the voltage across the device under test (DUT) and the current flowing through it:

$$\overline{Z_X} = \frac{\overline{U_X}}{\overline{I_X}}$$

Each vector can be broken down into *phase* and *quadrature* components with respect to some fixed reference:

$$Z_X = \frac{V_p + jV_q}{I_p + jI_q}$$

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

$$R_S = \frac{V_p\,I_p + V_q\,I_q}{I_{p^2} + I_{q^2}} \qquad X_S = \frac{V_q\,I_p - V_p\,I_q}{I_{p^2} + I_{q^2}}$$

Some LCR meters go down the analogue route (phase detectors) to obtain the phase and quadrature components of the voltage and current to be measured, the final measurement being performed by an ADC, often of the dual-ramp type for good accuracy, as the DC voltages to be measured are in fact 'contaminated' by a not-inconsiderable residual voltage, if one is looking for a fast response time. The "all digital" method does not suffer from this drawback, and the mathematical operation of discrete Fourier transformation (DFT) makes it possible to obtain the *phase* and *quadrature* values for the voltage (U_p U_q) and current (I_p I_q) from *N* samples d_i of one period of the signal to be measured

$$U_p = \frac{1}{N}\sum_{i=0}^{N-1} d_i \times \cos\left(\frac{2\pi i}{N}\right) \qquad U_q = \frac{1}{N}\sum_{i=0}^{N-1} d_i \times \sin\left(\frac{2\pi i}{N}\right)$$

This requires just a fast, accurate ADC, and a little bit of calculating power.

- Software, within the measuring head, capable of handling both these modes.

Before we take a detailed look at the complete circuit of the unit, I propose first taking a quick look at the block diagram, which gives a good idea of the whole project (**Figure 1**). It would be tempting to immediately examine each element in detail as we usually do. Sadly, we wouldn't understand a great deal of it without first taking a look at the problems posed by these measurements. All in all, they are fairly complex, and the precision requirement we've set for ourselves places the bar pretty high. Rest assured, though, that all I'm using for this is grey matter — there are no other hard-to-find components.

Principle and operation

In the "A little bit of theory" box, I provide a few details about the measurement principle. Let's take a look here at its practical implementation in the measuring head (**Figure 2**), which comprises those elements enclosed by the dotted line in Figure 1. The analogue section of the LCR meter uses the conventional technique of a self-balancing bridge to determine the unknown impedance of the device under test (DUT) by measuring the voltage across its terminals together with the current flowing through it when driven by a variable-frequency sinewave signal.

We can see from Figure 2 that the current flowing through the DUT also flows through the current sensing resistor R_{sense} (this simplified diagram doesn't show that the value of this sensing resistor changes with the impedance range, which we'll see in the full circuit diagram). The potential at the inverting input of the current/voltage converter (*IU converter*) is maintained at 0 V (virtual ground) so as to maintain the balance between the current through R_{sense} and that through the DUT. As the frequency is never more than 10 kHz, a fast opamp is suitable for this converter, which must introduce only a minimum of phase error into the signal path.

The complex impedance of our unknown component will thus be obtained from the voltage measured across the DUT and that across R_{sense} (an image of the current through the DUT), applied to a differential amplifier (INA128) via the I/U switch. Before being digitized by the microcontroller, the signal undergoes further amplification (PGA103), multiplication (by means of the DAC8811 fast converter), and filtering. The remainder of its path is handled by the software, which will first determine the basic series parameters of the unknown impedance R_s and X_s (where X_s is an inductive or capacitive component, depending on the nature of the DUT) and then lastly the

Table 1. Combined gain: variable from 1 to 866 in 48 steps															
PGA gain	1					10					100				
CNA gain	1	1.155	..	7.50	8.66	1	1.155	..	7.50	8.66	1	1.155	..	7.50	8.66
Total gain	***1***	***1.155***	***..***	***7.50***	***8.66***	***10***	***11.55***	***..***	***75.0***	***86.6***	***100***	***115.5***	***..***	***750***	***866***

Table 2. Measuring Ranges

Range	**IU converter resistor**	**U gain**	**I gain**	**Measuring Range (resistance)**	**Measuring Range (L or C - impedance)**
1	100 Ω	100	1	< 0.1 Ω	< 1 Ω
2	100 Ω	10	1	0.10 Ω to 11 Ω	1 Ω to 10 Ω
3	100 Ω	1	1	11 Ω to 900 Ω	10 Ω to 995 Ω
4	1 kΩ	1	1	900 Ω to 9.9 kΩ	996 Ω to 10 kΩ
5	10 kΩ	1	1	9.9 kΩ to 99.9 kΩ	10 kΩ to 100 kΩ
6	100 kΩ	1	1	99.9 kΩ to 1 MΩ	100 kΩ to 1 MΩ
7	100 kΩ	1	10	1 MΩ to 10 MΩ	1 MΩ to 10 MΩ
8	100 kΩ	1	100	> 10 MΩ	> 10 MΩ

other parameters derived by calculation: *Z*, *L*, *C*, *R*, *Φ*, *Q*, *D*.
The conformity of the successive stages is determining. In order to avoid problems of drift, the same chain is used for both the voltage and current measurements after the *IU converter*. The high precision of the programmable gain amplifiers used and the compensation for the stray differential phase error, which is a function of the gain of the chain, ensure a basic accuracy practically equal to that of the precision resistors used in the current/voltage converter. You could be forgiven if you now rushed ahead to look at the circuit, without reading what follows. However, do be aware that in order to understand everything, you will have to come back to the next two essential paragraphs, which may at first reading seem a bit hard going…

Better watch that gain

To obtain extended measuring ranges (see paragraph "Measuring ranges" below), the amplitude of the signals to be measured must be adjusted *before* they are digitized. This is achieved by:

- Selecting the appropriate value of R_{sense}, according to the impedance of the DUT. The values adopted are: 100 Ω, 1 kΩ, 10 kΩ, and 100 kΩ (we'll see these later on the circuit diagram).
- Modifying the gain of the measuring chain so as to drive the analogue/digital converter (ADC) with a voltage that's as high as possible, without overloading it. Using the same amplifying chain for the current and voltage measurements makes it possible to avoid a good part of the drifts and uncertainties about the global gain value. In point of fact, as the impedance value can be written:

$$Z_x = \frac{V_p + jV_q}{I_p + jI_q} \times \frac{Gi\ R_{sense}}{Gv}$$

Where
G_i and G_v are the current and voltage gains of the amplifier channel;
R_{sense} is the *IU converter* resistor;
V and *I* are the voltages measured by the ADC;
the ratio G_i/G_v will only keep the variable parts of the gains.

The linearity error of a successive approximation ADC (SAR) is at best ±1–2 LSBs. Now since the zero-crossing of the sinewave signal to be measured is the region where the digitizing error is greatest, the higher the amplitude of the signal, the more accurate will be the measurement. Most LCR meters, like for example the one described in *Elektor* in 1997 [1], use a programmable gain amplifier (PGA in Figure 1) using a step of ×10 between values (gain of 1, 10, or 100). It is quite simple to use a high-precision integrated PGA, such as the PGA103 from Texas Instruments.

Compensating the phase shifts in the *IU converter* and PGA103

The software measures reference capacitors

The phase shifts introduced by the *IU converter* are compensated for **by the software**. The phase shift to be compensated is determined by measuring components whose behaviour is known, such as SMD capacitors with NP0 dielectric, considered as 'perfect' (at these low frequencies). The advantage of this method: the stray capacitance of the connections is in parallel to the capacitance proper and hence is no longer of any importance.
We will be aiming for a −90° phase shift.
To do this under the best conditions, the amplification gains must be identical for *U* and *I*; this implies that @ 10 kHz the impedance of this DUT must be practically equal to R_{sense}: 159 nF for 100 Ω, 15.9 nF for 1 kΩ, etc.
To make things simpler, these capacitors are wired onto the PCB, but they are only brought into service manually via a jumper during the compensation procedure. After that, they are not used for anything at all!

The software measures low-value resistors

In order to compensate, via the **software**, the PGA103's phase shift for ranges 1 and 2 (not for range 7 and 8, as this would require high impedance with a great many stray components), we use low-value resistors (the influence of stray capacitance and/or series inductance is negligible), here too wired as closely as possible to the board, and aiming for a zero phase shift. These are only brought into use during the compensation procedure and are not used for anything else afterwards!

In this respect, adjusting the gain in this way means the voltage to be measured will change by steps in a ratio of 10. If this voltage had to be used "as-is" by the ADC, not only would the operating conditions not be ideal, but above all the measurements of the *current* and *voltage* by the ADC could be dissimilar, to the point of introducing a (severe) relative digitizing error.
To avoid this, we need to have intermediate gain values — without compromising the accuracy of the overall gain value!
Using a multiplying digital/analogue converter (DAC) (*DAC8811* in Figure 1) in conjunction with an operational amplifier (*Buffer* in Figure 1) allows us to obtain a gain that can be varied from zero to 1 (actually −1, but the phase is irrelevant here), in as many steps as the resolution of this DAC allows, and with the same precision as it!
An ingenious bit of wiring makes it possible to make the gain of this stage variable between 0 and *k* (*k* constant >1): it involves applying only a proportion of the output signal to the DAC8811's built-in feedback resistor.
If 16 gain values are used for the multiplying DAC, the basic gain variations will be selected in a ratio of $\sqrt[16]{10}$, i.e. 1.155, with the maximum gain being 1.155^{15}, i.e. 8.66. This will allow us to have a combined gain that is variable from 1 to 866 in 48 steps (see table 1).
If the gain control program is well designed, it will provide the ADC with a voltage to digitize whose maximum amplitude is close to its full-scale capacity for both the voltage and current parameters of the DUT, with a maximum amplitude difference in a ratio of 1.15.
Under these conditions, knowing that the impedance is equal to the quotient (give or take a coefficient) of the two measurements, and if the ADC has high resolution (ideally 16 bits), then digitizing errors will be practically eliminated. And there you have it as far as the gains are concerned.

Stray phase shifts

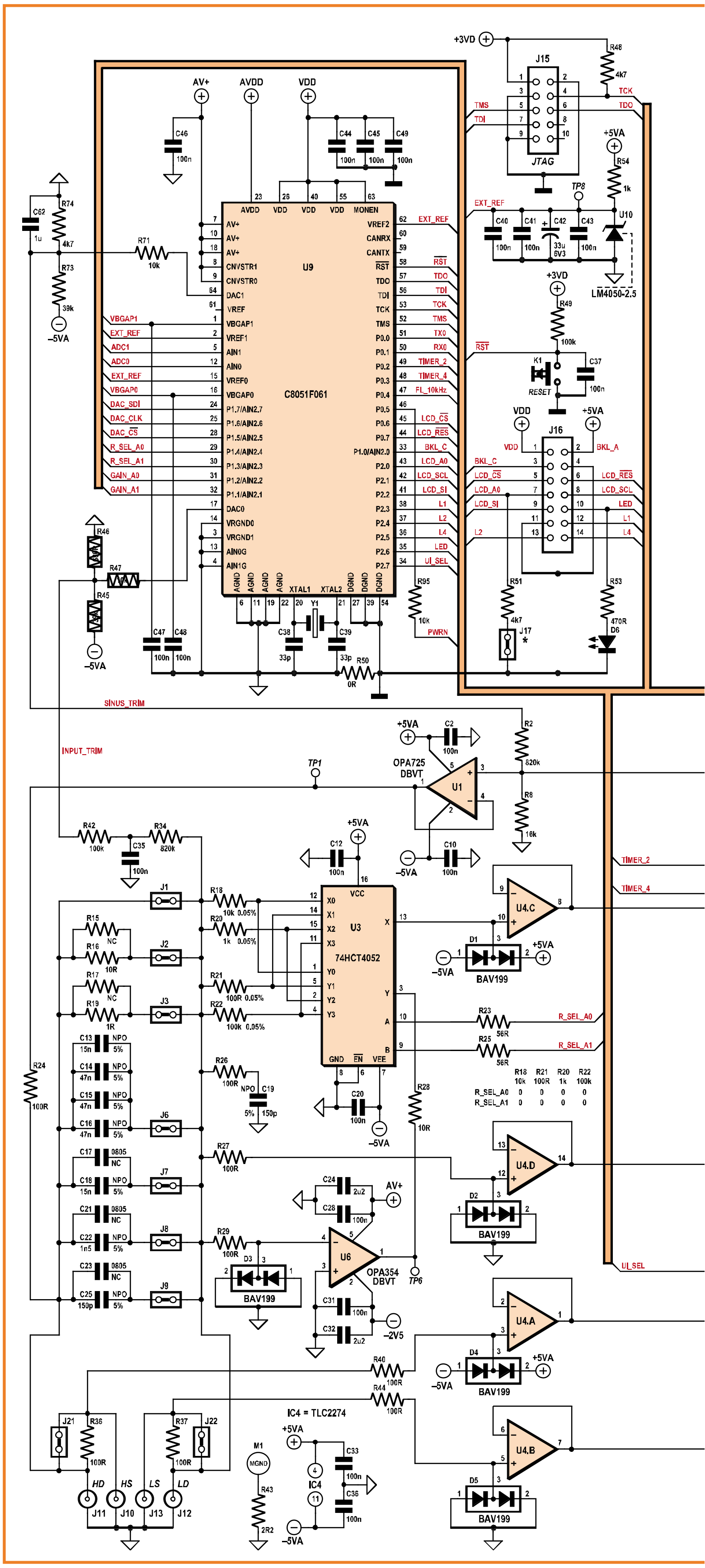
The accuracy also depends on the compensation for the stray phase shifts introduced by the measuring chain.
Two elements must be considered:

- The phase shifts introduced by the *IU converter*, as this is found only in the current measurement path. To achieve this, we're going to be using measurements of multilayer ceramic SMD capacitors with lossless dielectric (NP0 or C0G) — the compensation will be adjusted to obtain a phase shift as close as possible to the theoretical −90°. The appropriate capacitors are fitted to the PCB and brought into circuit by jumpers J6–J9.
- The differing phase shifts in the PGA103 when it is programmed for a gain of 10 or 100 in one of the measuring paths and 1 in the other: ranges 1, 2, 7, and 8 (see table below). In this case, we'll be using two SMD

resistors of 1 Ω and 10 Ω (R19 and R16), brought into circuit by J3 and J2, to achieve zero phase shift.

Measuring ranges (see table 2)
For the *voltage* and *current* measurements of the DUT impedance, eight ranges are defined from the value of R_{sense} (i.e. the *IU converter* resistor) and the gain of the main amplifier (PGA103).
The final amplification (DAC8811 multiplying DAC) between 1 and 8.66 is performed in 16 steps, denoted 0 to F, for both voltage and current measurements.

Analysing the circuit
Before looking at the circuit proper, I just want to point out that digitizing the sinewave signal directly by the microcontroller's ADC obliges us to use DC coupling from one end of the amplification chain to the other. Without such DC coupling, we'd have to wait for the ADC input to stabilize at the mean value of the signal every time the gain was changed or when switching between the *current* (*I*) and *voltage* (*V*) measurements. As the intended accuracy is ≤ 500 ppm (0.05 %), this stabilization time would be a problem. Considering the DC coupling, any offsets superimposed on the sinusoidal signal will need to be compensated so that the mean value of the signal is close to 0.000 V, even at maximum gain.
The main circuit diagram (**Figure 3**) breaks down into four sections, some of which we are already familiar with: at top left, the microcontroller; bottom left, the measuring bridge; bottom right, the amplification chain with the sinewave generator above it; and lastly, top right, the power supply and the interface with the outside world.



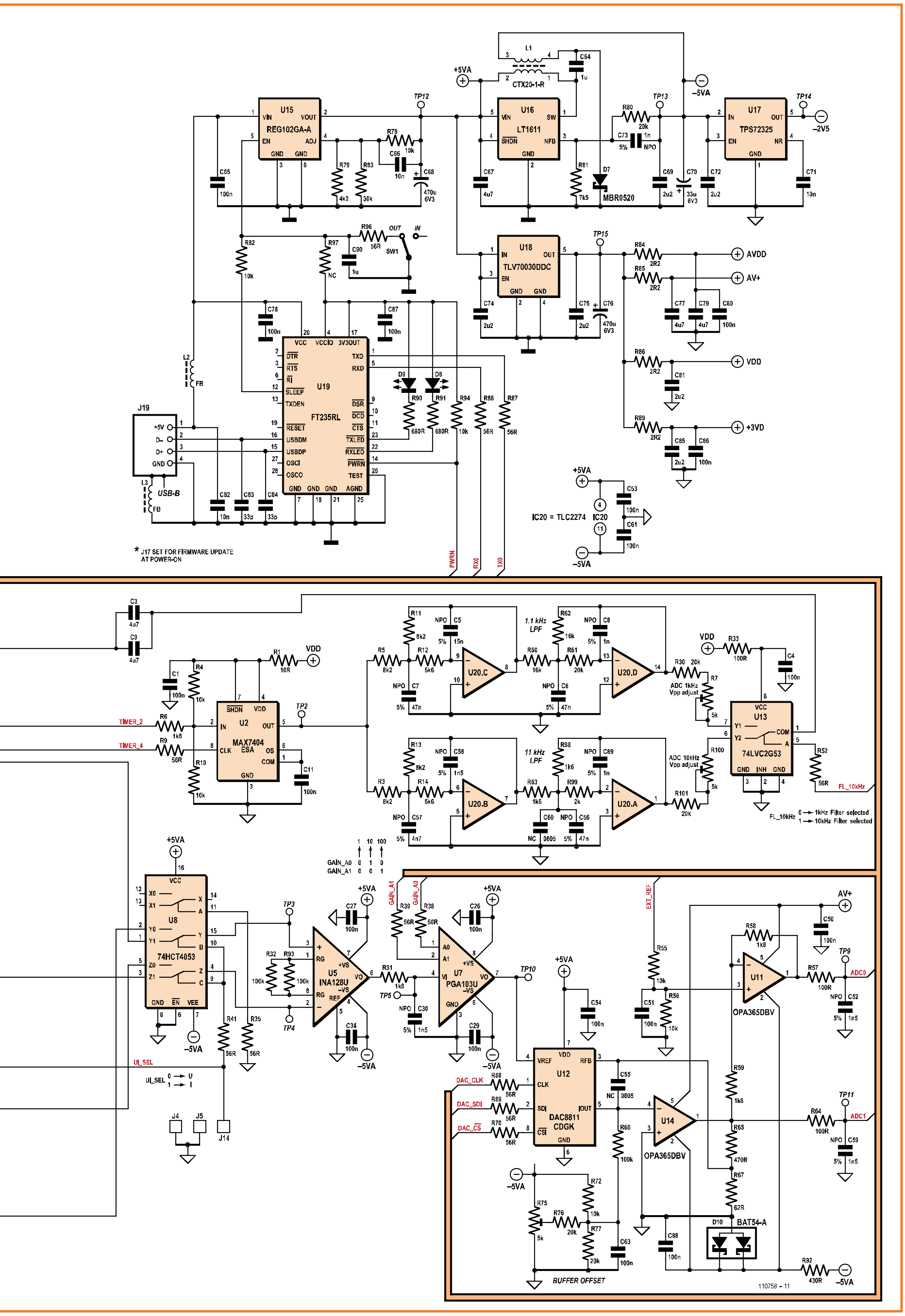


Figure 3.
Complete circuit diagram of the 500 ppm LCR Meter. Almost everything happens inside the microcontroller, but the development of the analogue part needed the greatest care in order to avoid compromising its accuracy in any way.

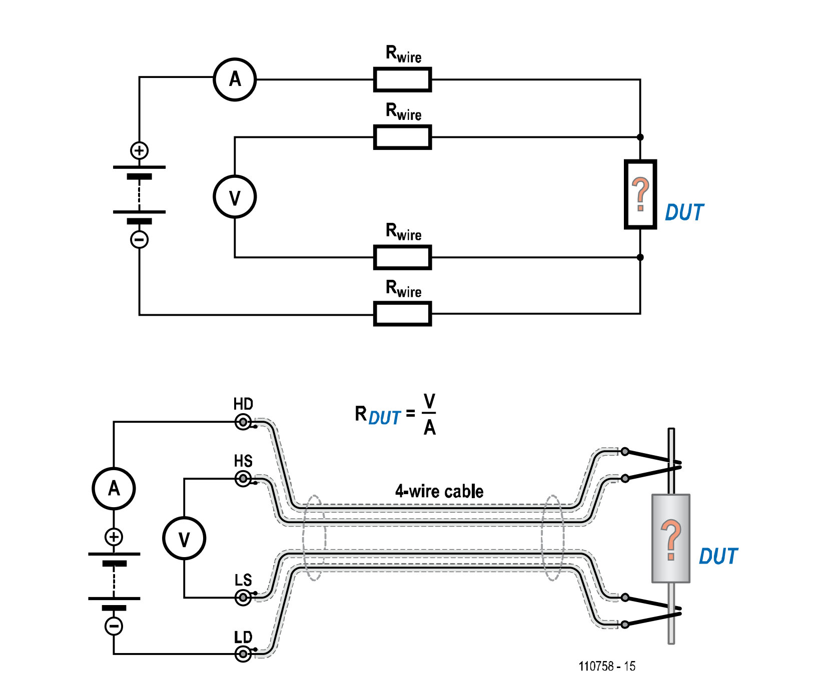


Figure 4.
The Kelvin measuring principle uses two separate wires for current flow and voltage measurement.

Measuring bridge

Having read the foregoing, you will have understood that the *IU converter* (U6, R18, R20, R21, R22) at the bottom left of the circuit diagram is a critical part of the device. The accuracy of the values of R18 and R20-R21-R22 determines the accuracy of the whole device.

The resistor R_{sense} is selected using one half of analogue switch IC U3. The buffer amplifiers U4C and D measure, at high impedance, the voltage across the R_{sense} resistor selected:

- U4D for the voltage at the junction of the resistors;
- And U4C, via the second half of U3, at the other end of these resistors (allows us to compensate for the R_{on} resistance of the switches in U3).

Hence the differential voltage between the outputs of U4C and D is equal to the voltage across the feedback resistor selected via U3.

The voltage across the DUT is measured using a 4-wire Kelvin connection: the DUT is connected between J11 and J12, i.e. the lines High Drive and Low Drive through which flow the current, while the voltage is measured, at high impedance, via J10 and J13 (the High Sense and Low Sense lines) and buffer amplifiers U4A and B.

The internal resistance of a single cable would introduce an error in the measurement result. The Kelvin clip, with its twin insulated cord, lets us perform the voltage measurement without its being interfered with by the voltage drops caused by the current flowing in the other wires (see **Figure 4**).

The dual diodes D1–D5, with very low reverse leakage current, provide overload protection.

All this is of course prone to stray capacitance, which is going to introduce phase shifts. It's essential to choose a very wide bandwidth amplifier for the *IU converter*, so as to maintain the linear relationship between phase shift and frequency. Knowing this phase shift at the maximum operating frequency (10 kHz) will enable us to calculate it at the intermediate frequencies.

The very wide bandwidth of U6 means we have to include a network to stabilize the gain at very high frequencies; this is the role of R26 and C19.

Its offset is compensated using DAC0 in the µC U9: the DAC0 output voltage is programmable between 0 and 2.5 V; this is taken to a value between about −75 mV and +75 mV via the resistor network R47, R46, and R45, which will let us inject a programmable current (via R42 and R34) into U6's inverting input so as to cancel out its output voltage.

Amplification chain

Up front comes the analogue switcher U8 (**Figure 2**, bottom left on right-hand page) which makes it possible to select the output voltages of either U4A and B for the *voltage* measurement (*U*), or U4C and D for the *current* measurement (*I*). After this, the amplification chain is common, thus avoiding the influence of offsets and stray phase shifts; but this does mean we must use amplifiers with stable, well-defined gain.

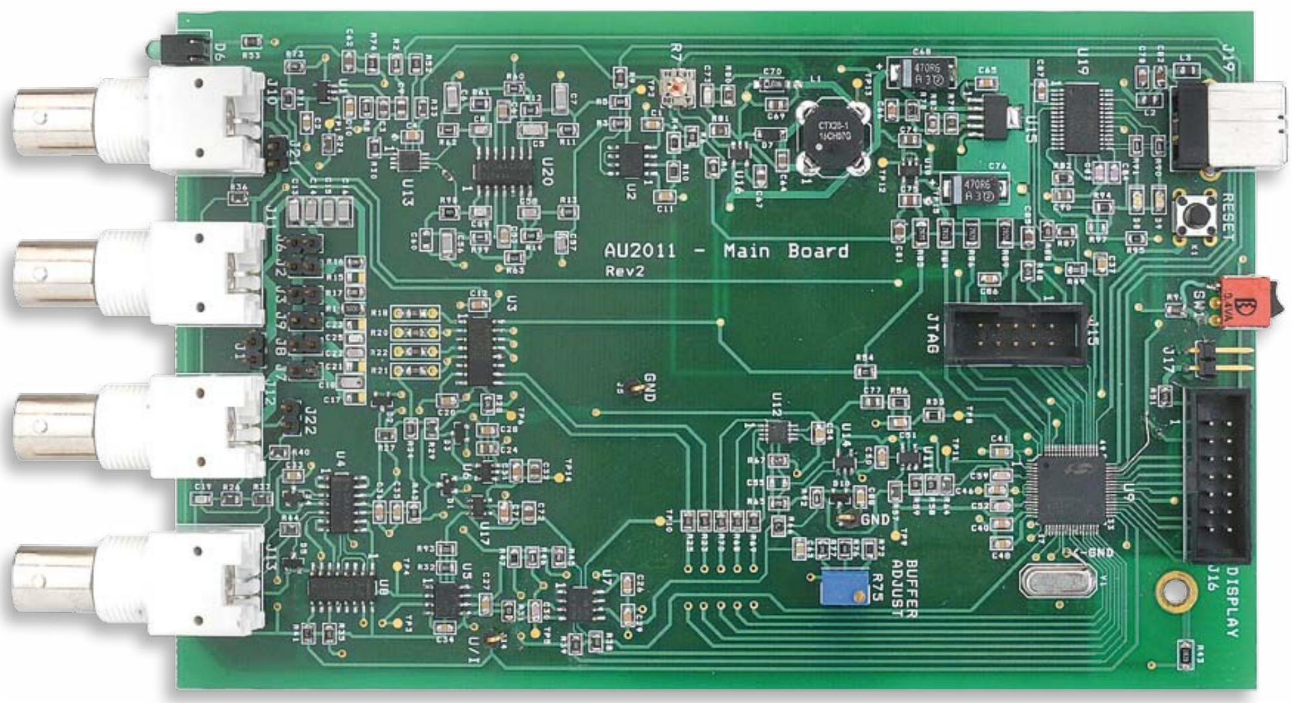
> *"To check it would be possible to use the C8051F06x, I bought the development kit (C8051F060-DK) and obtained a sample of the MAX7400 (8th-order filter). I wired up the latter on prototyping board. I wanted to measure a capacitor. But how could I do this simply, using just the kit's two analogue inputs (referenced to ground)?*
> *The solution was to use two series R_{sense} + C branches, as identical as possible, connected in parallel between the filter output and ground. One with the capacitor on the ground side, the other with the resistor on the ground side. Measuring at the junction of R_{sense} and C gave the C voltage measurement on one branch and on the other the measurement of the R_{sense} voltage – and hence, of the current flowing through C!*

Automatic bridge for measuring the impedance of passive components between 1 mΩ and 1000 MΩ

Differential amplifier U5 lets us go from a floating value to a value referenced to the 0 V power rail (ground). Its gain is set at 2 by R32||R93, i.e. 50 kΩ.
The gain of the precision amplifier U7 can be selected to be 1, 10, or 100 via two control lines. Then come U12, the DAC8811 multiplying digital/analogue converter amplifier, and its fast opamp U14. Taking the feedback voltage (RFB) from the divider R65/R67 rather than directly from U14 output allows us to obtain the quoted gain of 8.66. It is programmed in serial mode using 3 command lines.
In order to make the most of their performance, the ADCs in the µC U9 are used in differential mode. To this end, amplifier U11 inverts the phase of U14's output signal.
As each of the ADCx inputs can only accept a voltage between 0 and V_{REF} (2.5 V), a DC current is extracted from the summing point so as to obtain an offset voltage of ($+V_{ref}$ / 2) and thus a differential DC voltage of 0.000 V (adjustable via preset R75, at the very bottom on the right-hand side of the circuit).

Sinewave generator

The DUT is driven by a sinewave signal that can be set between 100 Hz and 10 kHz. I use three frequencies: 100 Hz or 120 Hz (i.e. twice the AC supply frequency), 1 kHz, and 10 kHz. Other frequencies are possible (with certain restrictions). To get the most out of the microcontroller's ADCs, the digitizing process is carefully synchronized with the sinewave signal. Timer2 in the µC supplies a squarewave signal at the desired frequency, applied to the (8th order) switched-capacitor filter U2. This filter requires a clock signal (CLK) at 100× its cut-off frequency, which is provided by Timer4.
At U2's output, we have a perfectly sinusoidal signal, but we do still need to remove residual clock frequency components from it. This is the job of the two active 4th-order filters built around U20, one with a cut-off frequency of 1.1 kHz,



Operates in standalone mode with display and mini-keyboard or in USB satellite mode with a computer (OSX, Windows, Linux)

used for the frequencies ≤ 1 kHz, and the other with a cut-off frequency of 11 kHz, for the frequencies > 1 kHz. The amplitude of the 1 and 10 kHz signals is adjusted using R7 and R100. Analogue switch U13 selects which filter is used. AC coupling (C3, C9) to buffer U1 eliminates the signal's DC component.
Offset compensation for U1 is performed using DAC1 in the µC U9, in a similar way to the offset compensation for U6.

Powering

The device's supply arrives via J19 (USB-B), with +5 V on pin 1 (V_{bus}) and return on pin 4. It will come either from a USB cable connected to a computer, or from a USB power supply such as a smartphone (max. 6 V_{pp} off load).
Our device draws **more than 100 mA**. So if it is powered via a computer USB interface, this will have to be a 'high power' one, capable of supplying up to 500 mA and **guaranteeing a minimum V_{bus} voltage of 4.75 V**. It is possible to indicate the nature of our peripheral (high-power bus-powered device) by modifying the Max Bus Power parameter of the USB_Config_Descriptors in the FT245R's EEPROM, to set it to 500 mA (see the documentation for the FT_Prog utility on the FTDI website [2]).
Low-dropout linear regulator U15 supplies a voltage of +4.60 V (typ.) when its input 5 (EN) is high. Its 470 mA current limit ensures we respect the USB standard specification for a bus-powered high-power USB device.

Table 3.

$\overline{PWRN}$ State	LCD_$\overline{RES}$ State	Action
low	high	display on computer
low	low	display on computer or standalone mode if any key of the extension's mini-keyboard is pressed during boot.
high	high	ERROR! neither a computer nor the display extension is connected
high	low	standalone mode

The switching regulator U16 supplies −4.60 V (typ.). It is wired in accordance with the LT1611 data sheet [3], using a double-wound inductor L1. Regulator U18 supplies +3 V, and lastly U17, −2.5 V.
Note: On the circuit diagram, the indications +5 V and −5 V actually correspond to the +4.60 V and −4.60 V rails.

Microcontroller

The C8051F061 from Silicon Labs™ is an 8-bit analogue/digital mixed-architecture microcontroller. I chose it for the quality of its 16-bit successive-approximation ADCs ADC0 and ADC1:

- Sampling up to 1 MS/s and direct memory access (DMA);
- Max. ± 1 LSB inherent linearity in different mode;
- ± 0.5 LSB differential linearity (guaranteed monotonicity).

The rest is just as good:

- Improved-architecture 8051 core (70 % of instructions are executed in just 1 or 2 clock cycles)
- Clock frequency up to 25 MHz
- 2 12-bit DACs
- 1 10-bit, 200 kS/s ADC and 8-channel multiplexer
- 4,352 bytes of RAM and 64 KB reprogrammable flash memory
- 5 16-bit timers
- 2 serial ports (UART); data rate up to 115,200 baud (24 MHz clock)
- SMBus and SPI interfaces
- CAN 2.0 bus
- 24 general-purpose inputs/outputs
- Numerous interrupt sources
- JTAG interface, etc.

The test frequencies of 100 Hz, 120 Hz, 1 kHz, and 10 kHz are derived from the crystal-controlled 24 MHz clock.
The 2.50 V reference voltage for the A/D and D/A converters is provided by U10 (to right of the µC). Its accuracy has no effect on the overall

accuracy of the device, as its value is eliminated in the calculations.
When fitted, jumper J17 notifies the boot program of an unconditional firmware update request. LED D6 lights at the end of each valid measurement to indicate that everything's OK.

USB interface
Communication with the PC is entrusted to the well-known FT245R USB/UART converter from FTDI (U19). It's configured for 115,200 baud, 8 bits, no parity, and 1 stop bit (8 N 1) and no handshaking. LEDs D8 and D9 light when data is being transferred.
Between 25 ms and 200 ms after applying power, this IC sends its SLEEP/ pin high. If SW1 is open, U15 is then validated and all the regulated voltages are supplied to the rest of the electronics. If the device is connected to a computer via a USB cable, the PWRN/ pin, which is high when power is applied, goes low after around 300 ms. The µC's internal program tests the status of this pin during the boot sequence, along with the status of pin 6 (LCD_RES/) on extension connector J16; if this pin is low, it indicates the presence of the Display/Keyboard extension. From these two statuses, we get four possibilities (see table 3).

Temporary conclusions
This is where our initial overview of the 500 ppm LCR Meter ends. In the next part, we'll be discussing the measurement accuracy and the error or uncertainty factors (gain, calibration, phase). We'll learn about the extension with its display and keyboard, which will turn it into a standalone device, together with the computer software for the Windows, Mac, or Linux platforms), which is going to enable us to make the most of its accuracy. But you've got plenty to think about while you're waiting. And if here or there you still find some things a bit unclear, don't despair — read through it all again, check out the documentation, and things will soon become clearer.

(110758)

Links and References

[1] **Elektor** April 1997 p. 12; May 1997, p. 12; June 1997, p. 32
All three articles can be found on the Elektor 90-99 DVD, www.elektor.com/dvd90-99.

[2] **FT245R User Guide from FTDI**
www.ftdichip.com/Support/Documents/AppNotes/AN_124_User_Guide_For_FT_PROG.pdf
[or]
http://goo.gl/USPOS

[3] **LT1611 data sheet**
http://cds.linear.com/docs/Datasheet/1611f.pdf

[4] www.elektor-magazine.com/110758

The fruits of passion

Born in 1943, Jean-Jacques Aubry, drawn at an early age to the phenomenon called Radio, developed an interest in listening-in to radio amateurs, using tube receivers. Graduating as a radio-electrical engineer in 1968, he joined a small electronics firm, where he stayed for 36 years.
He discovered computing with the ZX80 kit from Sinclair (commandeered from his young son) and taught himself programming, first with a stock management program in Turbo Pascal, then the software for various test benches in Visual BASIC. Moving on to the Mac in 1990, he applied himself to C and then C++, and discovered Qt with a Software Defined Radio (SDR).
Once retired, to keep busy, Jean-Jacques set out to design a high-performance LCR meter.

I initially noted a discontinuity in the measured value when changing range (modifying R_{sense} or the PGA103 gain).
In the end, I realized that one of the reasons was that I'd used capacitor with X7R dielectric for C30 (filter between U5 and U7), C52, and C59 (ADC0 and ADC1 input capacitors). With the signal's alternating voltage applied, the non-linearity of this dielectric introduced noticeable distortions! The solution was to use capacitors with NP0 (C0G) dielectric.

During my initial testing, I couldn't get the right frequencies for the timers. In the C8051F06x documentation, for the square-wave output mode, it gives (p. 298, equation 24.1):équation 24.1)

$$F_{sq} = \frac{F_{TCLK}}{2 \cdot (65535 - RCAP_n)}$$

RCAPn being the value to be loaded into a Timer n register.
In actual fact, you need to perform the calculation using 65,536 and not 65,535!

Part 2 The 7-uP Alarm Clock/ Time-Switch

By
Michael J. Bauer
(Australia)

The main board schematic shown in **Figure 3** gives the impression of a very simple industrial controller with minimal external I/O interface connections. A notable exception is the audio generator circuitry, which is more complex than you would expect on a general-purpose programmable controller, for example.

The choice of micro-controller chip has already been explained in the technical introduction (above). The At89c5131 is a derivative of the Intel 8051 family and thus has many 8051 core characteristics, including the inexplicably bizarre

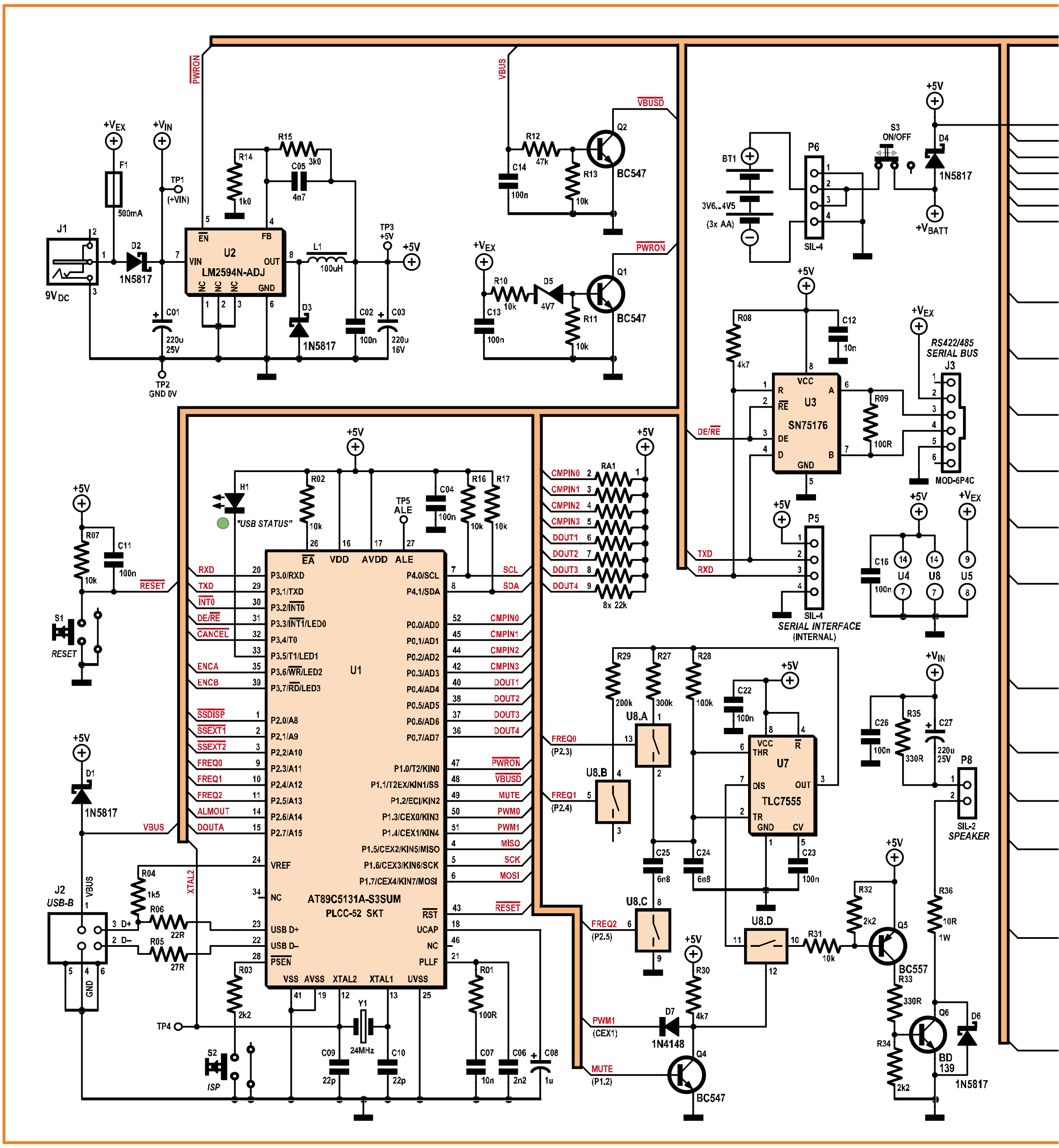
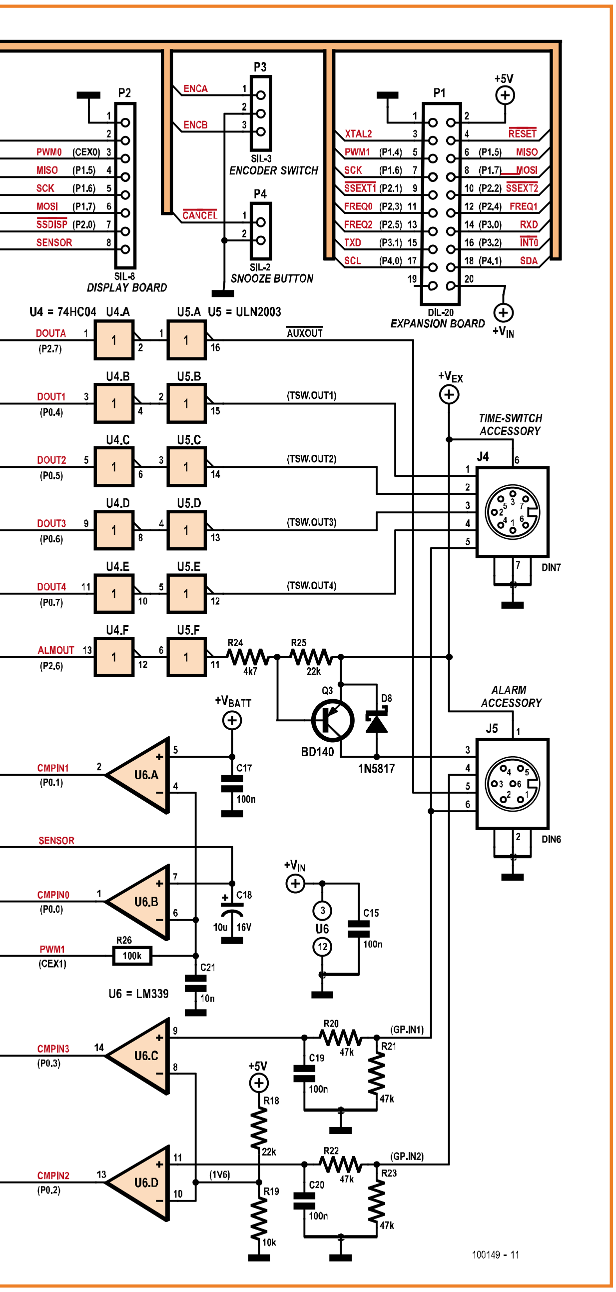


Figure 3.
Circuit diagram of the main board electronics.

Description of the main board, the display board, and assembly

I/O port configurations. Port 0, for example, is open-drain and therefore requires external pull-ups on pins to function as outputs. Other ports have internal weak pull-ups (~25 kΩ) which do not provide sufficient output drive capacity for most purposes.



Nevertheless, the 89c5131 has many redeeming features including a very fast processor core. With a 24 MHz crystal and 'X2' clocking mode, the MCU core runs at 48MHz! The device incorporates many extra peripheral modules, e.g. USB, Timer T2, PCA (used for PWM outputs), SPI, and TWI (IIC). The At89c5131 provides a separate flash program memory block (4 KB) for a bootloader. The device ships with a USB bootloader pre-programmed into this 'boot block', allowing user program code to be loaded into the main flash program area (32 KB) via the USB port, i.e. requiring no additional flash programming hardware. If the processor is reset while the PSEN# pin is pulled low (using the ISP button), the bootloader is started instead of the user application code.

The UART TX and RX signals are routed to the expansion header (P1), also to a 4-pin header (P5), in case the user wants to attach an RS232 serial interface adapter. The UART signals are also routed to an RS422/485 transceiver (U3 = SN75176) to support external accessories via a serial bus. The initial release of clock firmware does not include any feature which uses the RS422/485 serial bus, so the transceiver (U3) and bus port connector (J3) may be omitted until such time as might be needed.

Circuitry is provided to monitor the external DC supply voltage (+V_{EX}). If it drops below about 6 V, transistor Q1 will turn off and the logic signal PWRON# will be in the High state due to a pull-up resistor in the MCU. The firmware monitors this signal and turns on an enunciator (PF) if the supply drops below minimum required voltage. The signal PWRON# is also routed to the 5V switch-mode regulator IC (U2 = LM2594) EN# (enable) pin. If the external 9V power supply is removed, the 5 V regulator is shut down, allowing the +5 V rail to be energized either by USB power or battery power.

The USB 'V_{bus}' line is sensed by the circuit of transistor Q2. If USB power is not present, Q2 turns off and signal VBUSD# will float High. The firmware reads this signal and adapts its operating mode accordingly. The state of the USB V_{bus} signal can also be used by the firmware to control one (or more) of the accessory control outputs.

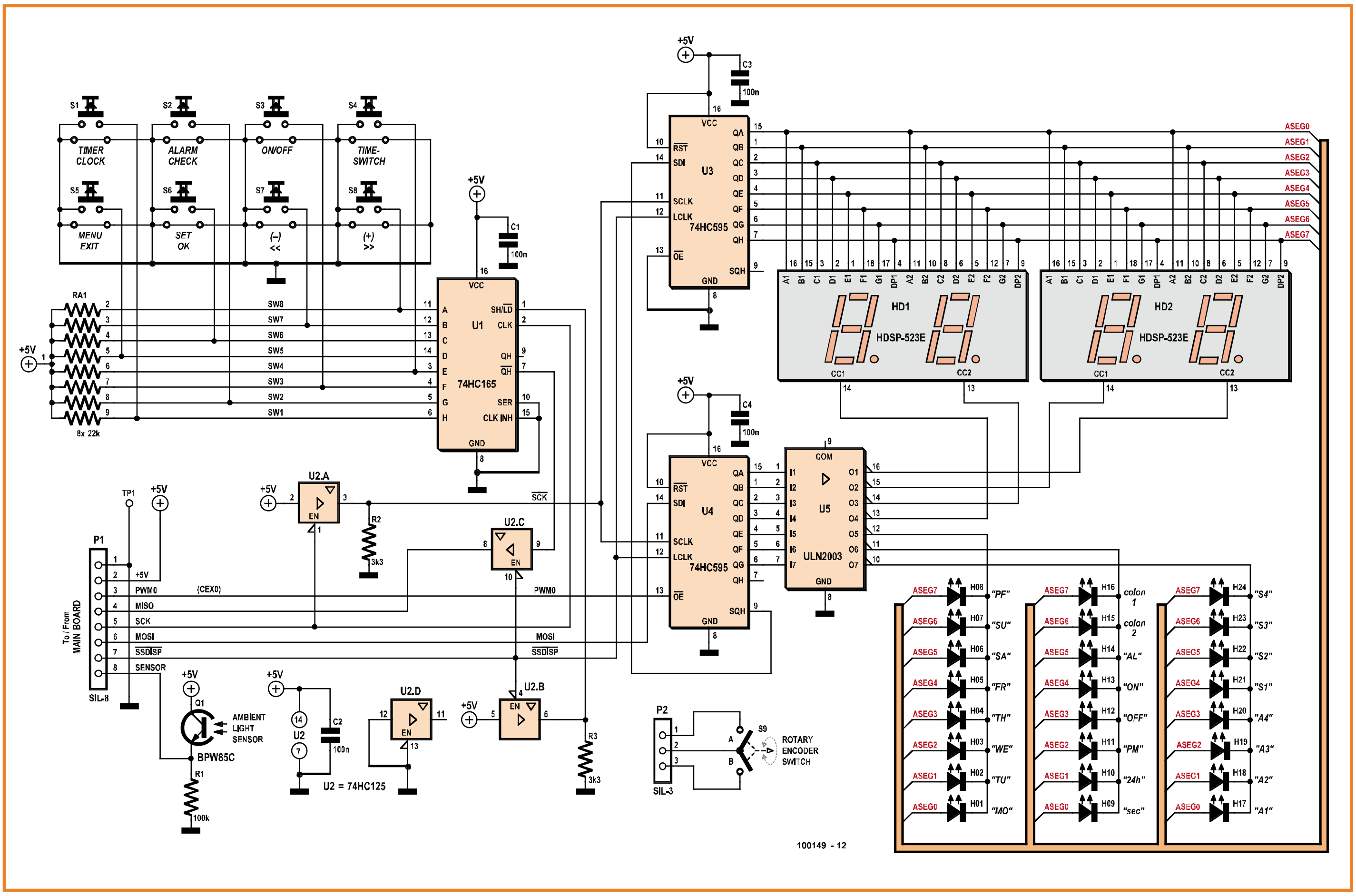


Figure 4. Circuit diagram of the display board.

Accessory logic inputs and outputs

Six logic outputs are provided on the accessory sockets J4 and J5. The four time-switch outputs are active low (current sinking) control signals which can sink up to 50 mA each. These are intended to drive optocouplers (isolated inputs) in a solid-state AC power board. The alarm-switched 9V DC output is active high (current sourcing) and can source up to 500mA (limited by the resettable fuse F1). The corresponding MCU output port pins are buffered with a CMOS hex-inverter (U4) because the port pins cannot source enough current to drive the inputs of the peripheral driver IC (U5 = ULN2003A).

Two logic inputs are implemented by comparators U6C and U6D. The logic threshold is set to +1.6 V. The external inputs are buffered and filtered by RC networks to protect the MCU input port pins from nasty external transients (e.g. ESD).

Analogue inputs

The AT89c5131 MCU does not provide analogue (ADC) inputs. The clock application needs two analogue inputs to monitor the ambient light sensor and backup battery voltage. Analogue-to-digital conversion is implemented by a firmware algorithm, making use of a pulse-width modulation (PWM) output from the MCU to provide a variable reference voltage. A low-pass filter (R26, C21) removes the AC component (47 kHz) from the PWM pulse train, so that the DC level is controlled by the PWM duty. The PWM reference voltage is fed into the (–) inputs of two comparators U6A & U6B. The analogue input voltages to be measured are fed into the (+) inputs of U6A & U6B. By changing the reference voltage at periodic intervals, the firmware can determine the input voltages. Conversion accuracy is limited by the resolution (and linearity and noise) of the PWM-generated reference voltage. About 1% error is the best we could hope for. Due to the filter time-constant, it takes a few milliseconds for the reference voltage to stabilize after a change in PWM duty. Consequently, the conversion time is quite slow compared to a hardware ADC. Fortunately, in this application, speed is not an issue.

Audio generator and power amplifier

A square-wave oscillator is realized with a CMOS timer IC (U7 = TLC555). The frequency of the timer audio output signal is controlled by switching the resistance and capacitance of the RC timing network. Eight different frequencies in the range 500Hz to 2kHz (approx.) are selectable by MCU logic output signals FREQ2, FREQ1 and FREQ0. A CMOS quad analogue switch (U8 = CD4066B) is used to switch the resistors (R27, R29) and capacitor (C25) in or out of circuit. The chosen frequencies are harmonically related.
Audio amplitude is controlled by a PWM signal generated by the MCU (CEX1), fed to the control input of analogue switch (U8D) which 'chops' the square-wave audio signal. The duty cycle of the PWM signal determines the effective audio output level. The PWM frequency is about 47 kHz; well out of range of human hearing and out of range of the speaker too. (The speaker itself acts as a filter to remove the 47 kHz PWM signal.) The PWM pulse duty is variable over a 256:1 range using the 8-bit duty register in the MCU. This is an adequate dynamic range for the application, but a 10-bit (or higher) resolution would have been really nice.
The audio signal remains a square-wave (chopped at 47 kHz), right up to the speaker driver transistor Q6. Hence, the power amplifier is a class-D design. Transistor Q5 is necessary to boost the base drive current required to turn on Q6. The PWM analogue switch (U8D) is not capable of passing currents higher than about 2 mA. Using a class-D power amplifier not only provides higher audio output power level than a linear amplifier, but it has another major advantage in this application. When the audio generator is "muted" (by firmware putting the MUTE signal High), the PWM switch (U8D) is off and hence the output transistor Q6 is off, so there can be no current whatsoever flowing in the speaker, no matter how much noise is present on the supply rail ($+V_{IN}$). This is a very important consideration for an appliance which may be sitting closer than 3 feet to your ears while you are trying to sleep.
The audio level is updated by the firmware 'sound synthesizer' driver routine every two milliseconds. The firmware driver is capable of synthesizing an amplitude 'envelope' for the alarm sounds. User-settable parameters determine the attack (slope), sustain (time), release (slope) and hold-off (time), in similar manner to the analogue sound synthesizers of the 1970s (for which the author has a fond nostalgia). The hold-off time determines the interval of silence between successive plays of an alarm sound. (This is not the same as the 'snooze interval', which is independently settable.)
In addition, the firmware provides a means to modulate the audio amplitude and/or frequency with (virtual) modulating signals, the period of which can be set to any integer multiple of two milliseconds (i.e. 2, 4, 6, 8 ... ms), up to 500 ms.
In summary, a simple low-cost audio generator circuit in combination with a smart firmware driver routine delivers a diverse range of appealing sound effects (and some not so appealing!).

Circuit description – display board

The display uses a 4-digit 7-segment LED display because it is a low-cost solution giving adequate readability, and because the technology lends itself to variable-brightness capability. A white or multi-color backlit graphical LCD panel would have been more aesthetic, but the extra cost was considered unjustified, and continuity of supply of LCD panels can be an issue.
Figure 4 shows that the 4-digit numeric display is realized using two 2-digit devices (Avago HDSP-523X), available in three colours: red, green or yellow. Pin-compatible parts are made by other optoelectronics manufacturers. Backlit enunciators are realized by 24 discrete LEDs, 5 mm round diffuse lens, of various colours.
The 7-segment display and enunciator LEDs are multiplexed in a common cathode 7 x 8 matrix. The cathodes are driven (active low) by a 7-element peripheral driver IC (U5 = ULN2003A), which is just an array of seven Darlington transistors with TTL/CMOS logic compatible inputs. The anode (segment) lines are driven by an 8-bit CMOS shift-register. Looking at the display schematic, astute readers will notice there are no current-limiting resistors in series with the anode lines. The design relies on the source resistance of the CMOS outputs (approx. 60 Ω) to limit the LED current. The ULN2003 outputs have a minimum low-state voltage near 1 V (because of the Darlington configuration). The LEDs' forward voltage drop is about 2 V. That accounts for 3 V, which when subtracted from the 5 V supply, leaves 2 V on the CMOS outputs' equivalent resistance (60 Ω). The LED peak current is therefore limited to about 33 mA (= 2 V/60 Ω).
Keep in mind the LEDs are multiplexed 8:1, and dimmed using PWM, so the peak current is

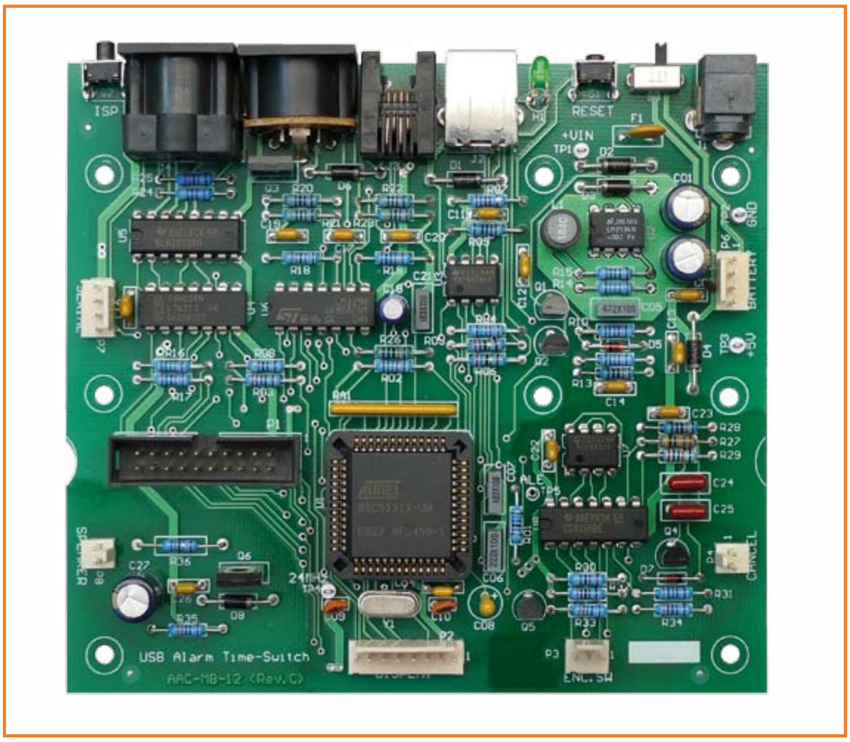
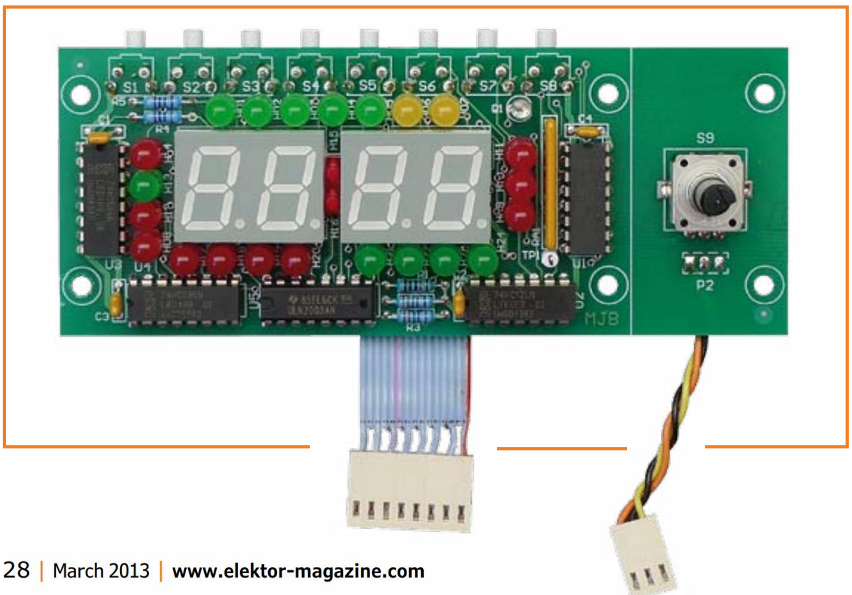


Figure 5.
Top view of
completed main board.

allowed to be near the LEDs' maximum rating. The average LED current will be much lower than this — about 2 mA in normal viewing conditions. Variable brightness is achieved by applying a variable duty 47 kHz pulse waveform (PWM) to the Output Enable (OE#) input of the LED cathode driver register (U4). The range of PWM duty, and hence brightness, is 256:1. At minimum duty (1/256), the display is still bright enough to be readable in low lighting conditions.

By the way, don't be tempted to fit blue or white LEDs for any of the enunciators. Their forward voltage drop is too high (typically about 4 V) to be used in this circuit design.

Figure 6. Top view of
completed display board.



The LED display requires two 8-bit output ports and the push-buttons need an 8-bit input port. MCU I/O port expansion is realized by 8-bit CMOS shift-registers with parallel data latches and serial data transfer logic. The shift-registers are (almost) directly compatible with the MCU 'SPI' (serial peripheral interface) bus. With a tiny sprinkling of "glue" logic, implemented by a quad tri-state buffer (U2 = 74HC125), the display output registers (U3, U4 = 74HC595) and button input register (U1 = 74HC165) connect easily to the SPI bus. Note that buffers U2A and U2B are wired to function as inverters, with the aid of pull-down resistors on their (tri-state) outputs. Inverting the polarity of the SPI clock signal (SCK) to the display registers allows the same SPI clocking mode to be used for simultaneous read (slave input) and write (slave output).

Using the SPI bus minimizes the number of signals required to interface the display board to the main board. The display and button registers share a common SPI 'slave-select' signal (SSDISP#). The display registers are written and the button register is read in the same 2-byte SPI data transfer cycle. Since the button input register is only one byte wide, the second byte of the received data is ignored by the firmware. The slave-select signal SSDISP# is inverted to obtain a shift/load signal SH/LD#. While the input port is de-selected (SSDISP# High), parallel data are accepted by the input latch. During an SPI transfer, SSDISP# is put Low, SH/LD# goes High, so the input latch is inhibited and the serial clock is enabled, i.e. the input data is held constant while being shifted out.

Output registers (74HC595) are comprised internally of an 8-bit serial-in/parallel-out SR with an 8-bit D-type latch. Output pins are driven from the latch bits. Whenever the latch clock signal (LCLK) is pulsed, the rising edge causes the 8 bits in the internal shift register to be transferred to the output latch. While LCLK remains High, the bits in the output latch remain unchanged. Bits in the internal shift register change whenever any SPI transfer cycle (read or write) is executed, regardless of whether the SPI slave select signal is asserted or not. In other words, the shift register data will change when other devices on the SPI bus, if any, are selected and written to. This doesn't matter, so long as LCLK remains static while there is garbage data in the internal shift-register. Connecting LCLK to the display slave-select, SSDISP#, ensures the desired behaviour.

The display board has a phototransistor (Q1 = BPW85C) which senses the ambient light level. The current flowing in the emitter resistor, and hence the voltage across it, increases with increasing illumination of the transistor junction. The emitter voltage is monitored by a comparator on the main board using a primitive analogue-to-digital conversion technique, as noted earlier. Provision is made on the display board for mounting a rotary encoder switch (S9). Project builders who choose to deviate from the 'standard' enclosure arrangement might want to relocate the encoder switch. This is easily done by cutting the display board into two pieces, separating off the smaller piece with the encoder switch. For this reason, wiring terminations to the encoder switch are separate from the other display board connections (8-way pad strip).

Internally, the encoder switch has two sets of contacts which make and break as the shaft is rotated. Using pull-up resistors to the +5 V supply rail, the two switch outputs (A and B) generate square pulse trains in a quadrature phase relationship (i.e. 90 degrees out of phase with each other). Each output produces 24 pulses per revolution of the shaft, but the relative phases of the two pulse outputs is different (by 180 degrees) depending on the direction of rotation. A firmware decoding routine removes noise from the signals and keeps track of changes in the encoder shaft position.

Assembly

The assembly of the clock is described in detail in a document called *Assembly Instructions* which may be downloaded free of charge from the Elektor website [1]. The documentation package also contains BOMs. Due to space limitations, we are limited here to printing photographs of the assembled main board (**Figure 5**), display board (**Figure 6**), the connector line-up at the rear of the clock (**Figure 7**), and how the display board is mounted upright in the casing (**Figure 8**).

You're invited!

The firmware is richly annotated, and could serve as a useful example to Electronics Engineering, IT and Embedded Technologies students developing embedded real-time firmware projects, whether or not they build the 7-uP Alarm Clock.

This project has the potential to spawn a number of spin-off projects, not only as accessories for the alarm-clock, but which could be used independently. Examples: Four outlet solid-state relay (TRIAC) AC power board; Low-voltage dimmable LED bed-side lamp; IR remote-control interface (for TV/AV equipment); PCM/MP3 add-on sound-effects board (with SPI link to MCU). The last three of these examples would require firmware updates for use with the clock.

Also, Elektor readers active at the **ElektorLabs** community website [2] are expressly invited to develop a Windows GUI software application for use with the clock, e.g. for the purpose of editing, saving and restoring alarm and time-switch schedules, option parameters, etc. Windows GUI software developer, please step forward. Note: The clock operates stand-alone — it is not dependent on Windows software.

Please feel free to suggest any changes you think might enhance the appeal or utility of the project. Express yourself at **ElektorLabs** and be heard.

(100149)

Internet Links

[1] www.elektor.com/100149

[2] www.elektor-labs.com

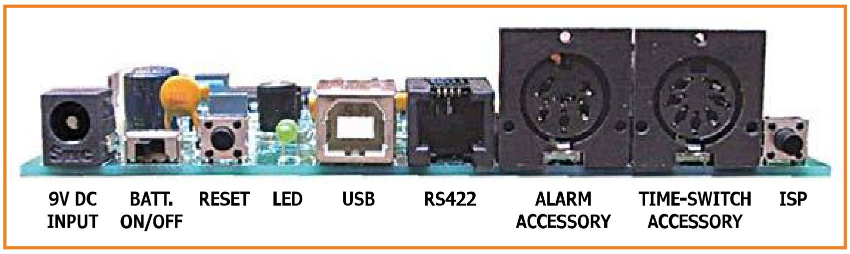


Figure 7.
There's an impressive array of connectors at the back of the 7-uP Alarm Clock.

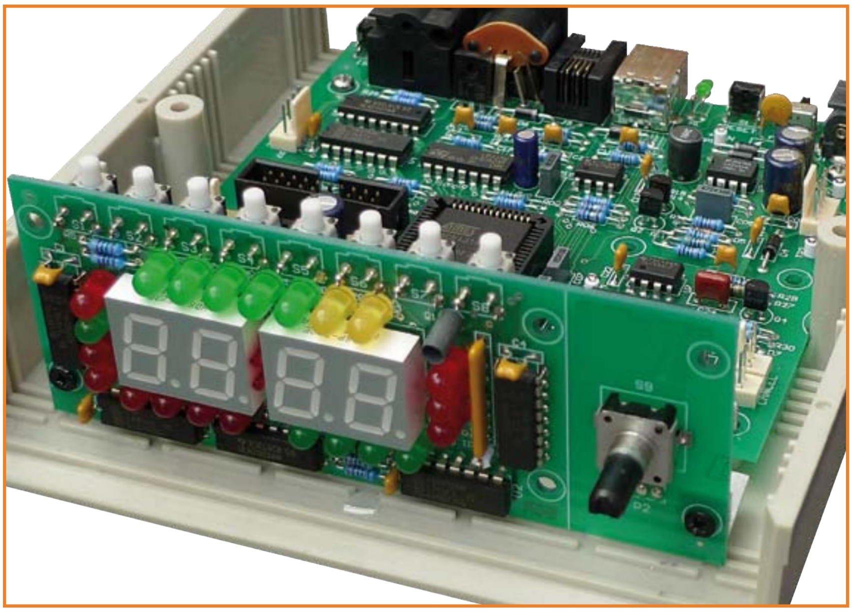


Figure 8.
Showing how the display board is held vertically in the Pactec CM6-225 case.

Raspberry Pi Prototyping Board

Make the RPi do things… Your Thing!

By **Tony Dixon** (UK)

One of the more interesting aspects of the Raspberry Pi, unlike most conventional personal computers, is that it has a small and simple expansion header connector to which a user can directly connect their own interfaces and circuits. To aid in this, this article presents a prototyping board which can be used to make building your own interfaces and circuits for the Raspberry Pi easier.

The Raspberry Pi is a bold and exciting development from the Raspberry Pi Foundation charity based in Cambridge in the UK. The Raspberry Pi Foundation [1] wants to bring tiny and affordable computers to the children of today, with the intention to re-ignite their interest in computer programming and all things technical.

The Foundation have enjoyed enormous interest, support and help from both the educational institutes (such as Cambridge University), electronic companies (RS and Farnell) and the software and maker development communities.

"The Pi" (pronounce /*pai*/, not /*pi*/) has been said to be "the beeb" (BBC Micro) of the Internet age. Elektor was quick to spot not only the RPi's potential, but also one of its makers, Eben Upton, as you can read in an interview [2].

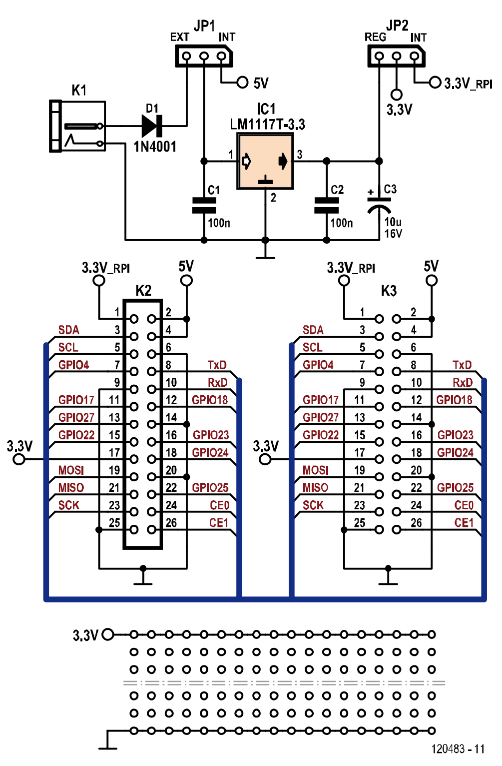


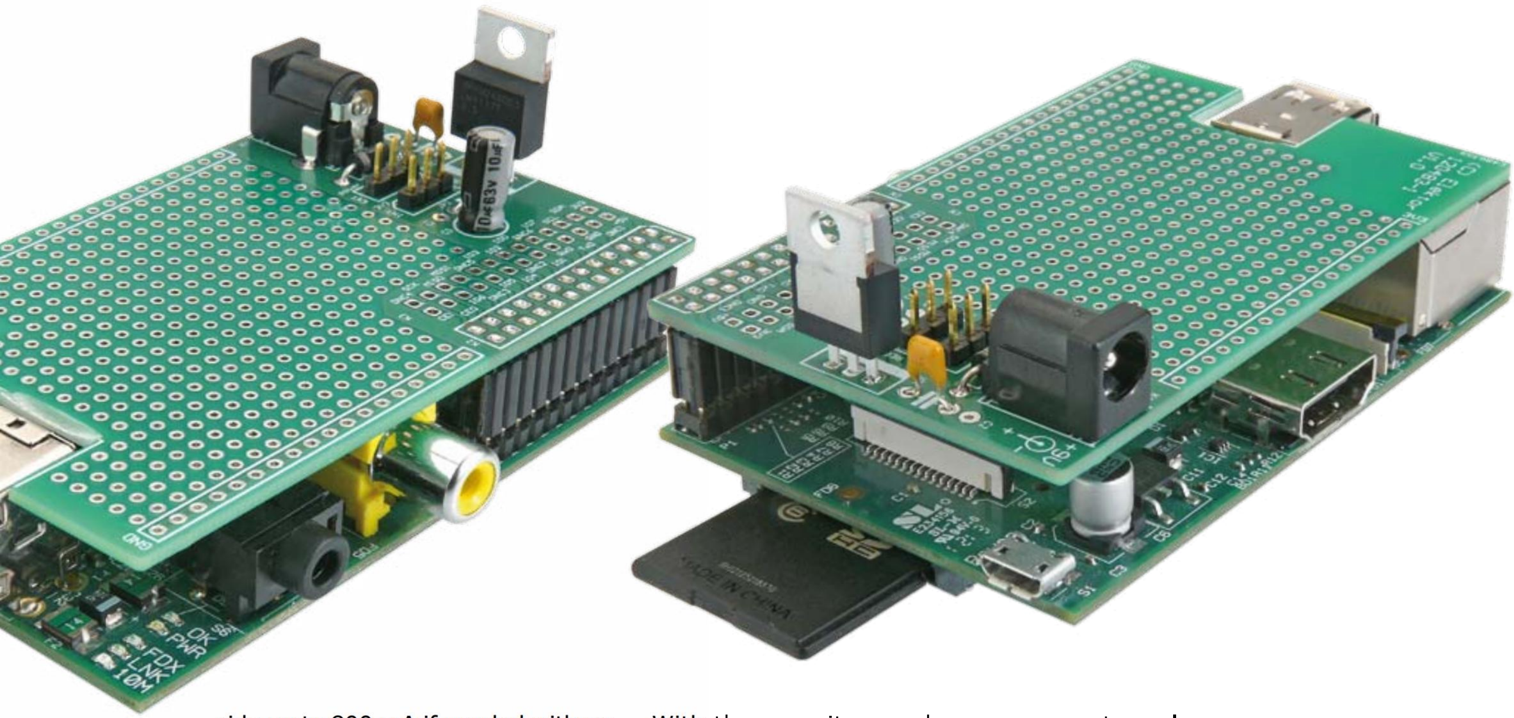
Figure 1.
Not much of a circuit, this, but quite essential for serious prototyping with your Raspberry Pi.

Circuit description

The RPi Prototyping Board is a simple board, designed to break-out the expansion signals from the Raspberry Pi board and provide additional power for any circuit built in the prototyping area. Its pleasantly simple schematic is shown in **Figure 1**.

The first thing the RPi Prototype board provides is an additional 3.3 V DC power supply source. The 3.3 V from the RPi expansion header can only provide a small current at around 50 mA, so if you want to build any circuits which need more than 50 mA, an additional 3.3 V power source will be required.

On the RPi Prototyping Board, this is provided by IC1 which is an LD1117 linear 3.3 V regulator. This can be either powered from an external or internal DC power source, with the selection made on jumper on JP1. The LD1117 can pro-



vide up to 800 mA if needed with an appropriate heatsink secured to it.

If jumper JP1 is set to 'INT' (internal then IC1 is powered from the 5 volts line directly from the Raspberry Pi board.

If JP1 is set to 'EXT' (external), then external power from a 9–12 V DC power adapter ("wall wart") is fed to IC1 through a standard 2.1 mm jack plug (center pin positive, outer sleeve 0 V) through diode D1. Jumper JP2 will need to be set to 'REG' (regulator) position. Capacitors C1 and C2 act as noise suppression devices on IC1, and C3, as a small reservoir.

The second thing the RPi Prototyping Board provides is an easy means to access the signals from the Pi's expansion header connector. A second connector K2, breaks the Raspberry Pi signals out, allowing the user designed (yes, your!) circuit to easily connect to them.

One final note on the circuit, if a small circuit design is being prototyped that doesn't need more than 50 mA from the 3.3 V supply then jumper JP2 allows the 3.3 V prototyping power rail to be connected directly from the Raspberry Pi board itself. To do this set JP2 to 'INT'.

Assembly

Assembly for the RPi Prototyping Board is straightforward. Using the PCB layout pictured in **Figure 2**, solder the small components first, that's D1, C1, C2, JP1 and JP2. The slightly larger components K1 and C3 should be soldered next.

With the capacitors and power connector soldered, solder he regulator IC1 next. Finally solder the connector K2. This needs to be mounted on the solder side of the PCB to allow the RPi Prototyping Board to be mated with a Raspberry

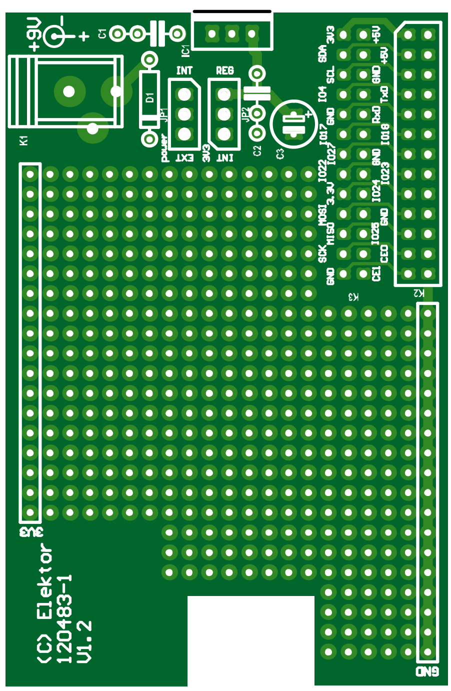


Figure 2.
A prototyping board is like a tennis lawn: just waiting to be played on.

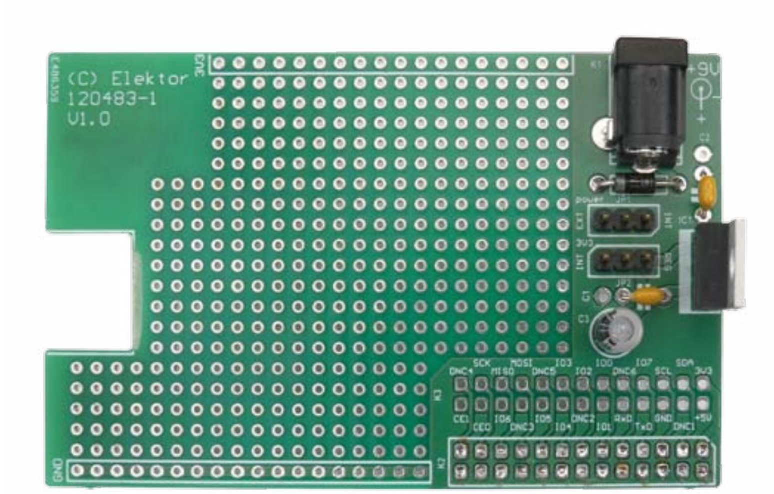


Figure 3.
Impeccable construction courtesy of Luc @ Elektor Labs. Note that the PCB silkscreen pertains to Rpi Revision 2.

Pi. K3 is not a connector but a set of solder pads where you connect the signals you need for your experimental circuit. Feel free to solder pins in the relevant holes.
The choice of component for connector K2 is important. The Raspberry Pi is a very compact board and has a number of raised components, namely the RJ45 Ethernet connector (height = 13 mm), RCA Composite Video connector (height = 13 mm) and Dual USB connectors (height = 17 mm), which have the potential to mechanically interfere with any expansion boards connected to it. Therefore, for the RPi Prototyping board to be mated successfully with a Raspberry Pi, connector K2 should have a total height of not less than 10 mm. The connector specified for K2 has a height of 10.8 mm, this together with the height from the standard 0.1" pinheader fitted to the Raspberry Pi brings the standoff height to 13 mm. This is sufficient to just clear the RJ45 and RCA connector of the Pi. A rectangular clearance is provided on the RPi Prototyping Board to avoid mechanically interfering with the Raspberry Pi's USB connector.
All done with your soldering? Then compare your work with the *proto*-prototyping board pictured in **Figure 3**.

Using the board — generalities

A couple of things to remember about the Raspberry Pi expansion header are, firstly, the signals from the RPi expansion header are 3.3 V only and are **not 5 V compatible**, so be careful what you connect in the way of voltages to them. Secondly, the current the RPi signals can sink or source is small, at around 8 mA, so again be careful what you connect, **don't connect anything that needs a lot of current because it's not going to get it**. If you need to source or sink more, then use a buffer chip such as 74LVC245 to provide additional source/sink current capabilities, or just provide 5 V compatibility.

An RPi LED blinking demo

Whenever Elektor publishes about microcontroller boards, readers appear to be divided in two camps. A typical Camp #1 member says: "I am invariably smarter than you, but I've never published anything myself. No example application for me." From Camp #2 may be heard: "I'm not buying or building anything micro without a fully illustrated application, so yes, please show it to me". To complicate things, both Camp #1 and #2 members may choose to a) not tell anyone, b) tell their peers only, or c) widely disseminate their opinion in forums, tweets, etc. Add variables like language, print or online, age 16 or 66 — and you get an interesting matrix.

The following example is for all of you who want to delve into the basics of controlling stuff with an RPi, be it a few LEDs, as shown in **Figure 4**, or a minibar in a girls-only limousine with GPS tracking, slowly cruising Lower Manhattan. Your imagination is the only restricting factor. Why? Because software is involved.
This project shows a ULN2803 8-channel Darlington Array driving a set of LEDs. Just as it's the tradition for your first program on a new system to display "Hello World", so it's the tradition on any new hardware project to flash an LED to show it all works. So keeping up with these traditions **Listing 1** shows a simple Python program to control the LEDs.

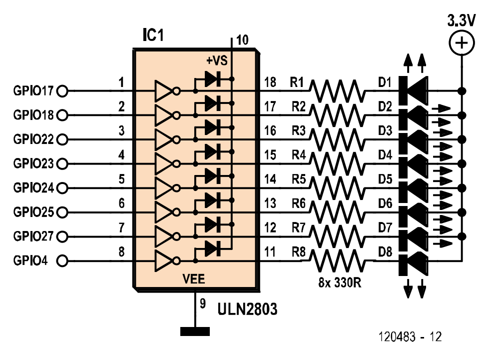


Figure 4.
Circuit diagram of the RPi Blinking LED Demo hardware. All the real power is in the software!

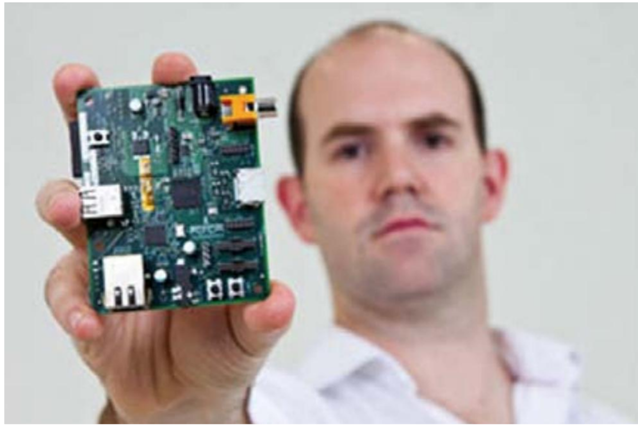
Says Eben Upton, Raspberry Pi co-creator & prophet

Q: Whence the name?

A: We wanted to have a computer especially for Python, and there is a great tradition of naming computers after fruit: like Apricot, Acorn and even today there are computers named after fruit. So Raspberry is following the line of a rich tradition with the Pi, and yes, we wanted this connection with Python. That is where the Pi comes in.

Q: The Raspberry Pi is a bare PC board; no keyboard, no HD, no screen… how will this product become successful?
A: Basically, there is no reason why a computer has to cost more than $50. The peripherals like a screen and keyboard and storage will create a higher price, but with the Raspberry Pi we have taken another route — a normal TV can be used as a screen. Combine that with a 'charity shop' keyboard for a few dollars and you have a full working system. The Raspberry is specifically aiming at youngsters learning to program.

(reproduced from Elektor, April 2012)



This program uses a Python GPIO library to give us access to the GPIO pins. If you've not already downloaded the Python development tools or Python GPIO library, then using an LX Terminal on your Pi, we'll type the following commands. First, however, we'll download the Python development tools by typing

```
sudo apt-get install python-dev
```

In order to access the Raspberry Pi GPIO port we need to download and install the GPIO package. Type the following:

```
wget http://pypi.python.org/packages/
source/R/RPi.GPIO/RPi.GPIO-0.4.1a.tar.
gz
```

Once downloaded, we'll need to extract the files. Type

```
tar -zxf RPi.GPIO-0.4.1a.tar.gz
```

Once extracted a new a directory will be created with the Python files in. Now type

```
cd RPi.GPIO-0.4.1a
```

Now we'll install the package by typing

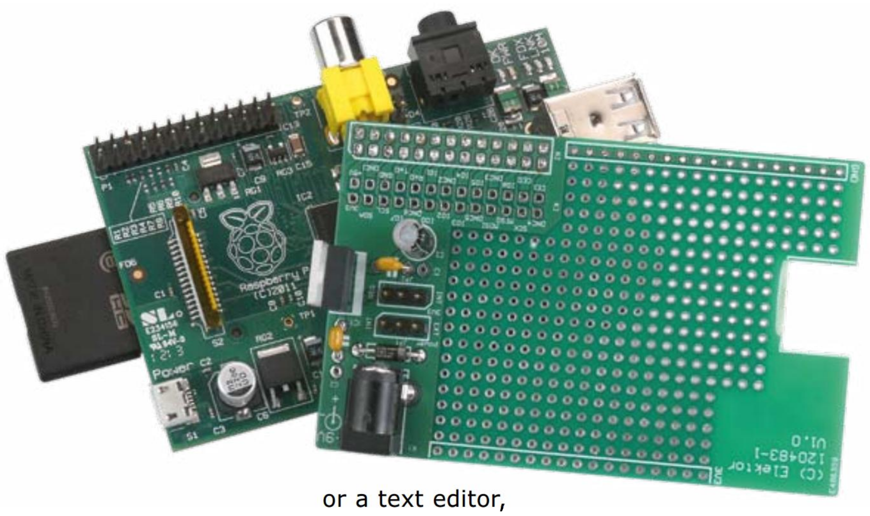
```
sudo python setup.py install
```

Once that's done we should have the Python GPIO library installed. On your Pi, using either IDLE

Listing 1: Blinky.py

```
# !/usr/bin/python
import time
import RPi.GPIO as GPIO

# Configure Pi's GPIO pins
GPIO.setmode(GPIO.BCM)

pins = [17,18,22,23,24,25,21,4]

for pin in pins:
  GPIO.setup(pin,GPIO.OUT)

# Program loop
while True:
  for pin in pins
    GPIO.output(pin, True)
    time.sleep(0.01)
    GPIO.output(pin, False)
    time.sleep(0.01)
```

More RPi circuits and applications shortly in the Elektor.POST newsletter



or a text editor,
type the program as shown in Listing 1. Looking at the program, Setmode defines the numbers used to address individual pins. The instruction `GPIO.setmode(GPIO.BCM)` employs the symbolic names assigned to the RPI's I/O lines. In this case the following eight pins are employed in the array called 'pins': GPIO17, GPIO18, and GPIO22 through GPIO4. **Note that the allocation follows the RPi PCB Revision 1. Here, schematic label GPIO27 is indicated as '21', i.e. the older I/O numbering.**

Next, `GPIO.setmode(GPIO.BOARD)` sets up a direct link between the IO lines and the physical numbering of expansion connector on the Rpi board. Using this setting, the pin array looks like this to achieve the same effect on the LEDs:
`pins = [11,12,15,16,18,22,13,7]`

Once you've typed the program, save it as 'Blinky.py', switch to an LX Terminal and type the following command to make your program an executable:

Raspberry Pi Expansion Header

Table 1. Expansion Header Pin Out

Pin Name	Pin Function	Alternative
P1-02	5.0V	-
P1-04	5.0V	-
P1-06	GND	-
P1-08	GPIO14	UART0_TXD
P1-10	GPIO15	UART0_RXD
P1-12[1]	GPIO18	PWM0
P1-14	GND	-
P1-16	GPIO23	
P1-18	GPIO24	
P1-20	GND	-
P1-22	GPIO25	
P1-24	GPIO8	SPI0_CE0_N
P1-26	GPIO7	SPI0_CE1_N

Pin Name	Board Revision 1		Board Revision 2	
	Pin Function	Alternative	Pin Function	Alternative
P1-01	3.3V	-	3.3V	-
P1-03[2]	GPIO0	I2C0_SDA	**GPIO2**	**I2C1_SDA**
P1-05[2]	GPIO1	I2C0_SCL	**GPIO3**	**I2C1_SCL**
P1-07	GPIO4	GPCLK0	GPIO4	GPCLK0
P1-09	GND	-	GND	-
P1-11	GPIO17	RTS0	GPIO17	RTS0
P1-13	GPIO21		**GPIO27**	
P1-15	GPIO22		GPIO22	
P1-17	3.3V	-	3.3V	-
P1-19	GPIO10	SPI0_MOSI	GPIO10	SPI0_MOSI
P1-21	GPIO9	SPI0_MISO	GPIO9	SPI0_MISO
P1-23	GPIO11	SPI0_SCLK	GPIO11	SPI0_SCLK
P1-25	GND	-	GND	-

Notes: 1. GPIO18 (Pin 12) supports PWM output. 2. I2C0_SDA0 and I2C0_SCL0 (GPIO0 & GPIO1) have 1.8K pull-up resistors to 3.3V.

Referring to K2 in the schematic, the Raspberry Pi expansion interface is provided by a simple double row, 0.1" (2.54 mm) pinheader connector giving the little computer 26 expansion signals.
These signals fall into one of three categories:

```
chmod +x blinky.py
```

Once done, you can run your program by typing the following command:

```
sudo ./blinky.py
```

With a little more code we could just as easy show CPU temperature or Network activity on the LEDs.

Conclusion

The Raspberry Pi PC offers enormous programming and software development potential for little money. The prototyping board described in this article allows the more hardware oriented electronics enthusiast to make the RPi do useful things in the real world. If you develop an RPi application everyone should know about, then do not hesitate to e-publish at www.elektor-labs.com. And don't forget to download that free Rpi poster from www.elektor.com/poster.

(120483)

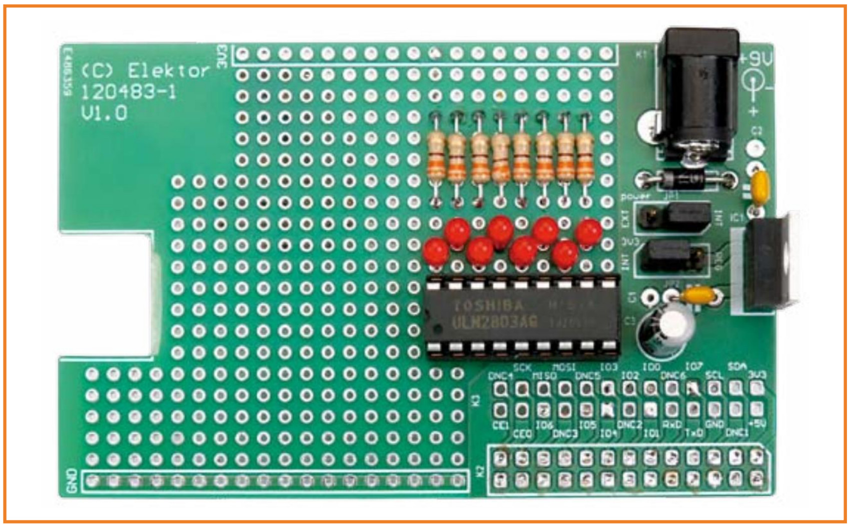


Figure 5.
The experimental circuit laid out in the prototyping area and connected up to the RPi through expansion signal connector K2.

Internet References

[1] Raspberry Pi site: www.raspberrypi.org

[2] Eben Upton interview: "What are you Doing?", Elektor April 2012; www.elektor-magazine.com/120228

[3] www.elektor-magazine.com/120483

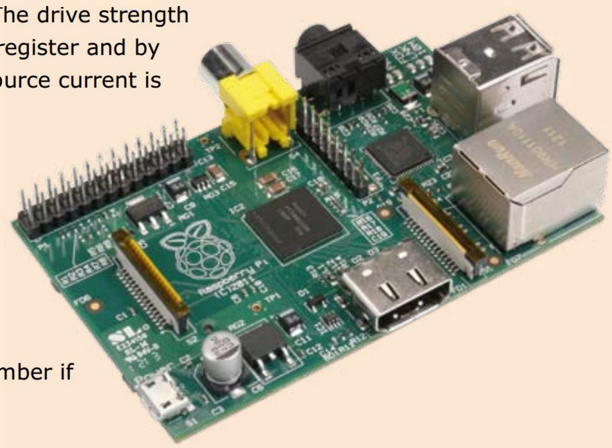
Table 2. P5 Header (Revision 2 Boards only)

Pin Name	Pin Function	Alternative
P5-01	5V0	
P5-02	3.3V	
P5-03	GPIO28	PCM_CLK
P5-04	GPIO29	PCM_FS
P5-05	GPIO30	PCM_DIN
P5-06	GPIO31	PCM_DOUT
P5-07	GND	
P5-08	GND	

Power: +5 V DC and 3.3 V DC* as well as 0 V
Input/Output: General Purpose Input/Output (GPIO) signals
Communications Interfaces: Serial UART, SPI and I²C

* Note: 3.3 V can only provide about 50 mA of current.

There are 17 general purpose input / output (GPIO) signals on the expansion header. Most of these can have alternative function. These alternative functions provide a UART, SPI and I²C interfaces.
Each GPIO pad can source between 2 and 16 mA depending on its drive strength configuration. The drive strength is set in a configuration register and by default after reset the source current is set to 8 mA.



In addition to the P1 Expansion Header, Revision 2 of the Raspberry Pi saw the introduction of a second, smaller expansion header, designated P5 (see **Table 2**). This adds another four GPIO signals but more importantly allows access to the PCM audio interface of the Broadcom 2835 chip. Also, the signals for P1 Expansion Header were slightly revised on the revision 2 boards, see **Table 1**. Notably, the I2C0 interface was replaced by the I2C1, a small but important thing to remember if you are planning to interface I²C devices.

USB-IO24 Cable

Use your PC for measurement, control and data conversion tasks the easy way

By Dr. Ing. Uwe Altenburg (Germany)

Transferring up to 24 digital signals using your computer's USB interface is simple with the multifunction cable described here. You can also sample up to eight analogue signals, generate PWM and servo signals and more besides. This article offers a simple protocol for controlling the cable and covers PC programming in the C# language as well.

These days you can get computers in every imaginable form — be they desktops, laptops or tablets. A welcome side effect is that prices have fallen, meaning that people who a few years ago would have employed a redundant Mini-Tower computer to control the lighting of their beloved aquarium can now do the same with a micro-sized Box PC and develop a neat control panel with touch-screen control.

Having said this, hooking up your homebrew electronics to a PC using USB connectivity is not exactly child's play and a computer would not be one's first choice for generating time-critical signals. On top of this you will need to select a communications protocol and find a suitable programming language.

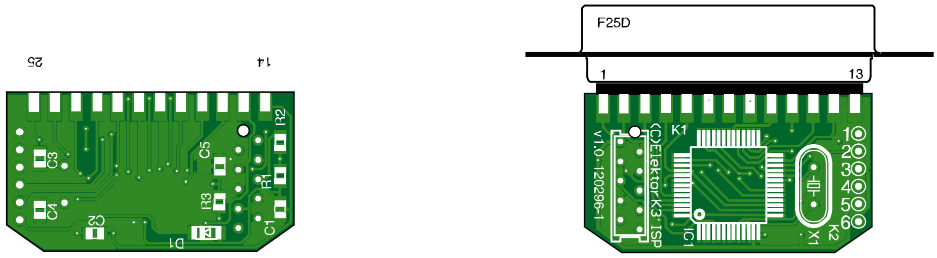


Figure 1.
There's more than enough room for the PCB inside the housing of the Sub-D connector.

COMPONENT LIST

Resistors
(SMD 0805, 5%)
R1= 100kΩ
R2 = 4.7kΩ
R3 = 100Ω

Capacitors
C1,C2,C5 = 100nF 50V, 10%, X7R, SMD 0805
C3,C4 = 22pF 50V, 5%, NP0, SMD 0805

Semiconductors
D1 = LL4148 (Minimelf)
IC1 = R8C/25 (SMD TQFP52)

Miscellaneous
USB/TTL Adapter cable TTL-232R-5V-WE, Elektor # 080213-71 [6]
K1 = 25-pin Sub-D socket (Amphenol type 77SD B25S)
K3 =10-way MicroMatch connector, with underside solder connections (TE Connectivity/Amp type 8-215079-0)
X1 = 18.432MHz quartz crystal, 18pF, 50ppm
Shell for 25-pin Sub-D connector, flip top (Assmann type A-FT 25)

Optional
10-way MicroMatch plug (8-215083-0 TE Connectivity/Amp)

No problem — all these challenges are solved here!

Circuit

For some years a USB/TTL converter cable (the TTL-232R-5V-WE [1]) has been sold by FTDI. It is also available from Elektor [2]. Using this cable, which besides RxD, TxD, RTS and CTS signals also provides a 5 V supply voltage, it is extremely easy to connect a microcontroller to a PC. The cable contains an FT232R USB-to-serial converter, located within the epoxy-potted USB connector. The idea springs to mind that you could attach a 25-pin Sub-D socket at the other end of the cable, with a second microcontroller inside the connector housing there. The 8-bit microcontroller employed here is a derivative of the R8C family, which has been featured already in Elektor many times.

The R8C25 [3] has adequate memory resources with 64 KB Flash and 3.5 KB RAM. Its 10-bit analogue converter and high-performance timer structure make it ideally suited for control tasks. The 25 connections of the Sub-D connector are sufficient to make all the pins of three controller I/O ports and the necessary ground connection available externally. There is sufficient room in the connector shell to install a small PCB with the dimensions 37x20 mm (**Figure 1**). Besides the microcontroller there is little else on the board: only an 18.432 MHz crystal, a programming connector and a handful of discrete components. The plan is to offer the PCB ready-assembled (more about this on the web page for this article [6]). All you do with the PCB is solder it onto the FTDI cable and then fit this assembly inside a suitable connector housing.

In terms of circuitry (**Figure 2**) there are few items of note. The microcontroller takes its power over the USB cable. The RTS wire of the USB-

Characteristics

- 24 digital in/outputs or alternatively up to
- 8 analog inputs (10-bit resolution)
- 8 PWM outputs (10-bit resolution)
- 8 RC servo signals (10-bit resolution)
- 4 counter inputs (16-bit up/down)
- 2 ICs MAX7219 selectable for up to 128 LEDs

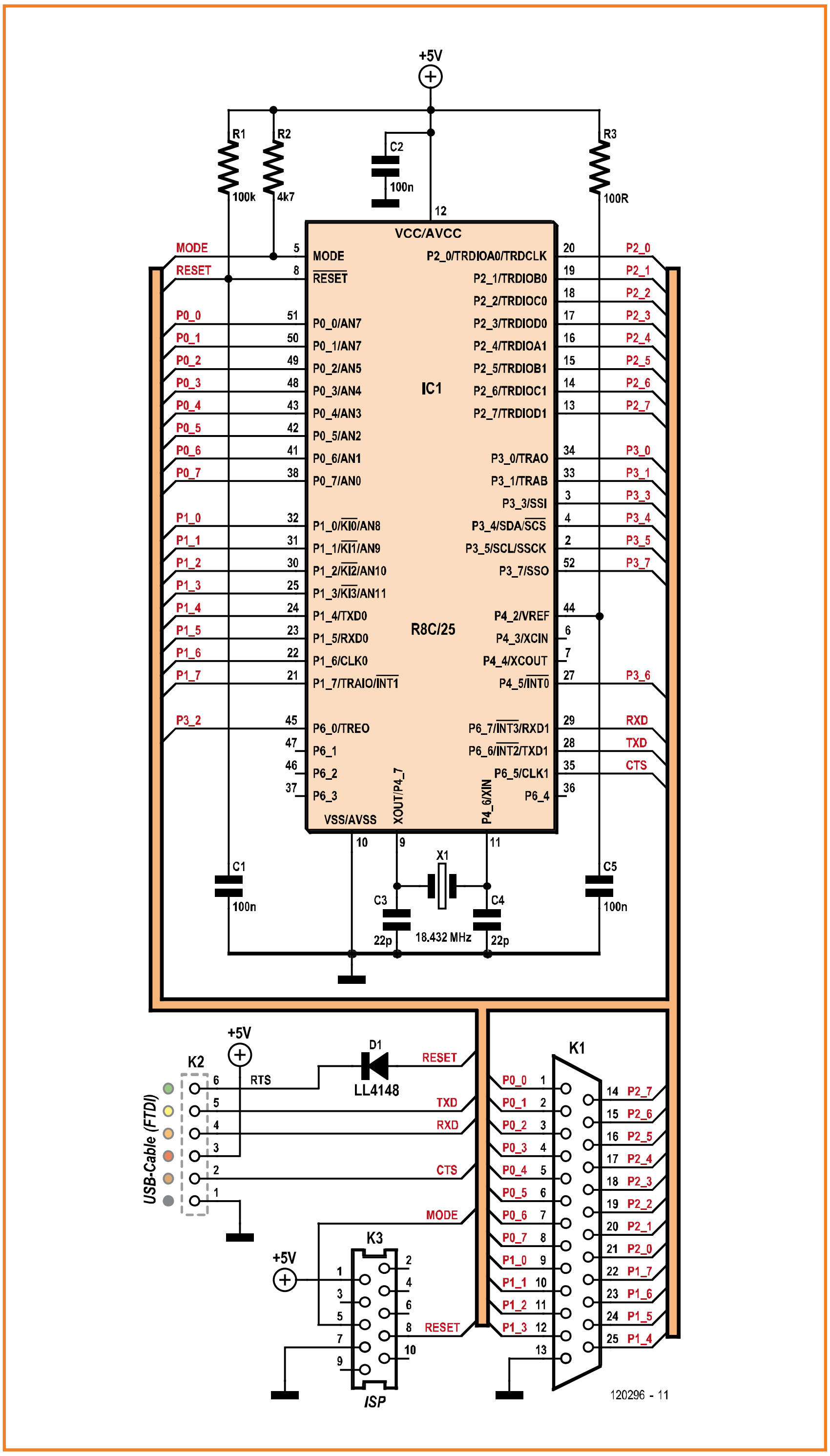


Figure 2.
Besides the microcontroller you need just a crystal and a few discrete components.

serial cable is connected via diode D1 to the reset pin of the microcontroller. This enables you to initiate a hardware reset of the microcontroller over the USB interface. The TxD and RxD signals of the USB-serial cable are used for communication between the PC and microcontroller.

Ports P0 to P2 of the microcontroller are taken to the Sub-D connector (**Figure 3**). This makes all the essential signals of the microcontroller available for use.

The pins of Port0 can be configured as analogue inputs, enabling up to eight analogue signals to be sampled. In line with standards, the reference voltage input of the analogue-to-digital converter is linked via an R-C combination to the 5 V supply, which provides a measurement range from 0 to 5 V with adequate accuracy. However, for situations when you place a value on higher accuracy in the measurement of analogue signals, an LM4040 reference voltage source IC can be fitted in addition.
Alternatively Port1 offers up to four interrupt inputs, which would enable, for example, high-speed counters to be realized. There is additionally a serial interface on Port1 that can be used in UART and SPI modes.

The outputs of the two 16-bit timers of the R8C25 are on Port2. These enable you to generate up to eight PWM signals simultaneously. A PWM signal can be used, for example, for rotational control of DC motors or to generate analogue signals. Altogether our USB IO24 cable provides a multitude of possibilities! Right then, how does the software check out now?

Firmware

The author has produced firmware for the controller that makes driving it with a PC very straightforward (the controller on the ready-built PCB is preprogrammed with this firmware). Data transfer between PC and microcontroller take place at a data rate of 115,200 baud with 8 data bits, 1 stop bit and no parity (115200,8,N,1).

Instead of a binary data protocol with a fixed format (comprising, for example, type and length of data plus checksum) we use simple text commands for communication. This approach is flexible and simple to understand, rendering the communication extremely ‘transparent’ and obvious. To compose commands we need nothing more than a simple terminal program, which makes the whole affair independent of the operating system used.

Following reset, the microcontroller begins with initialization. There is not a lot to do at this stage

because the exact function of the ports and timer is determined later on by the commands. All that needs to be initialized right now is the clock generation and naturally the serial interface for connection to the PC.
Beyond this the main program consists only of an endless loop — the Command Interpreter (**Listing 1**). In this a command is formed of a sequence of ASCII characters, followed finally by a CR (Carriage Return). More precisely, several commands can follow one another and only the last one needs to be finalized with a CR. This provides greater flexibility for applications. The character string *X10 P0.0=1 P0.0=0 <CR>* is therefore a valid command and means: "Set Port0.0 to 1 and back again 10 times".

The function `ReadCmd()` reads in a command sequence up to the CR. All blank characters are removed from the sequence using `SkipBlanks()` and following this, the function `Execute()` is invoked. If the function of each command can be executed successfully, a positive value is given, otherwise a null. According to the outcome either *OK* or *ERROR* is then transmitted back to the PC. Execution of a command sequence can take some

Listing 1. Main routine with command interpreter.

```
// --- Main ---
VOID main()
{
  InitCrystal();                          // init crystal
  InitUart();                             // init uart
  EI;                                     // ints

  SendString("\xCPIOCable V1.5\r\nOK\r\n");// version

  for (;;)      // endless..
  {
    ReadCmd();                            // read command
    SkipBlanks();                         // remove spaces

    if (Execute())                        // execute commands
      SendString("\r\nOK\r\n");
    else
      SendString("\r\nERROR\r\n");
  }
}
```

Listing 2. Interpretation of port commands.

```
// --- Macros ---
#define Digit(p)    (*p++ - '0')          // get a digit
#define GetIf(p,c)  (*p == c ? p++, 1 : 0)// get if char

// --- Read port ---
BYTE GetPort(BYTE nPort)
{
  switch (nPort)
  {
    case 0: return p0;                    // p0 input
    case 1: return p1;                    // p1 input
    case 2: return p2;                    // p2 input
  }
}

// --- Write port ---
VOID SetPort(BYTE nPort,BYTE nBits,BYTE nMask)
{
  BYTE nSet = nBits & nMask;

  switch (nPort)
  {
    case 0: p0 = p0 & ~nMask | nSet; break; // p0 output
    case 1: p1 = p1 & ~nMask | nSet; break; // p1 output
    case 2: p2 = p2 & ~nMask | nSet; break; // p2 output
  }
}

// --- Port command [Pn|Pn=v|Pn.b|Pn.b=v|Pn~|Pn.b~] ---
BOOL PortCmd()
{
  BYTE nPort = Digit(pCmd);                 // port
  BYTE nMask = 0xFF;
  BYTE nPin  = 0;

  if (GetIf(pCmd,'.'))                      // '.'
  {
    nPin = Digit(pCmd);                     // pin
    nMask = 1 << nPin;                      // mask
  }

  if (nPort < 3 && nPin < 8)                // check
  {
    BYTE nValue;

    if (GetIf(pCmd,'='))                    // '='
    {
      nValue = GetValue() << nPin;          // value
      SetPort(nPort,nValue,nMask);          // set port
    }
    else if (GetIf(pCmd,'~'))               // '~'
    {
      nValue = GetPort(nPort);              // get port
      SetPort(nPort,~nValue,nMask);         // set invert
    }
    else
    {
      nValue = GetPort(nPort) & nMask;      // get port
      nValue = nValue >> nPin;              // get value
      SendValue(nValue);                    // send value
    }

    return TRUE;
  }

  return FALSE;
}
```

Table 1. Summary of commands supported			
A to	A*n*	n = 0 to 7	Configure Pin An as analogue input. Analogue values returned in the range 0 to 1023.
C to	C*n*, C*n* = *x* C*n*+, C*n*-	n = 0 to 3 x = 0 to 65535	Return/set current value of counter CNTn, in the range 0 to 65535. Use + or – to determine count direction.
P to	P*n*, P*n* = *x* P*n*.*b*, P*n*.*b* = *y* P*n*~, P*n*.*b*~	n = 0 to 2 b = 0 to 7 x = 0 to 255 y = 0 to 1	Read/write to an 8-bit wide Port (or an individual Pin). Use ~ to select between Port and Pin modes.
S to	S*n*, S*n* = *x*	n = 0 to 7 x = 0 to 1023	Program one of the PWM Pins as output of the RC servo signal. Mid-position of the servo is when x = 512.
W to	W*n*, W*n* = *x*	n = 0 to 7 x = 0 to 1023	Program one of the PWM Pins as output of a pulse-width signal. The pulse-width has a range from 0 to 1023.
D to	D*n* = x,x,x,x,x,x,x,x	n = 0 to 1 x = 0 to 255	Control a MAX7219 device, to which up to 64 LEDs can be connected. Simultaneously 8 data bytes 'x' are sent.
R to	R*n*,R*n* = x	n = 0 to 6 x = 0 to 255	Read/write to the access control register: R0: Data flow direction from P0, 0=Input,1=Output R1: Data flow direction from P1, 0=Input,1=Output R2: Data flow direction from P2, 0=Input,1=Output R3: reserved R4: PWM frequency, 0=1 kHz, 1=2 kHz, 2=4 kHz R5: PWM time delay, from 0=none to 99=slow R6: Brilliancy 1. MAX7219, from 0=dark to 15=bright R7: Brilliancy 2. MAX7219, from 0=dark to 15=bright.
T to	T*n*	n = 0 to 6500	Time delay of n * 10 µs.
X to	X*n*	n = 0 to 65535	Repeat all instructions in the command line following the X symbol n times.
?	?		Display a summary of all commands implemented.

time. During this period additional characters can already be transmitted across the serial interface by the PC, however. With the baud rate of 115200 baud set, transmitting a character takes a mere 90 µs! For this reason reception takes place in an interrupt routine that deposits the characters in a ring buffer. `ReadCmd()` then reads the characters out of the ring buffer.

The commands that are possible are distinguished or differentiated by their first character. So the port command starts with a *P*, polling the analogue inputs with an *A*, setting a counter with *C* and so on, with either capital or small letters allowable in each case. For each command there is a special function in the code that recognizes the syntax feasible. **Listing 2** gives the functions for the port command.

In the most straightforward case the current state of a port is sampled using *P0*, *P1* or *P2*. The interpreter sends back the result as a numeric value from 0 to 255. To alter the state of a port you can also assign a value to it (*P0 = 0 to 255*). The function `PortCmd()` knows whether it is in assigning or sampling mode by the "=" symbol. This also makes it possible to address individual bits of a port. To do this you suffix the port number with the pin number *P0.0, P0.1 to P0.7*. Another function here is the "." symbol, which indicates whether the entire port is intended or just one pin. Finally the command can also toggle a port pin — this is done by including "~" (*P0.0~*).

Command set

The port commands already described provide an extremely effective means of controlling a digital I/O card with relays and optocouplers. However, we have not yet configured the data flow direction of the individual pins of the USB IO24 cable. Following a reset all pins are programmed to be inputs. This factor needs to be taken into consideration at the outset when developing your own hardware — the output stages must take up a fixed state unconditionally. The command *R* allows various control registers to be addressed (written to). Registers R0, R1 and R2 contain the data flow direction of the three ports. Bit 0 of R0 determines the data flow direction of pin P0.0, Bit 1 determines that of pin P0.1 and so forth. Digit 1 indicates output and a 0 signifies input. The command *R0 = 15<CR>* switches the data flow direction of the lower four pins of Port0 to output. Alternatively you can also write *R0 = $0F<CR>* or *R0 = %00001111<CR>*. The dollar symbol indicates hexadecimal numbers and the percent sign binary figures.

Table 1 sets out all currently supported commands for the USB IO24 cable. As well as the commands for ports and registers already mentioned there are also further, rudimentary commands.

For sensor applications the command *A* is definitely vital for sampling the analogue inputs. Up to eight analogue values can be digitized. The corresponding pins AI0 to AI7 do not have to be configured additionally — they are automatically programmed as analogue inputs the first time you type the command *An*.

Since it is in principle possible to transmit several commands at once, it is also possible to sample multiple analogue inputs with a single command. The command sequence *A0 A1 A2<CR>* polls the first three analogue inputs and sends their value as the response. Multiple values in a response are always separated by empty spaces.

The equivalent of an analogue input would be an analogue output. It is true that the microcontroller employed does not support analogue outputs directly but we can generate up to eight analogue values nevertheless by using the workaround of producing PWM signals. PWM signals are turned on with the command *Wn*. As with the analogue inputs, no extra configuration is required; however, you can also set the PWM frequency in control register R4. To derive an analogue value from a PWM signal we need a low-pass filter. An R-C combination is sufficient in the simplest of cases but it is better to use an opamp to construct a higher order low-pass, for example a Butterworth filter.

Using the commands presented so far it is already feasible to cover many standard applications with digital or analogue in and outputs. If you reckon that theoretically up to 127 USB IO24 cables could be connected to a computer, this would provide 127 × 24 = 3,048 digital inputs and outputs! All the same, every once in a while special solutions are required for those challenges that occur occasionally in the realm of DIY electronics. For those alone additional powerful commands are implemented, which we will deal with shortly along with sample applications.

Instructions for RC servos

Servo motors, used widely by radio-controlled model aircraft hobbyists and available with all manner of power capabilities, have recently met approval also for robotics projects, model railroads and other mechanical applications. They regulate their position against a predetermined set point. The set point for RC servos is coded as a pulse with a width of 1 to 2 ms, repeated every 20 ms (**Figure 4**).

To produce this pulse sequence is a task that cannot be handled directly by a PC. The timing is extremely critical — particularly the pulse width. Even slight jitter causes clearly audible buzzing in the servo (and tied up with this, raised power consumption). For this reason we have a special

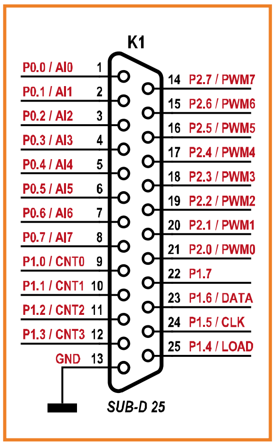


Figure 3. Pinout of the Sub-D connector.

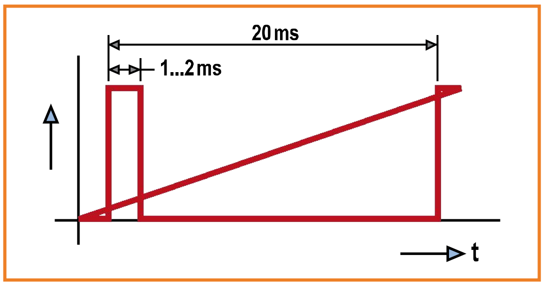


Figure 4. Pulse sequence for RC servos.

command *Sn*, which can generate up to eight servo signals as an alternative to PWM signals. It is also possible to produce both PWM and servo signals simultaneously. However, the timer structure in the R8C25 means that four pins of Port2 must always share the same timing — so that pins P2.0 to P2.3 generate PWM signals, whilst pins P2.4 to P2.7 take care of servo signals.
The newer RC servo motors are distinguished by their very fast setting speed. In principle this

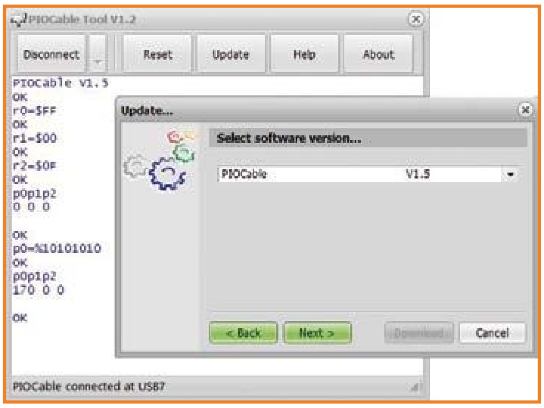


Figure 5.
PIOCable-Tool PC software

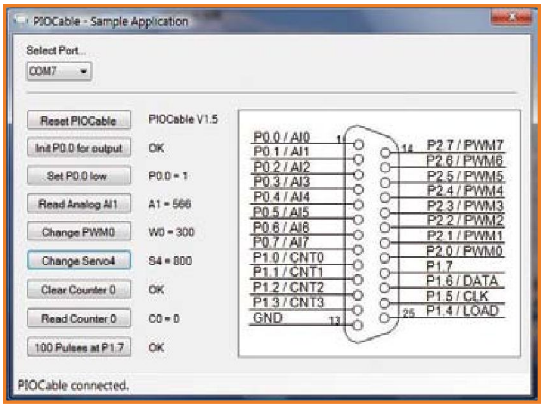


Figure 6.
Demo application in C#.

is an advantage, although not for intentionally slow movements, where very many set points need then to be provided. Doing this produces a kind of judder in the servo mechanism, however, leading to substantial oscillations in a robotic arm for instance. For this reason control register R5 allows the setting speed to be reduced specifically. The delay is introduced directly into the pulse generation process and reduces the amount of judder in the servos quite significantly. The value set in R5 affects the generation of PWM signals simultaneously.

Position or rotational speed

The control of low-power DC motors, as used in many projects, is a wide subject. At the simplest level a bipolar (or for higher currents, a field effect transistor) will suffice, and if the direction of travel needs to be reversed, also a relay or a bridge rectifier. The rotational speed can then be varied with the help of a PWM output. Difficulties begin when we want to control the rotational speed or advance to a precise position. This requires us to provide a feedback pulse generator on the motor shaft, for example an optical timing disk that interrupts a light beam. The pulses (leading edges) must then be counted.

Aggregating this count delivers the position or, referenced against a time interval, the rotational speed. Both can be determined using the command *Cn*. The USB IO24 cable possesses four inputs CNT0 up to CNT3, which in each case can be linked to a 16-bit counter internally within the controller. The command *C0+<CR>* activates the counter input CNT0, with the counter counting upwards), whereas *C0-<CR>* causes it to count downwards. The counter position can be read out using *C0<CR>*.

The direction of count can be toggled at any time, without affecting the current count. This makes it possible for the counter to always reflect the current position of a motor shaft. All you need do is change the direction of count before reversing the motor direction. On the other hand, to determine the rotational speed, you reset the counter to 0 with C0=*0<CR>* and read out the counter again after a suitable time. The count value is then a measure of the rotational speed of the motor.

LED displays

In the MAX7219 [4] Maxim provides universal circuitry for controlling up to 64 LEDs. Apart from a resistor this requires no other wiring. *Elektor* has already produced a 7-segment display for general application using this chip [5], although people frequently produce matrix displays as well.

Two of these chips can be connected to our USB IO24 cable cascaded in series. This makes it possible to control up to 128 individual LEDs (or two 8-element numeric displays or even combinations of the two). The data is transmitted to the displays serially. To achieve this all we need do is connect the three DATA, CLK and LOAD wires to the pins of the same name on the MAX7219. The command *Dn* then transmits eight data bytes at once — e.g. *D0=48,109,121,51,91,95,38,105<CR>*. The MAX7219 deposits the bytes sequentially in its Digit 0 to Digit 7 registers; each bit corresponds to an LED connected. The exact correlation can be found in the data sheet for the chip.

PIOCable Tool

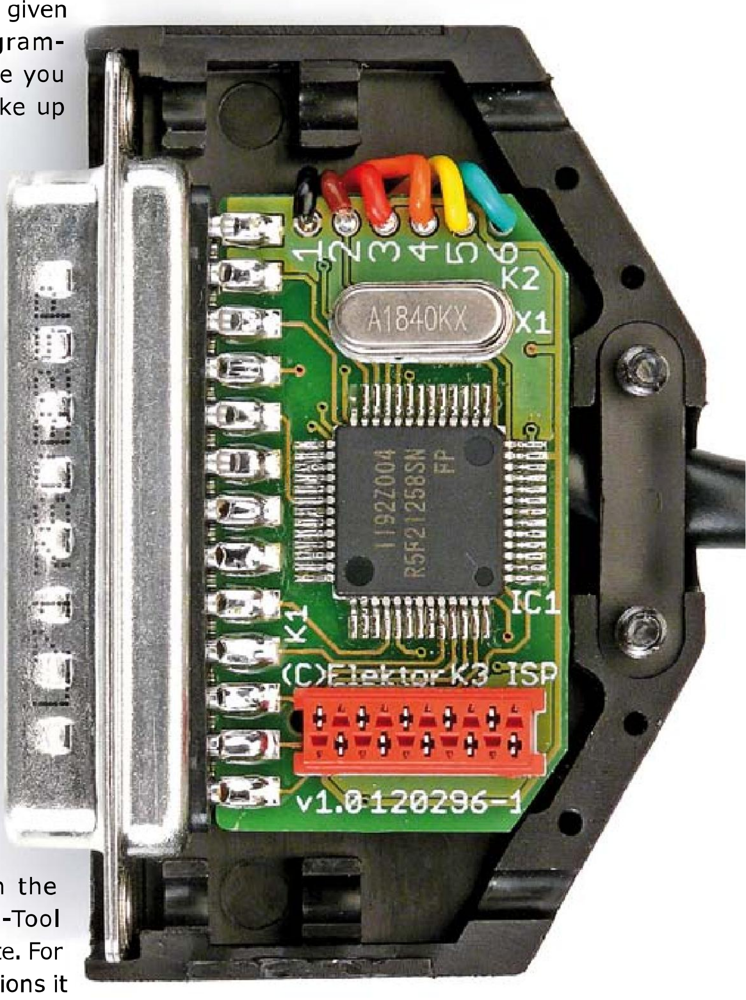
How tricky is it to commission the cable? Where do you find the necessary drivers? To test the functions do you need to write a program first? These are all typical questions when you want to hook up some new hardware to your PC. Luckily putting our USB IO24 cable to work is extremely easy. When you connect the USB plug into the PC, Windows searches for a suitable driver. With the drivers provided by FTDI you generally don't need to take any further action (assuming you are connected to the Internet) because they are installed automatically. In case of difficulty the FTDI website has some instructions for installing the VCP drivers necessary (VCP = Virtual Com Port).

Next you start up the PC program *PIOCable-Tool*, which was produced specially for commissioning and testing all of its functions (**Figure 5**) and can be downloaded from the web page for this article [6]. The source code for PIOCable-Tool can be found here too. The tool was produced using Delphi XE2 — essentially you should also be able to interpret the code with earlier versions of Delphi.

When you activate the software and click on the *Connect* button, a search for the cable is made automatically. If the search is unsuccessful you will need to look in Device Manager to find out which COM port addresses the cable and then enter this manually in the PIOCable-Tool program. The PIOCable-Tool software is fundamentally a so-called terminal program. All inputs are transmitted direct to the USB IO24 cable's microcontroller, which for its part responds immediately. This means you can enter all commands manually and at the same time check the reactions on each pin (using a 'scope for example).

The development of the cable firmware is ongoing, meaning it is possible to install new software versions on the microcontroller using PIOCable-Tool. The microcontroller on the ready-built PCB is equipped with a bootloader. Further software updates will be found both at Elektor [6] and also on the author's TinyBasic website [7].

If you wish to run firmware that you have developed yourself, the bootloader can of course do this for you likewise. The *High-Performance Embedded Workshop* development environment together with the matching C compiler can be downloaded from the Renesas website [8]. The controller can also be programmed without a bootloader, given a suitable programmer. In this case you will need to make up a small adapter cable that is fitted with a MicroMatch connector (see *Optional* section of component list). The bootloader (and the original firmware) are naturally overwritten by conventional programming.



Demo application

For simple applications (perhaps like a servo tester) entering the commands with the aid of PIOCable-Tool would be adequate. For all other applications it would of course be sensible to produce suitable PC software that then transmitted the corresponding commands to the USB IO24 cable. For the Delphi programmers among us, using the source code of PIOCable-Tool is recommended as a starting point.

However, self-written applications can also be produced extremely easily using Microsoft's programming language C# for .NET (a free 'Express' version of the development environment is available). As well as the usual control elements like menus, buttons or labels, .NET also provides complete serial interface components, with which our cable can be addressed very easily.

First we drag some components in the *SerialPort* category onto a task list. The components need to be given a name, e.g. "PIOPort". All other settings are then made in the source code. To do this we add a button to the schedule and type in its OnClick routine the following lines:

```
PIOPort.ReadTimeout = 100;
PIOPort.BaudRate = 115000;
PIOPort.NewLine = "\r\n";
PIOPort.PortName = "COM5";
PIOPort.Open();
```

For `PortName` we enter the port by which the cable is currently addressed. If everything works, the connection to the hardware is established. The author has produced a demo application in C# [6] and you can see a screenshot in **Figure 6**. In the code the lines shown above are invoked when the user selects one of the available interfaces in the drop-down box.

The setting `PIOPort.NewLine = "\r\n"` informs the component PIOPort that the invocations of `PIOPort.WriteLine()` and `PIOPort.ReadLine()` that follow should always end with the character sequence <CR><LF>. When writing a command you must not enter an extra <CR><LF> and when reading a response, this is read as far as the end of the line.

As an example let's now write a simple function that sets port pin P0.0 to 1. Again we drag a button onto the schedule and write the following line into the OnClick routine:

```
PIOPort.WriteLine("r0=1p0.0=1");
```

The command first sets the pin into output mode and then immediately to High — job finished! It would be a bit more complicated if you wanted to read analogue values for example. Then you would first need to await the response and then read this in using `PIOPort.ReadLine()`. To see how this works, look in the code of the demo application.
Now nothing stands in the way for you to start programming your own measurement and control software!

(120296)

Internet Links

[1] www.ftdichip.com/Products/Cables/USBTTL-Serial.htm

[2] www.elektor.com/080213

[3] www.renesas.com/products/mpumcu/r8c/r8c2x/r8c25/index.jsp

[4] www.maximintegrated.com/datasheet/index.mvp/id/1339

[5] www.elektor.com/081154

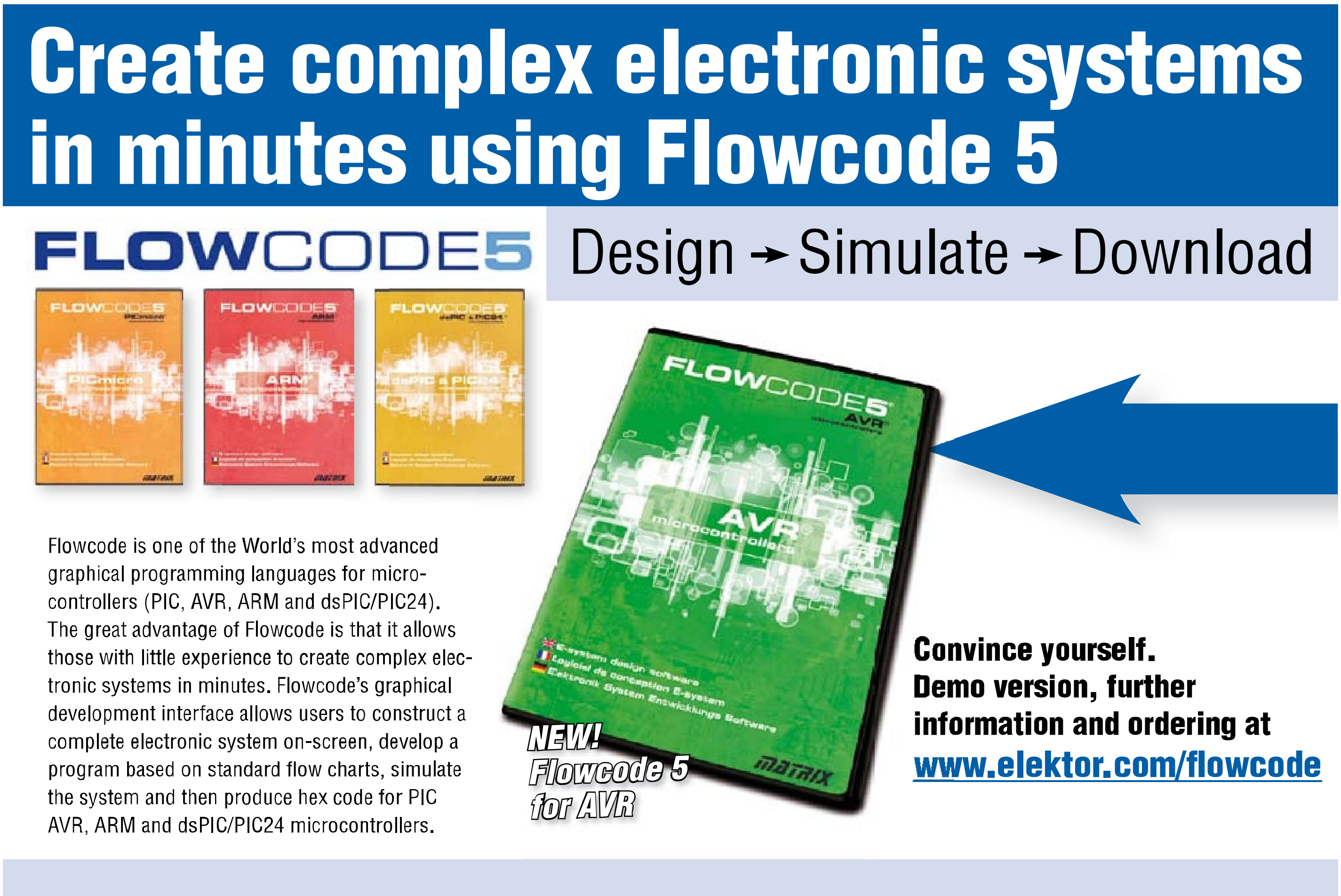
[6] www.elektor.com/120296

[7] www.tinybasic.de

[8] www.renesas.eu/products/tools/ide/ide_hew/index.jsp


THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT
create : electronics
FREE Stencil
with every PCB Prototype order
Assembly service
From as little as one component
Embedded RFID
authenticate, track & protect
your product
www.magic-pcb.com
NEW!
Free Phone UK: 0800 389 8560
sales@pcb-pool.com
PCB-POOL® is a registered trademark of Beta LAYOUT GmbH
Registered brands remain the registered trademarks of the respective manufacturer !
www.pcb-pool.com
Beta LAYOUT
create : electronics

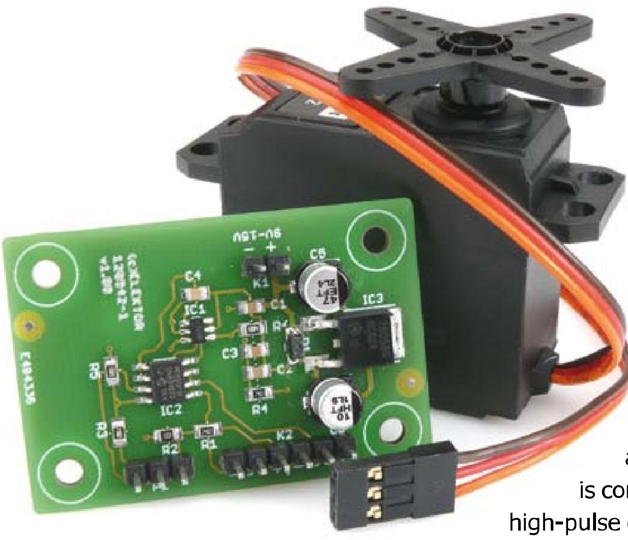

eSTORE®
Beta LAYOUT
Develop, assemble, solder
€ 36,50*
Arduino Mega (ATMega 1280-16AU)
compatible
Reflow Controller
€ 129,00*
UHF RFID Starterkit
Basic
€ 279,00*
Large Beta-Reflow-Kit
€ 129,00*
* Incl. VAT and exclusive of shipping costs
eSTORE® is a registered trademark of Beta LAYOUT GmbH
www.beta-eSTORE.com
Beta LAYOUT
create : electronics


Create complex electronic systems
in minutes using Flowcode 5
FLOWCODE5
Design → Simulate → Download
Flowcode is one of the World's most advanced graphical programming languages for micro-controllers (PIC, AVR, ARM and dsPIC/PIC24). The great advantage of Flowcode is that it allows those with little experience to create complex electronic systems in minutes. Flowcode's graphical development interface allows users to construct a complete electronic system on-screen, develop a program based on standard flow charts, simulate the system and then produce hex code for PIC AVR, ARM and dsPIC/PIC24 microcontrollers.
NEW!
Flowcode 5
for AVR
Convince yourself.
Demo version, further
information and ordering at
www.elektor.com/flowcode

Simple Servo Tester

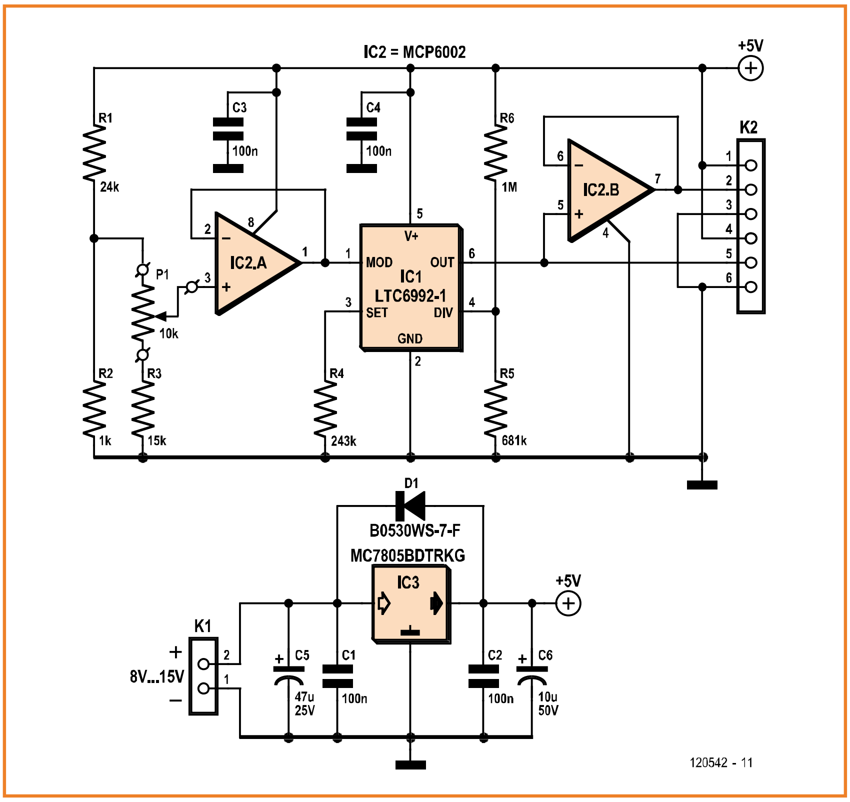
Nice SMD soldering exercise

By
Cederique Prevoo
(Netherlands)



This small test circuit is a handy aid for anyone who frequently works with servos. It is often sufficient that you can connect a servo to this circuit and try with a potentiometer whether the servo operates properly over its entire range.

The control of a servo is done with a PWM-signal (Pulse Width Modulation). The period is generally around 20 ms. The exact value is not that important, as long as the frequency is constant. The duration of the high-pulse determines the position of the servo. The minimum and maximum duration of this pulse are nominally 1 and 2 ms (with the occasional exceptions in some types). At 1.5 ms the servo is in the centre position.



This circuit generates such a PWM signal, where the pulse-width can be set with the aid of a potentiometer. In contrast with most simple servo test circuits, a 555-timer is not used here, but instead a Voltage-Controlled Pulse Width Modulator IC from Linear Technology, the LTC6992, was selected. In this IC the pulsewidth can be set using an analogue voltage and the frequency is set with a few resistors. This IC is a member of the so-called TimerBlox series made by LT. LT supplies a program to select the component values around the IC, where you only need to enter the desired function, range and frequency.

The control range for the positioning pulse was selected to be from 0.5 to 2.5 ms, to ensure that all servos can be tested to the full extents of their operating ranges. To achieve this the LTC6992 requires a control voltage between 0.12 and 0.2 V. To prevent influence between components, potentiometer P1 and attenuator resistors R1/R2/R3 are not directly connected to the MOD input of the LTC6992, but via a buffer instead. An MCP6002 from Microchip was chosen for this purpose, an opamp with a rail-to-rail input and output range, so that the relatively small voltage of the potentiometer is reproduced accurately.

The output signal and the power supply connections for the servo to be tested are available on K2. Because there is a spare opamp in the MCP6002, we have used this as a buffer for a second output signal on K2. This allows for the possibility of controlling two servos simultaneously or for connection to an oscilloscope to look at the output signal. Note that the pin-out of K2 does not correspond to the standard plugs of most servos. This requires making an adapter cable with suitable connectors.

A 7805 regulator provides a stable power supply voltage for both the circuit and the servo. The

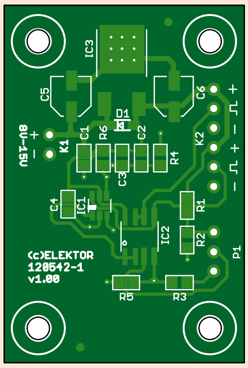
COMPONENT LIST

Resistors
(all SMD 0805)
R1 = 24kΩ, 5%
R2 = 1 kΩ, 5%
R3 = 15kΩ
R4 = 243kΩ
R5 = 681kΩ
R6 = 1MΩ
P1 = 10kΩ linear potentiometer

Capacitors
C1,C2,C3,C4 = 100nF
C5 = 47µF 25V (e.g. Panasonic EEEFT1E470AR)
C6 = 10µF 50V (e.g. Panasonic EEEFT1H100AR)

Semiconductors
D1 = 30V 0.5A Schottky diode, SOD323 case (e.g. Diodes Inc. B0530WS-7-F)
IC1 = LTC6992CS6-1, 6SOT-23 case, Linear Technology
IC2 = MCP6002-I/SN, SOIC8 case, Microchip
IC3 = MC7805BDTG, DPAK case, On Semiconductor

Miscellaneous
K1 = 2-pin pinheader, 0.1'' pitch
K2 = 6-pin pinheader, 0.1'' pitch
P1 = 2-pin pinheader, 0.1'' pitch
PCB # 120542-1 (www.elektor-magazine.com/120542)



input voltage may range from about 7 to 15 V. Don't operate the servo for extended periods of time at the higher end of the input voltage range, because the 7805 can start to thermally limit because it is only provided with a small cooling surface on the circuit board.

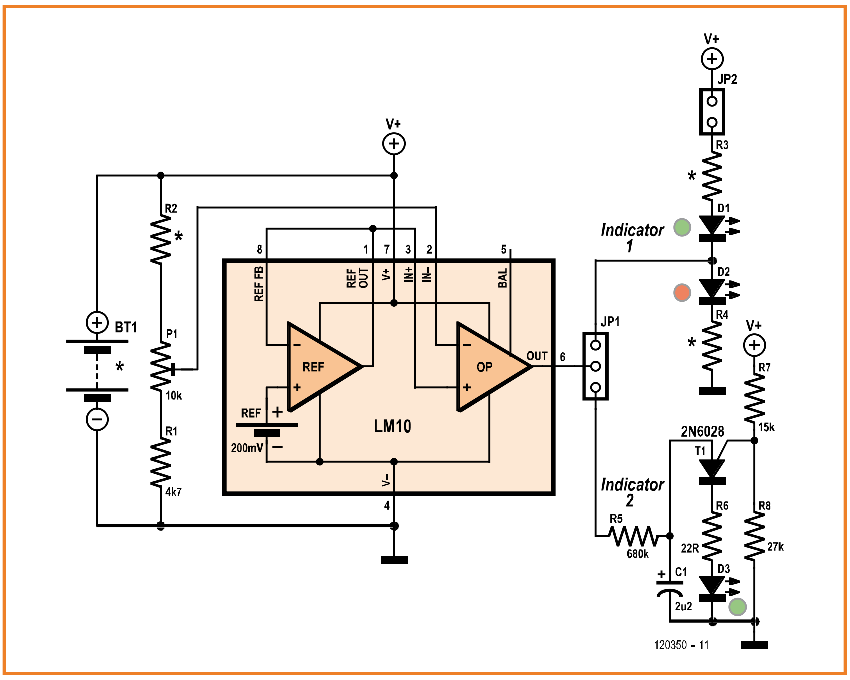
As was already indicated in the heading, this project is a nice soldering exercise for those who would like to get some experience with SMD components. As it happens, all the parts used here (with the exception of the headers) are SMD versions. Good luck with the construction!

(120542)

Battery-Nearly-Empty Indicator

Simple but effective

By J.F. Verrij
(Netherlands)



This circuit gives a signal using an LED when the voltage of a monitored battery drops below an adjustable minimum value.

The NEI (Nearly-Empty-Indicator) consists of a voltage reference, a comparator and an LED indicator.

The LM10 IC contains a reference source generating 200 mV, which is buffered by an opamp. The output voltage of this buffer is compared by the second opamp in the LM10 to a fraction of the battery voltage. The output of this comparator, pin 6, goes high when the battery voltage drops below the value set by P1. The output is connected to the indicator section. Here you can choose between two versions with two or one LEDs respectively.

With version 1 of the indicator a red LED will turn on when the battery is (nearly) empty (the green LED can serve as an on/off indicator). The version 2 indicator uses a programmable UJT and a high-efficiency LED. This produces flashes that even in full daylight will attract attention from a considerable distance (useful, for example, in remote controlled model boats).

The entire circuit can be built on a small piece of prototyping board. As an example, both indicators are built on the circuit board shown here and you can choose one or the other by placing a jumper on the 3-pin header. But in practice you would build the board with only one indicator.

The calculation of the voltage divider is done as follows. Choose the threshold voltage of the battery at which the LED should turn on. For example, for a LiPo cell this is 3.3 V per cell. At this point the battery is not quite empty, but it has the advantage that you can return a model boat to the shore before discharging the battery below the dangerous voltage of 3 V, when battery damage will occur.

As an example we take a battery consisting of 2 LiPo cells. The minimum voltage U_{min}, is then 2 x 3.3 V = 6.6 V. (For NiCd cells a suitable minimum value is 1.1 V per cell. 6 cells will then also result in U_{min} = 6.6 V.)

Calculate R2 as follows:
R2 = (48.5 x U_{min} − 14.7) kΩ

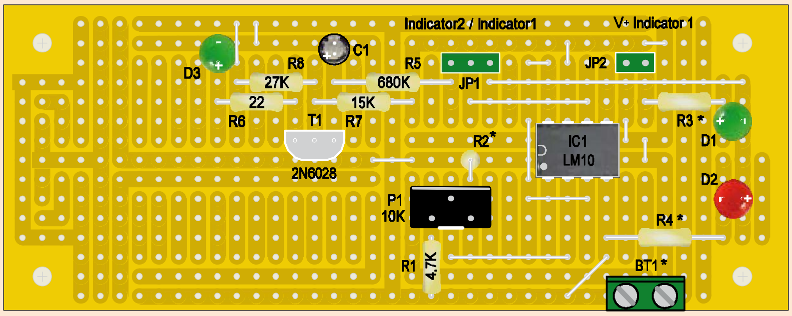
Therefore at U_{min} = 6.6 V:
R2 = (48.5 x 6.6 − 14.7) = 305.4 kΩ.

Take the nearest E12 value, which in this case is 330 kΩ.

Instead of the battery connect a lab power supply and set it to the desired U_{min} value (6.6 V). Now adjust the trim-pot so that the indicator is just on the threshold of on and off.

The series resistors R3 and R4 for the LEDs are calculated as follows (the difference in voltage drop between the green and red LED is relatively small and so for simplicity's sake we take an average of 2 V):

COMPONENT LIST



Resistors
R1 = 4.7kΩ
R2 = see text
R3,R4 = see text
R5 = 680kΩ
R6 = 22Ω
R7 = 15kΩ
R8 = 27kΩ
P1 = 10kΩ trimpot

Capacitors
C1 = 2.2µF 16V

Semiconductors
D1 = LED, green
D2 = LED, red
D3 = LED, high-efficiency, red
T1 = 2N6028
IC1 = LM10

Miscellaneous
Prototyping board e.g. Elektor UPBB (Elex) Type 1

$R_{series} = (U_{battery} - 2) / I_{LED}$

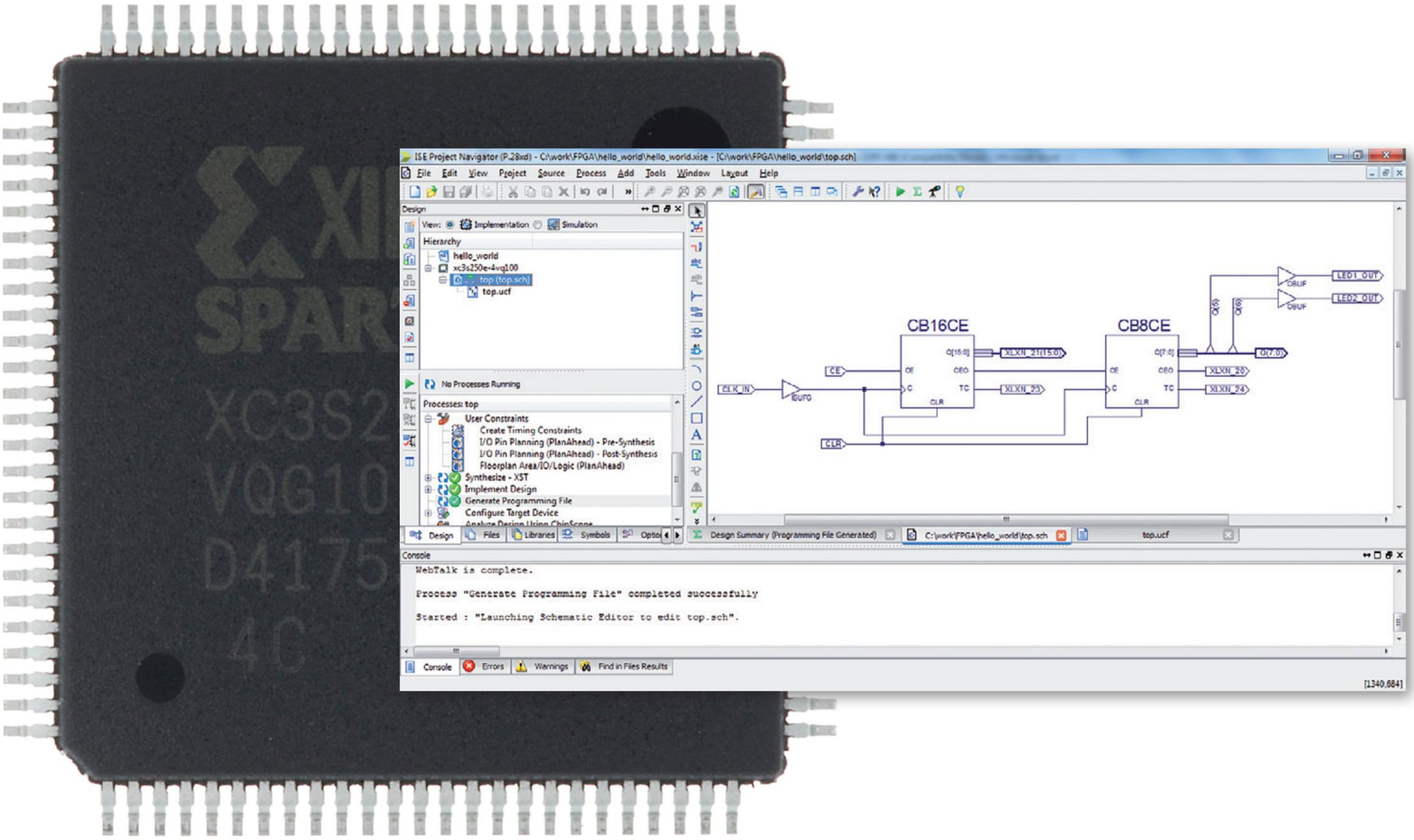
Assuming high-efficiency LEDs requiring about 3 mA and the aforementioned battery-pack, this results in a value of 1533 Ω; rounded this becomes a resistor of 1.5 kΩ.

The circuit is now ready to be built into a device, model aeroplane or model boat.

(120350)

Taming the Beast (3)

Counting to 100 with 250 K gates



By **Clemens Valens**
(Elektor Labs)

Last time we showed you how to set up an ISE project for programming the Elektor FPGA development board, with a simple LED blinker as a sample application. In this instalment we look at how to set up a hierarchical project structure with components you develop yourself. As a sample application for this instalment, I chose a simple up/down counter with a two-digit seven-segment LED display. In this instalment I also tell you how to define the pins on the board directly in the User Constraint File (UCF) without using the PlanAhead tool.

Here we go again

Open the "Hello World" project from the second instalment in ISE Project Navigator and click *File → Copy Project...*. Enter a name for your new project (I called it *part3*) and tick the option *Exclude generated files from the copy*, since you will be generating everything from scratch. All we're interested in at this point is creating a project with the right settings for the Elektor development board. Also tick the option *Open copied project*, since that will save you a few mouse clicks later on. Click *OK* and enjoy the view of crocuses coming up (it must be spring) while ISE prepares the new project and finally opens it.

In order to create a hierarchical project, you have to start by clearing out most of the items in the *top* level. Move the circuit in *top* to a new file, which I named *clock*. To do this, right-click the FPGA symbol on the *Design* tab and select *New Source...*, then select *Schematic*, enter the file name, and click *Next* and then *Finish*. Now you can use the usual Windows copy and paste operations to move the contents of *top* to *clock*. Make sure that the *CLK_IN* net with its IBUFG buffer remain in *top*, as well as the *LED1_OUT* and *LED2_OUT* nets (assuming you did your homework; otherwise you will only have *LED1_OUT*) with the associated OBUF buffers and bus taps. In the *clock* schematic you now have to add an I/O label in order to provide a proper termination for the clock input that you just cut off. Place an I/O marker as described in the second installment and assign the name *CLK* to the net that joins the clock inputs (*C*) together. Later on you will connect the LEDs in *top* in the same way as in the “Hello World” example.

Our aim here is to build an up/down counter that is driven by the clock signal from the *clock* circuit. We don’t need the *CLR* signal for this, and we can tie it directly to a fixed low level in the FPGA without using a pin for this purpose. This can be done in various ways; I chose the option of using a pull-down resistor so that the signal remains usable. You can find the pull-down resistor in the *Symbols* list after you click the *Add Symbol* button (or select the *Symbols* tab) and then select *General* under *Categories*. Drag the pull-down resistor to the schematic and place it on the *CLR* net.

Creating a component

Save all the files that haven’t been saved yet (marked by an asterisk in ISE). Open the *Design* tab and select the *clock* file in the *Hierarchy* list. If necessary, click the plus sign next to *Design Utilities* in the *Processes* window in order to open the item. Then double-click the entry *Create Schematic Symbol*. A bit later, after ISE has finished processing the task, you should see a big check mark next to the *Create Schematic Symbol* entry. Your new component has now been added to the top of the *Categories* list on the *Symbols* tab. There you will see an new entry with the project path name, which in my case is <*C:\work\FPGA\part3*>. Once you have selected this entry, a *clock* component appears in the *Symbols* window underneath, and you can place it in *top* in the usual way. Do this, and then have a good look at it (**Figure 1A**).

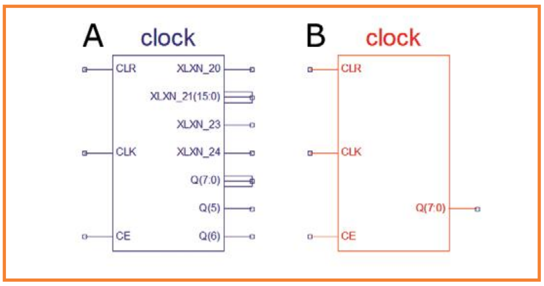


Figure 1.
The A version of the *clock* component has a lot of unnecessary inputs and outputs, which we remove to arrive at the B version.

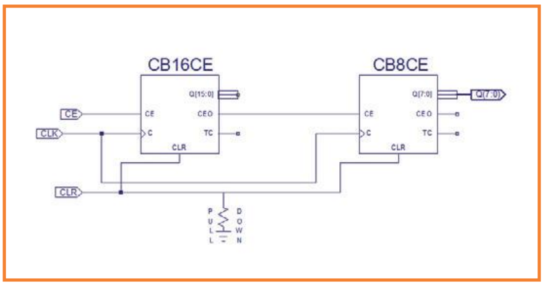


Figure 2.
The schematic diagram of our *clock* component.

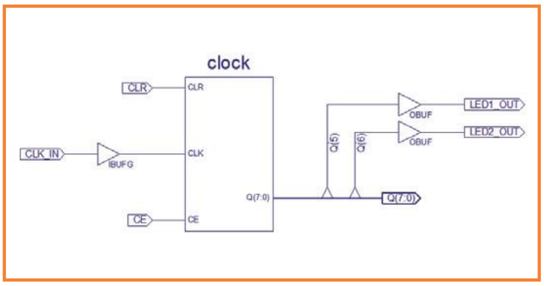


Figure 3.
The *clock* component in the *top* module.

As you can see, the component has a lot of “extra” inputs and outputs with names that start with *XLXN_*. You don’t need these for the present project, so you can simplify the component a bit. Remove all unnecessary labels and wire segments in the *clock.sch* file (see **Figure 2**). Select the *Design* tab, then select the *clock* module, and then place it under *top* with a different name (which includes “clock”). Right-click *Create Schematic Symbol* in the window underneath, select *Process Properties...* , and tick the option *Overwrite Existing Symbol*. Click *OK*, and then right-click *Create Schematic Symbol* again and select *ReRun*. Wait until ISE is finished, and then go to *top.sch*. Here you will still see the previous version of the symbol, but if you click on it ISE opens the *Obsolete Symbols* window, where you can edit the symbol. Select *clock*, click *Update,* and then click *OK* to close the window.

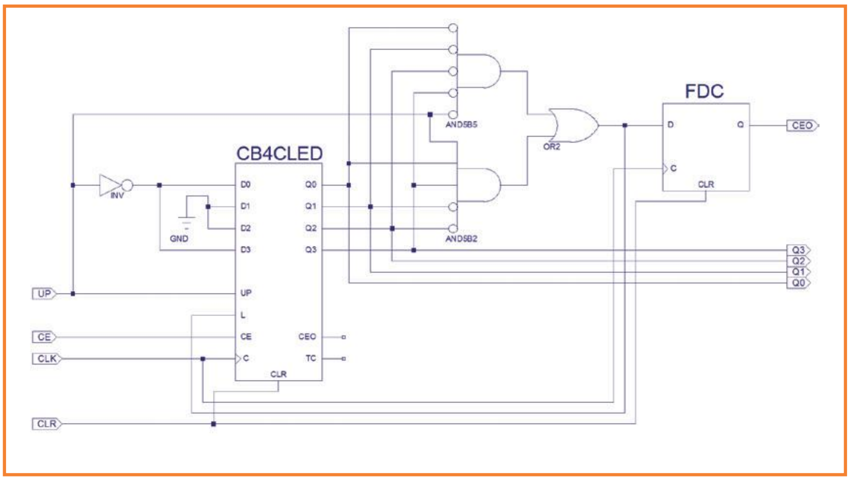


Figure 4.
A 4-bit BCD up/down counter derived from a 4-bit binary counter.

Now the symbol has been changed to the version in **Figure 1B**.

Connect the clock symbol as shown in **Figure 3**, and then launch *Implement Top Module* by clicking the button with the green triangle or by right-clicking the *top* module. A number of warnings of type *Xst:753* will appear on the screen:

```
WARNING:Xst:753 - "C:/work/FPGA/
part3/top.vhf" line 1962: Uncon-
nected output port 'Q' of component
'CB16CE_MXILINX_top'.
```

These warnings relate to the unconnected outputs of the components in the *clock* module. Although we always try to avoid warnings, it appears that ISE – or more precisely, XST – does not have any useful way to get around these warnings. In this regard, Xylinx says that these warnings can be ignored if the open outputs actually do not need to be used [2]. Even so, it's a bit annoying to have a yellow warning triangle next to the *Synthesize – XST* entry on the *Design* tab. Fortunately, the rest of the implementation delivers green check marks. Next, generate a bitstream by right-clicking *Generate Programming File* (or in some other way), and copy the file to the SD card on the FPGA board as described in the second instalment. Reset the board, and if you did everything right you should see the LEDs blinking as before. If so, you're ready to continue with the rest of the project. Otherwise you will have to find the problem and correct it.

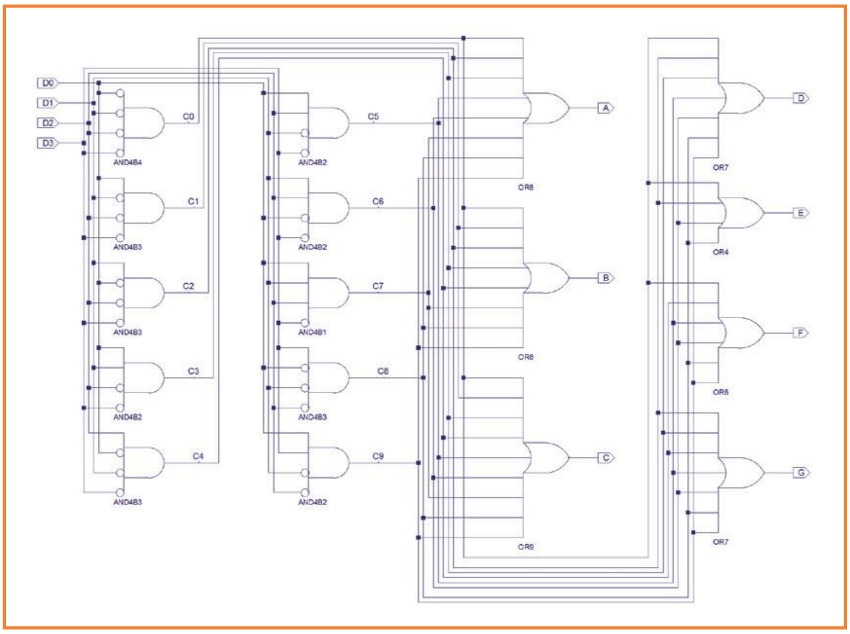


Figure 5.
The BCD to 7-segment decoder is a purely combinational design.

Building a BCD counter

The ISE libraries contain all sorts of counters, but they are all binary and what we need here is a decimal counter. That means we'll have to build it ourselves. Our counter needs to be able to count in both directions – up and down. To avoid unnecessary design effort, we base our work as much as possible on existing components in the ISE libraries. A bit of searching turns up the *CB4CLED* counter, a 4-bit binary up/down counter with reload capability. This is the only type of up/down counter available, so it will have to do. This counter counts from 0 to 15 or from 15 to 0, so we have to modify it so it counts from 0 to 9 or 9 to 0. This can be done by reloading the counter with a new starting value when it reaches the end count. For instance, when it is counting up and it reaches 9, it must be loaded with 0 on the next clock pulse, and if it is counting down and it reaches, 0, it must be loaded with 9. We can use bit of clever circuit design to derive the initial values from the *Up/Down* signal. This only requires a single inverter, as shown in **Figure 4** (note the signals UP and D0–D3).

A load signal must be generated when the counter reaches the end count corresponding to the counting direction. We use two AND gates to detect the end counts 0 and 9 for the counter. A wide variety of AND gates are available in the ISE libraries, and you can choose the ones with exactly the right number of inverting and non-inverting inputs. That saves a bit of work with the wiring. The two load signals are combined by an OR gate to generate the signal *L*. The *CB4CLED* counter responds to this signal on the ***next*** rising edge of the clock signal *C*, which is exactly what we want. You should bear in mind that the output of the counter always changes ***after*** the rising edge of the clock signal. Nothing happens immediately; there are always small delays due to propagation times in the logic.

On the next rising edge of the clock signal, the active level of signal *L* is loaded into flip-flop *FDC* and the *CB4CLED* counter loads the new starting value. As a result, signal *L* goes inactive a short time later, causing the output of the flip-flop to return to the inactive state ***on the next clock pulse***. Consequently, a pulse appears at the output of *FDC* each time ***after*** the counter has reached an end count. This pulse can be used to clock a subsequent counter.

Flip-flop *FDC* is more important than you might think at first glance, since it allows the next counter to be clocked directly by signal *L*. However, if you do this ISE will generate an error message during implementation:

```
WARNING:PhysDesignRules:372 - Gated
clock. Clock net XLXN_116 is sourced
by a combinatorial pin. This is not
good design practice. Use the CE pin
to control the loading of data into
the flip-flop.
```

In this message *XLXN_116* is the name of the net, which depends on the design. As the warning indicates, using combinatorial signals as clock signals in FPGA designs (and in other complex logic circuitry) is not recommended. “Combinatorial” means that the signal is generated entirely by a number of logical operations (logic gates) without any clock stage, and the delay of this type of signal depends on the number of gates it passes through. The delay may be variable or poorly defined, so there is no guarantee that the signal will be synchronized to the main clock signal. As a result, undefined states (such as glitches) can occur, and they may cause problems. Signal *L* is a combinatorial signal. You can synchronize it by feeding it to a flip-flop clocked by the main clock signal, which causes the warning to disappear.

Once the design is finished, you can turn it into a component in the previously described manner.

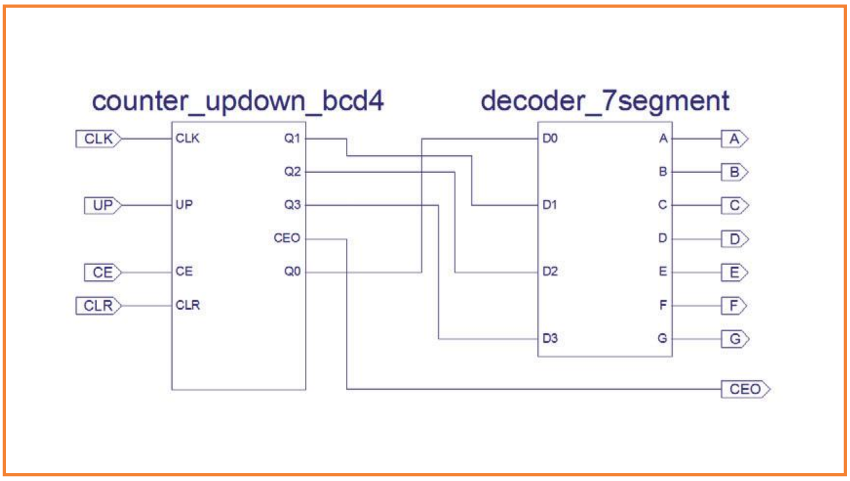


Figure 6.
A single-digit BCD counter with outputs for a 7-segment display.

Seven-segment decoder

To go with your counter, you need a 7-segment decoder with outputs for a 7-segment LED display. I couldn’t find one in the ISE libraries, so you will have to make it yourself. This sort of component is fairly simple, but it is somewhat bulky due to the four inputs (BCD) and seven outputs. My design, made entirely from logic gates, is shown in **Figure 5**. Another option would be to use lookup tables (LUTs), but that’s not what I chose. This component can remain purely combinatorial because it does not generate any signals that need to be synchronized with the clock signal. Draw this design in a new schematic source file and turn it into a component.

Seven-segment counter

Now you can use the two components you have just implemented to build a new component: a counter with an integrated 7-segment driver. Start by creating a new schematic source file, then drag the BCD up/down counter and the 7-segment decoder onto this file. Connect them to each other as shown in **Figure 6**. The outputs of my *counter_updown_bcd4* component are not arranged in an entirely logical order, which makes the schematic look a bit messy. So far I don’t know how to force them into a particular sequence.
After the schematic is finished, you can turn it into a component. After you do this, you will see that the two other components are shown below the new component on the *Design* tab of ISE. The hierarchical structure is starting to take shape.

The complete design

Now all the component are ready, and you can

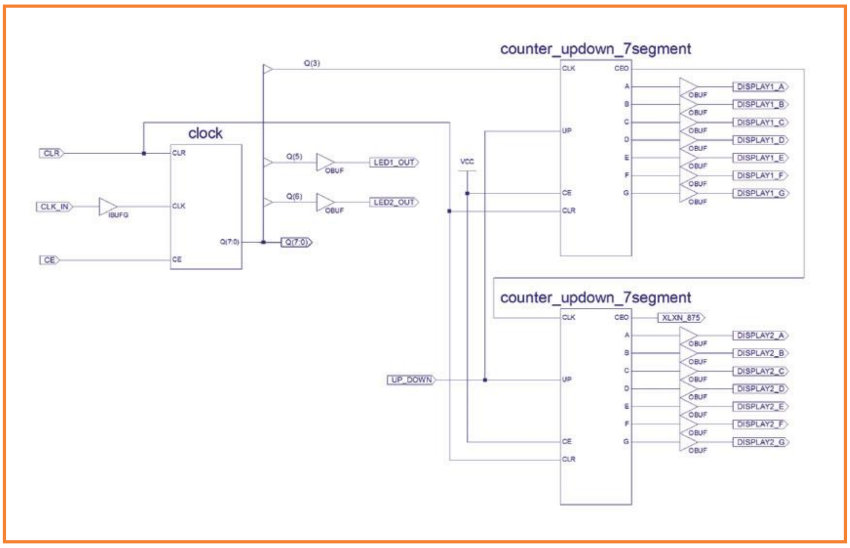


Figure 7.
The top-level design of the two-digit up/down counter.

start putting them together on the top sheet. In addition to the clock component, you need two seven-segment counters. Assign an OBUF to each output that must be connected to a pin of the FPGA. Except for the CLK signal, no buffers are necessary for the inputs, but you should put labels on them. Also put a label on the unused *CEO* output of the second 7-segment counter and a label on the *Q(7:0)* bus, to avoid warnings about unconnected outputs (our old friend *XST:753*). Anther thing worth knowing is that ISE secretly connects unconnected labels at the top level to FPGA pins. You can see this for yourself by using an oscilloscope to check the unconnected pins of the FPGA on the board. This means that you should think twice before connecting "sensitive" circuitry to FPGA pins without first explicitly driving the pins concerned in the FPGA (for example, if you think you might want to do so later on). It is better to temporarily set such pins to defined levels of your choice.

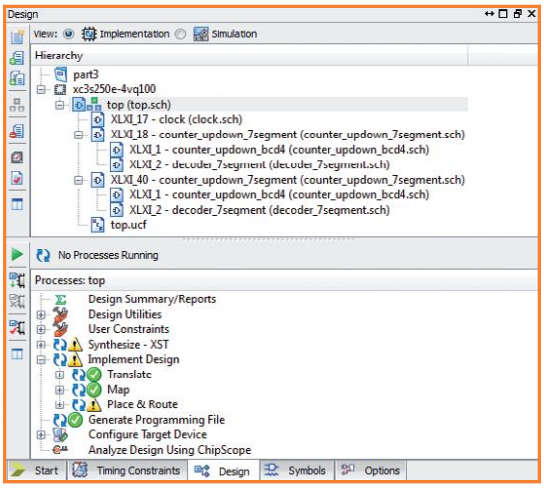


Figure 8.
The structure of the hierarchical design.

In the end, you schematic should look something like the one in **Figure 7**. Here I chose to clock the counters at 7.6 Hz (clock output Q3), but you may find this too fast. If so, choose another output for your counting rate.
The hierarchy on the *Design* tab is now complete, with the device above the top level and the components below it (**Figure 8**).

Editing the UCF file

All you have to do now is to connect the circuit to the pins of the FPGA. You can use the UCF file attached to the *top* module for this. Open the *Design* tab, and if necessary expand *top* by clicking the plus sign. You should now see the item *top.ucf*. Double-click it and wait for ISE to open the file. Now you are looking at a simple text file containing the definitions of the pins from instalment 2. It's a bit chaotic, but it is not especially difficult to understand. Each of the pins is connected to a net by a LOC variable. Naturally, the net name must appear in the design. "LOC" stands for location, and it includes FPGA pin number. Each pin also has an IOSTANDARD, which is always the same for the Elektor FPGA board: LVCMOS33. There is no distinction between inputs and outputs. Some pins can also have the attribute PULLUP or PULLDOWN, which refers to a pull-up or pull-down resistor in the FPGA.

The CLK_IN, LED1_OUT and LED2_OUT nets are always the same for the Elektor FPGA board, so you can group them at the head of the file and leave them alone. The other nets can be deleted or edited, since you need other ones. You can add a comment by starting the line with a hash sign (#). When assigning pins, remember that pin 37 (FPGA pin 13) can only be an input. The first part of my list looks like this:

```
# Hardwired pins
NET “CLK_IN” LOC = P32;
NET “CLK_IN” IOSTANDARD = LVCMOS33;
NET “LED1_OUT” LOC = P90;
NET “LED1_OUT” IOSTANDARD = LVCMOS33;
NET “LED2_OUT” LOC = P91;
NET “LED2_OUT” IOSTANDARD = LVCMOS33;

# 7-segment display 1
```

```
NET "DISPLAY1_A" LOC = P15;
NET "DISPLAY1_A" IOSTANDARD = LVCMOS33;
NET "DISPLAY1_B" LOC = P16;
NET "DISPLAY1_B" IOSTANDARD = LVCMOS33;
…
```

Implementing the design

Now you have everything you need to generate a bitstream for the FPGA. To do this, click the *Implement Top Module* button (refer back to the first instalment if necessary) and wait until ISE has finished. Unfortunately you will see a couple of warnings, but (fortunately) you can ignore them. Warnings of the following type:

```
WARNING:Xst:653 - Signal <dummy> is used
but never assigned.
```

are the fault of XST itself, since it creates auxiliary nets to build the 7-segment decoder and the 7-segment counter but doesn't actually do much with them.

The *Place & Route* (PAR) step also generates a warning:

```
WARNING:Route:455 - CLK Net:Q_3_OBUF
may have excessive skew because
```

It's hard to know what to do with this warning, since there's nothing after "because". The reason for this warning is therefore a mystery. For now, let's simply ignore it and hope that the circuit does not have a real problem with excessive skew. Next, select *Generate Programming File* to generate the bitstream, give it the name *config.bin*, and copy it to the SD card on the FPGA board. Restart the board, and if you have connected it as shown in **Figure 9**, it should start counting. You can control the counter by setting the appropriate levels on pin 27 (reset; active high), pin 28 (stop; active low) and pin 29 (up/down; high = up, low = down). The two LEDs on the board will blink as in the previous project.

Homework

Build the 7-segment decoder using LUTs. Remember that whenever you edit a component, you must not only generate it anew but also update it in the higher-level sheets. Otherwise you will get an error message during implementation. Maybe there is some way to have ISE handle all the updating automatically?

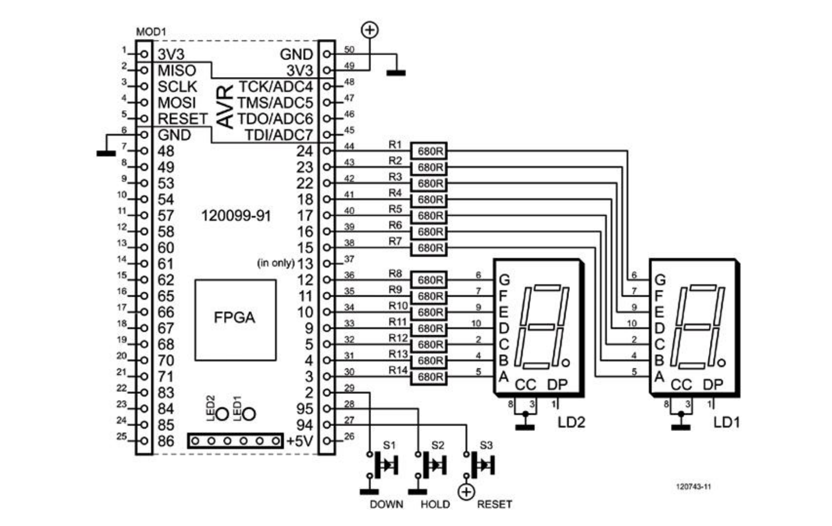


Figure 9.
Two 7-segment displays connected to the FPGA board @ Elektor Labs.

To be continued

In the next instalment we will delve into the simulation of this design. In the process there's a very good chance of coming in contact with a hardware description language (HDL), such as VHDL or Verilog.

(120743)

Internet Links

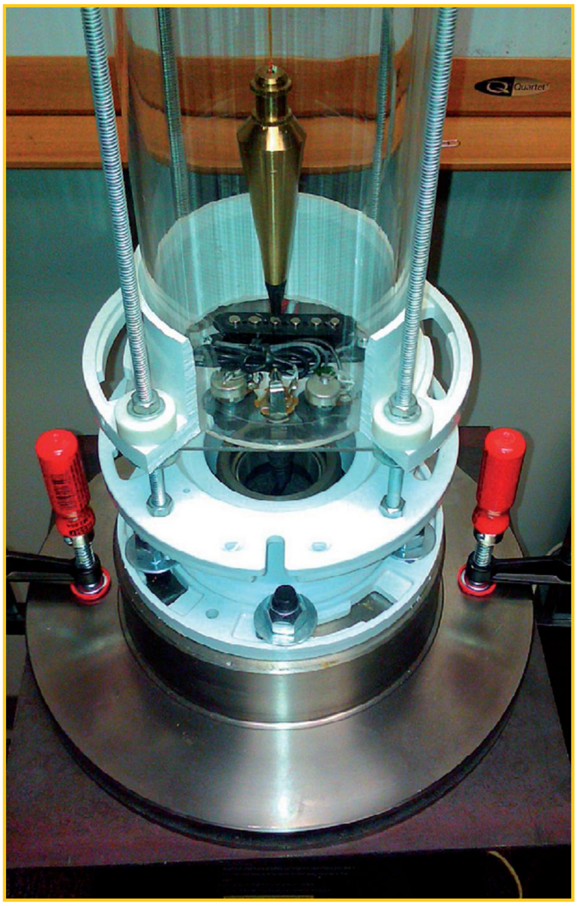
[1] www.elektor-magazine.com/120743

[2] www.xilinx.com/support/answers/14065.htm

Frontline breaking news

By Clemens Valens
(Elektor .Labs)

Elektor dot Labs is at the heart of Elektor, it is the place where all electronics related designs start. Projects and ideas are posted on the website; circuits get developed, debugged, tried and tested in our labs, and progress and results are reported back to you. Elektor dot Labs is also the frontline where all the action takes place. Here are some heartbeats from the front.



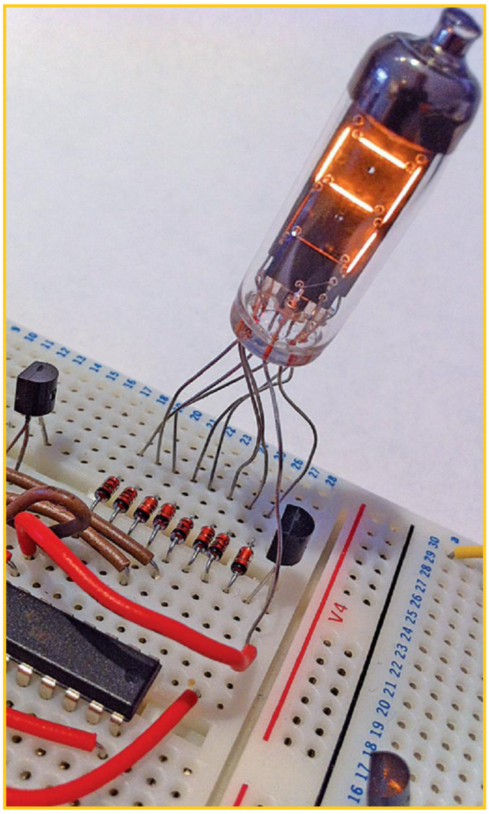
What's in a name?

Titles and product names have to be short to immediately grab the attention of the potential reader or buyer, or at least that's what they always told me. The problem with snappy titles is that they cannot convey too much information about the content or the product, which is why marketing departments and advertisement agencies got invented. Although Dot Labs original poster (OP) SuperlabTV is working on an interesting project, so far he has not found time, inspiration or both, to come up with a catchy name for his project. This is a pity because it definitely needs one. For now his project is called "*Electric guitar pickup seismograph with Arduino cell phone remote control*", a title that has the merit of being detailed and informative, but on the down side is hard to remember.

SuperlabTV created this seismograph as a platform for both demonstrating and learning about earthquake seismographs, Arduino microcontrollers, cell phone remote controlled servo motors and upcycling. According to Wikipedia, *upcycling is the process of converting waste materials or useless products into new materials or products of better quality or for better environmental value*. How much exactly this project 'upcycles' depends probably on the reader, but it is true that several parts for this seismograph were recovered from waste equipment: the base of the instrument is an old automobile brake rotor and foe the sensor the OP used an electric guitar pickup from an old guitar.

Wondering what the Arduino cell phone remote control is for? It allows you to create an artificial seismographic shock simply by sending a text message to the instrument so you can see that it works. Neat, ain't it?

Do you have a suggestion for a good title for this project?
www.elektor-labs.com/9121102688



No schematics, no followers

Elektor magazine has had its share of clocks and thermometers with nixie tube displays and now they have also surfaced on .Labs! The project "*Numitron Arduino Clock and Thermometer*" by 'Courty' is one of them. The nice thing of this project is that the OP, not having any previous experience with Numitron tubes, explains from the start how he went about and how he managed to get it all working in the end. Unfortunately, the OP did not post any schematics or source code so the project has limited use. If you, as a reader, would like to build Courty's clock/ thermometer please post a contribution or comment. If there are enough people interested we will try to get Courty to publish his design in the magazine.
OP JmBee posted a similar project, but based on a PIC16F887 instead of the ATmega328 used by Courty. He even went so far as to post his project in English ("*Numitron desktop clock - Using only classical components*") and in French ("*Horloge du bureau à tubes Numitron et composants classiques*"). JmBee used the same Numitron tubes as Courty. Also like Courty, JmBee did not post any schematics or source code so nobody can replicate his design. Therefore I ask you again: please post a message if you would like to see this design published or ask the OP to post his design files. Apparently, finished projects lacking details for other people to replicate the project are interesting, but will not attract a lot of attention.

Make some noise:
Courty: www.elektor-labs.com/9120902460
JmBee (English): www.elektor-labs.com/9120702370
JmBee (French): www.elektor-labs.com/9120702371

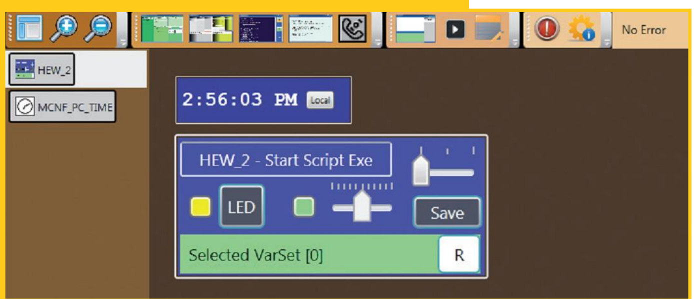
Editor's Choice

A number of .LABS projects have been selected by our editors and should be published in the near future. For some of these projects, sadly we found that the **original poster** (**OP**) does not reply to our messages. Therefore, if you posted a project, please check on a regular basis the email account you used to access .LABS. We will not get you in publication if we cannot get in contact with you.
Here is a selection of projects we think is interesting and which we would like to publish in the printed magazine.

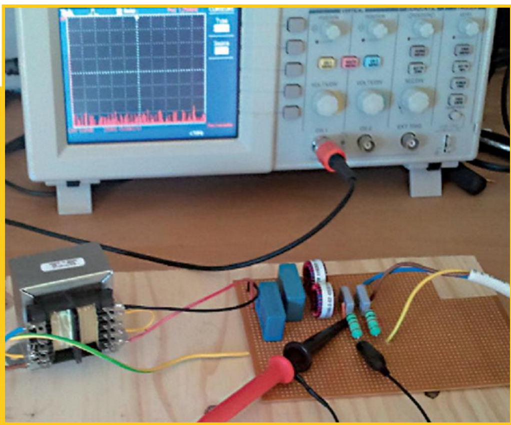
EDITOR'S CHOICE

Microcontroller Networking Framework

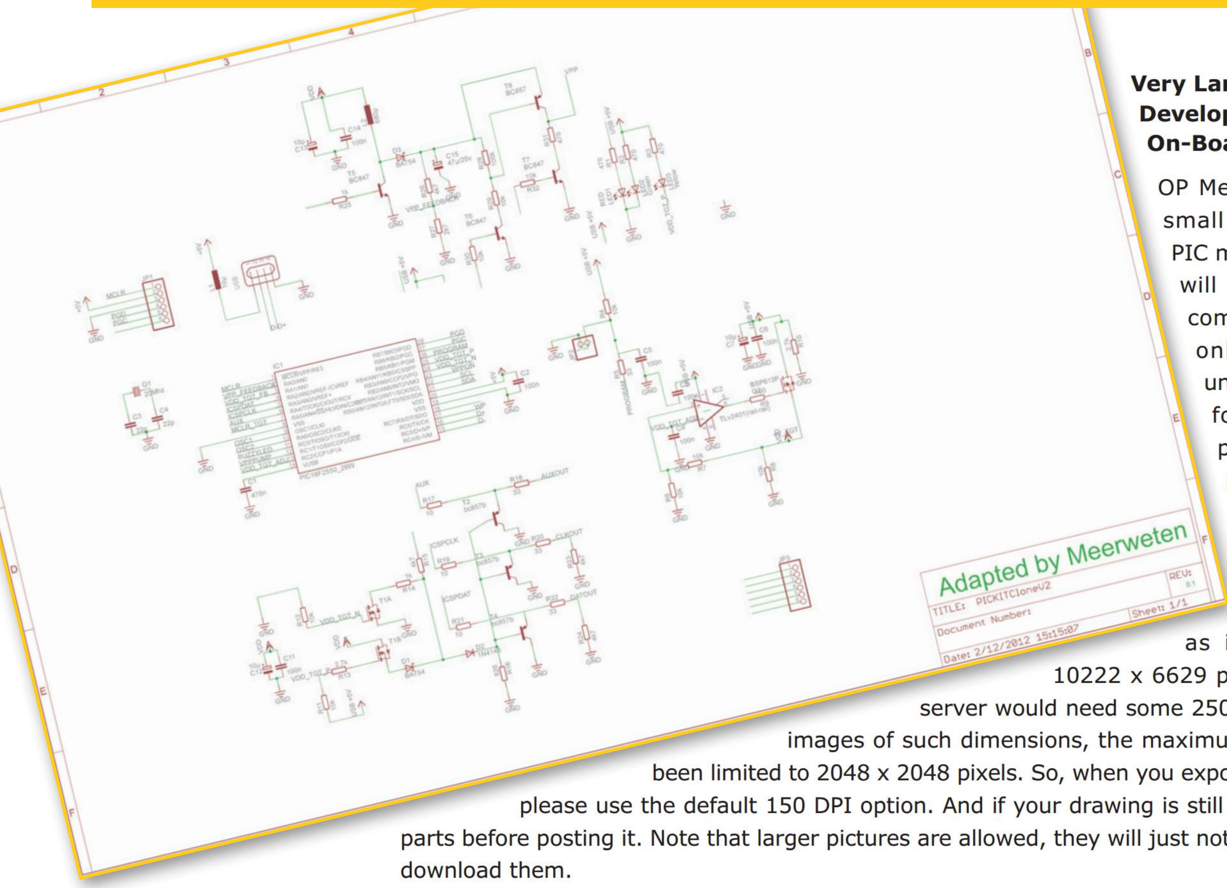
MCNF is a framework to build measurement, control and automation applications. It targets systems where a PC controls real-time functions in an embedded system. Networking is based on a command-response protocol handled by a small kernel on each microprocessor. Many standard functions are built in, like connectivity tests, read and write of embedded variables and saving them to EEPROM, gateway functions, etc. The network allows mixing of UART, I²C, SPI, Ethernet and other communication protocols.
www.elektor-labs.com/9121202735



Conducted Emissions Tester



Imagine the following scenario: you are working on a project which works fine in the morning but becomes unstable in the evening. After three days of head scratching you conclude that the problem is due to the energy saving lamp that helps you work when it is dark outside! Bizarre? Nope, it happened to OP Hooligan0. Annoyed by this waste of time the OP decided to develop a small line impedance stabilization network so that he can visualize with a spectrum analyser or oscilloscope with FFT the noise on the power line.
www.elektor-labs.com/9121202710



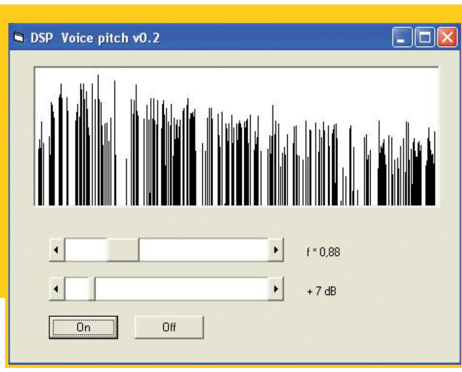
Very Large Image or PIC Development Board With On-Board Programmer

OP Meerweten is working on a small development board for PIC microcontrollers. The board will include its own PICkit2-compatible programmer not only to program the PIC under development, but also for use as a stand-alone programmer. Now the OP posted his design files including a PNG version of the schematic. The .LABS website however could not display this picture as it was a flabbergasting 10222 x 6629 pixels in size! Because the server would need some 250 MB of memory to display images of such dimensions, the maximum size of illustrations has been limited to 2048 x 2048 pixels. So, when you export a schematic from Eagle, please use the default 150 DPI option. And if your drawing is still too big, cut it into several parts before posting it. Note that larger pictures are allowed, they will just not be displayed, but you can download them.
www.elektor-labs.com/9121102689

www.elektor-labs.com

Real-time Pitch Shifter

Some people do not believe in coincidences, but what else can you make of this? OP bkelektronik, also known as the reputed Elektor author Burkhard Kainka, posted a project on .Labs in German entitled *Echtzeit Stimmhöhen Teiler*, a real-time pitch down-shifter for people that have trouble hearing high frequencies. A few days later I received an email from such a person looking for such a circuit. Is that a coincidence or not? Check it out for yourself, it may be exactly what you needed.
www.elektor-labs.com/9121002536



Power supply issues

By
Thijs Beckers
(Elektor Editorial)

After several wrecked prototypes and blown up components, we started to suspect one of our power supplies might be the common denominator. A quick measurement on the supply's output seemed to eliminate the supply as being the culprit. The voltage on the output seemed to be stable and correspond to the supply's voltage display.

But when referenced to ground, something odd was going on: both the positive and negative outputs carried a negative voltage compared to ground. For example, when set to 12 volts, the positive output measured –14V and the negative output measured –26V referred to ground. In most situations this doesn't create a problem, but when this supply is used to power circuits that are referenced to ground somehow, for example when they are also connected to a PC, things start to get messy (or smoky).

So we opened up the case to determine what was wrong with this PSU. At first we couldn't see anything wrong. No black copper traces, no burned components, nothing. We started measuring the resistance between various key components and ground and soon discovered that the collectors of the output transistors seemed to be hard wired to ground. Tapping into several points of the circuit with our multimeter, we narrowed the fault down to one of the transistors not being isolated from the heat sink, which was directly bolted onto the case and therefore grounded.

With the isolating sheets clearly in view, could it be that during construction the plastic washers for isolating the collector from the screws and the heat sink got omitted? We removed the screws and we found a solder blob of substantial size seized between transistor and heat sink (of course on the last one examined). The blob had torn a hole in the isolation sheet, establishing a conducting path from collector to ground.

'Luckily' the rest of the circuit was floating with respect to ground, so no (internal!) components were damaged. However, this unit having passed the factory's Quality Testing leaves much to doubt and fear about the test methods applied.

With the solder blob removed we reassembled the device and tested it. No abnormalities were measured and the power supply now functions as expected.

Now who's going to repair those blown up prototypes… Any volunteers?

(130020)

7805 replacement *grilled*

While setting up a test with his 'switching 7805 replacement' project (November 2012 edition) for the purpose of generating some better 'scope screenshots for his webinar on the subject (webcast on November 22, 2012), designer Raymond Vermeulen noticed a slight irregularity in the circuit's behaviour. Raymond created a difficult to drive ballast for his circuit: a switching load using one 47 Ω 5 W resistor as a static load and one 6.8 Ω 10 W resistor in series with an IRF530 MOSFET to act as a switching load.

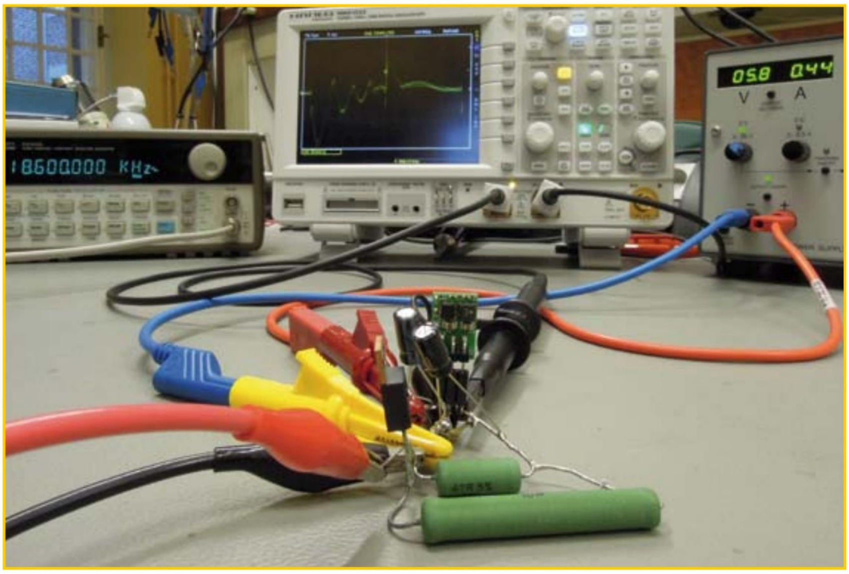
The gate of the FET was connected to a function generator set to square wave o/p — duty cycle 50%, allowing a load to be realized that was constantly switching between 106 mA and 840 mA. At a frequency of about 18.6 kHz (the resonant frequency of the output filter) the output of the 7805 replacement was stable, but a little

wiggling and ringing could be seen on the scope image (see photograph).
Of course its 'normal' regulated output voltage isn't nearly as bad as you can see in the photograph, where the input voltage had been cut down to a bare minimum of 5.8 V (power supply on the right) to test the circuit to its limits, so no worries really on the circuit's performance. Also note of the scope settings: the vertical grid is only 50 mV, so the maximum amplitude of the spike shown here is about 200 mV.
In conclusion we can — luckily! — state that it is a sound design; even in circumstances where many other designs fail and even possibly break down.

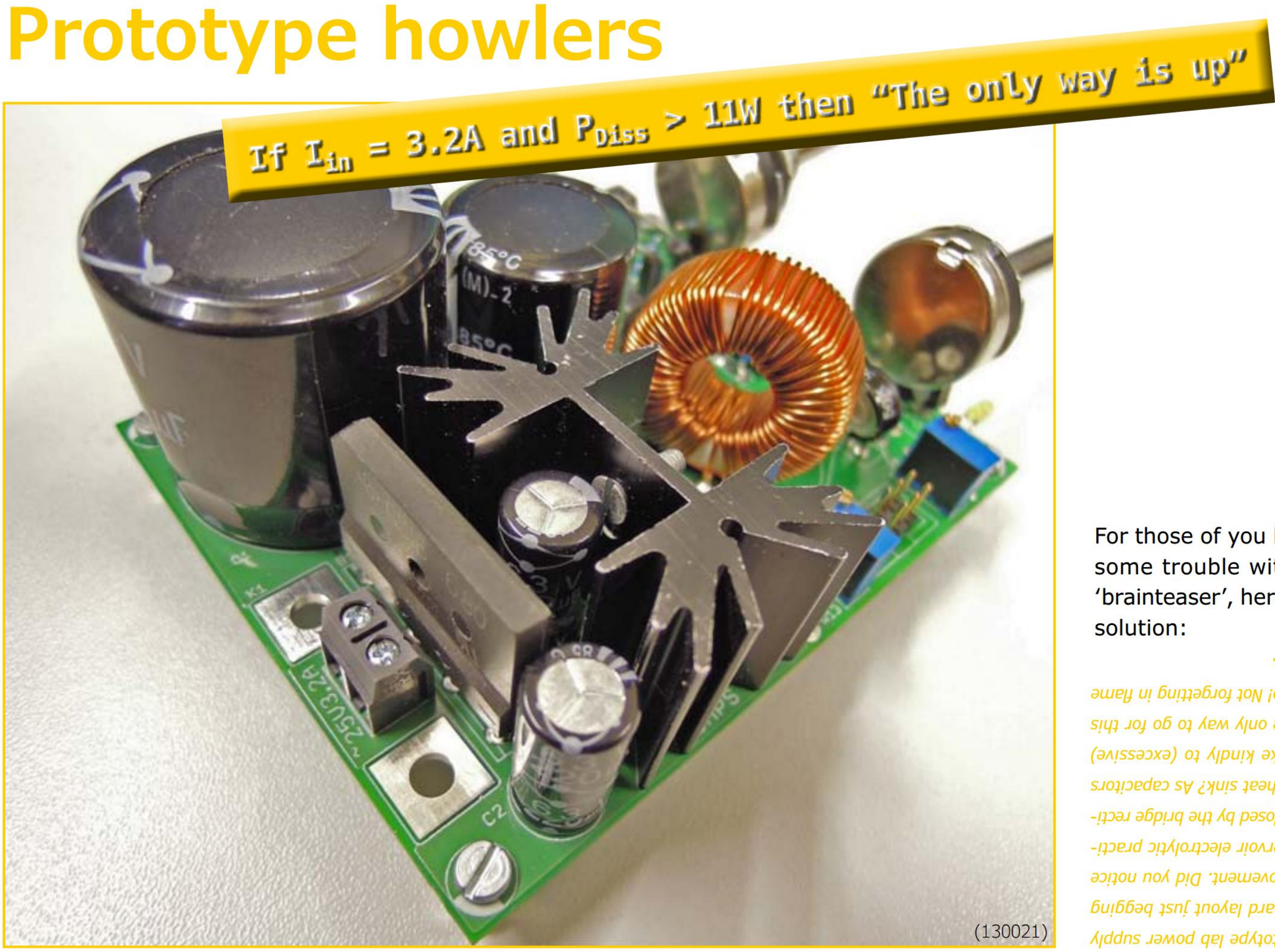
(120702)

Internet Link

www.elektor-labs.com/120212



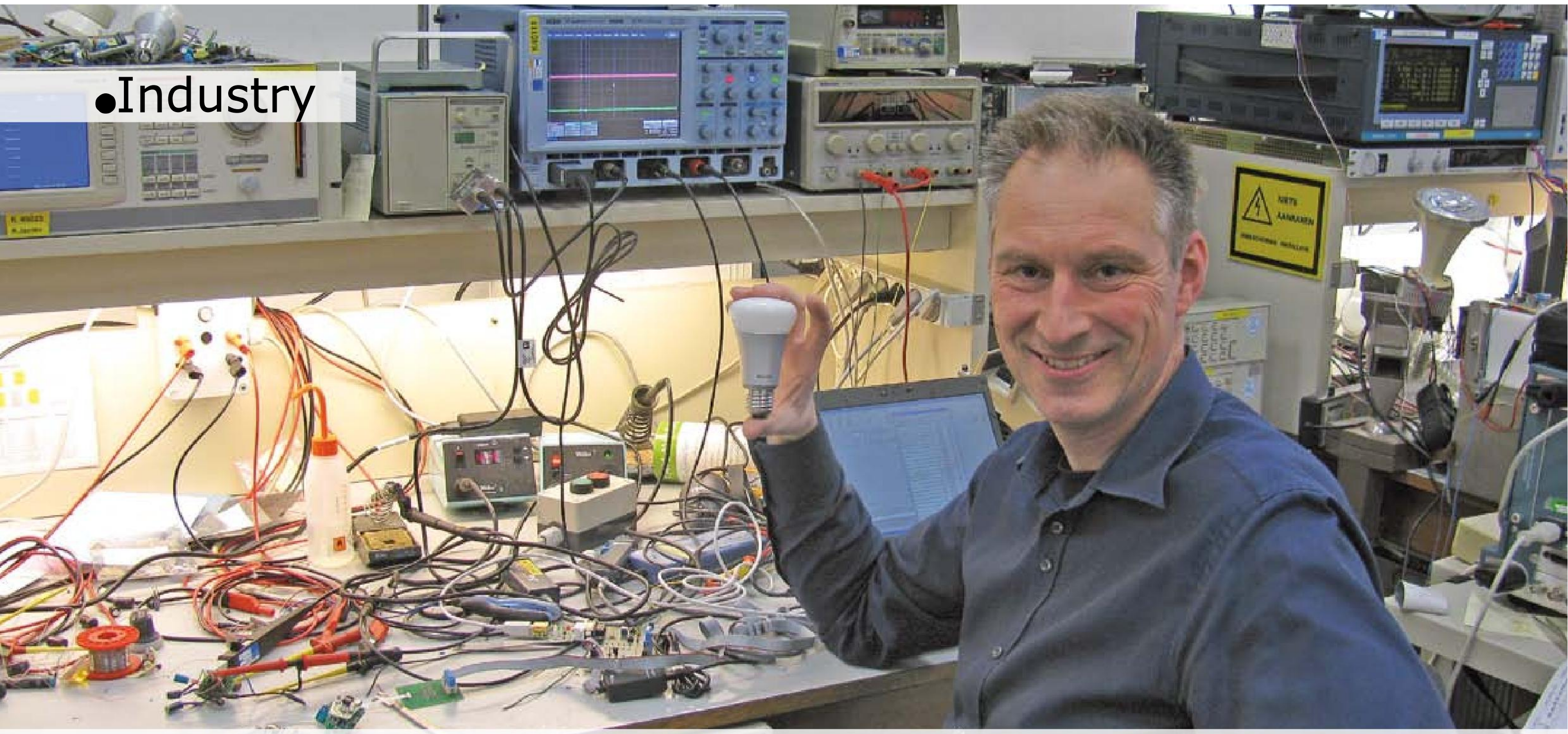
Prototype howlers



(130021)

For those of you having some trouble with this 'brainteaser', here's the solution:

This prototype lab power supply has a board layout just begging for improvement. Did you notice that reservoir electrolytic practically enclosed by the bridge rectifier and heat sink? As capacitors don't take kindly to (excessive) heat, the only way to go for this cap is Up! Not forgetting in flame & smoke.



Philips 'hue'
Behind the scenes of an innovative bulb

By **Thijs Beckers**
(Elektor Editorial)

Philips' latest lighting concept, *hue* [1], is sold via the Apple Store and intended for general use, hence its slick appearance and easy installation of the system. But what we electro-enthusiasts are interested in isn't on the outside. So after *experiencing* the *hue* during an impressive demonstration of its countless possibilities in a showroom/living room specially set up for the occasion, I sat down with one of the technical engineers, Aart Vroegop, at the Eindhoven based Philips Lighting offices to get a glimpse of what technologies are used in this ingenious design.

Cleverly named *hue* — the technical term for colour tone, and sounding not unlike "you" — Philips' latest product seems to be the first on the market to offer personalized lighting schemes in a very attractive way. Up to 50 intelligent RGB light points can be controlled via the 'Bridge' — the brains of the system, connected to LAN and accessible via Wifi and remotely via the Internet. A specially written app for the iPad controls the whole system. An app for Android devices boasting the same control possibilities — which is a lot — is also available. I have to admit, being able to control room lighting the way you can with *hue* has a much larger impact on your state of mind than one would imagine.

Philips Lighting has a high reputation to defend, so their new product must meet the high quality standard customers have come to expect. To ensure a successful design, extensive research has been done. Some 1000 prototype lamps were tested by volunteers in Shanghai, Berlin and New York — truly a global test! Aspects to evaluate included (ease of) installation, light quality and overall system impression. Engineers have been constantly battling regulations and safety demands in order to be able to deliver a product that strikes a balance between a good product and these regulations.

Inside *hue*

The light bulb is built in an aluminium case with a plastic overmold (**Figure 1**). After the electronics is in place, the whole case is filled with a potting material. This electrically isolating material is selected by its heat-conducting capacities, so any heat generated by the circuit is evenly spread

and diverted to the aluminium case where it can be transferred to the surroundings.
In order to be able to generate the required colour set — carefully researched using things like the Black Body Radiator Curve, especially important to create a decent colour temperature of white light — engineers applied a set of red, blue and lime-green Rebel LEDs from Luxeon (**Figure 2**). Several prototypes were made with a silicone 'bulb', but the final version employs a specially coated glass to ensure even diffusion of the light emitted by the LEDs.
The actual hue (colour temperature) is influenced by the temperature of the LEDs. During production each lamp is calibrated and a reference table is saved in the lamp's internal microcontroller to take this into account. This ensures different lamps to generate *exactly* the same colour when they are asked to do so.

Power Supply

Aart Vroegop and his team have been responsible for the SMPSU and LED current control parts of *hue*. **Figure 3** shows the basic schematic of the power supply and LED control circuits (we can't reveal the whole schematics due to copyright restrictions).

Figure 1. This is what makes *hue*. In the upper row the various components that make up the bulb; below, the Bridge.

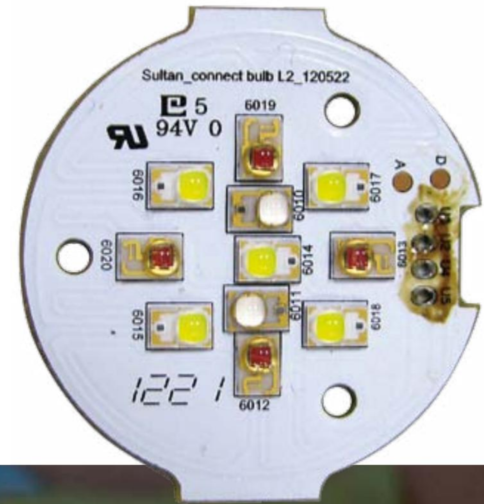
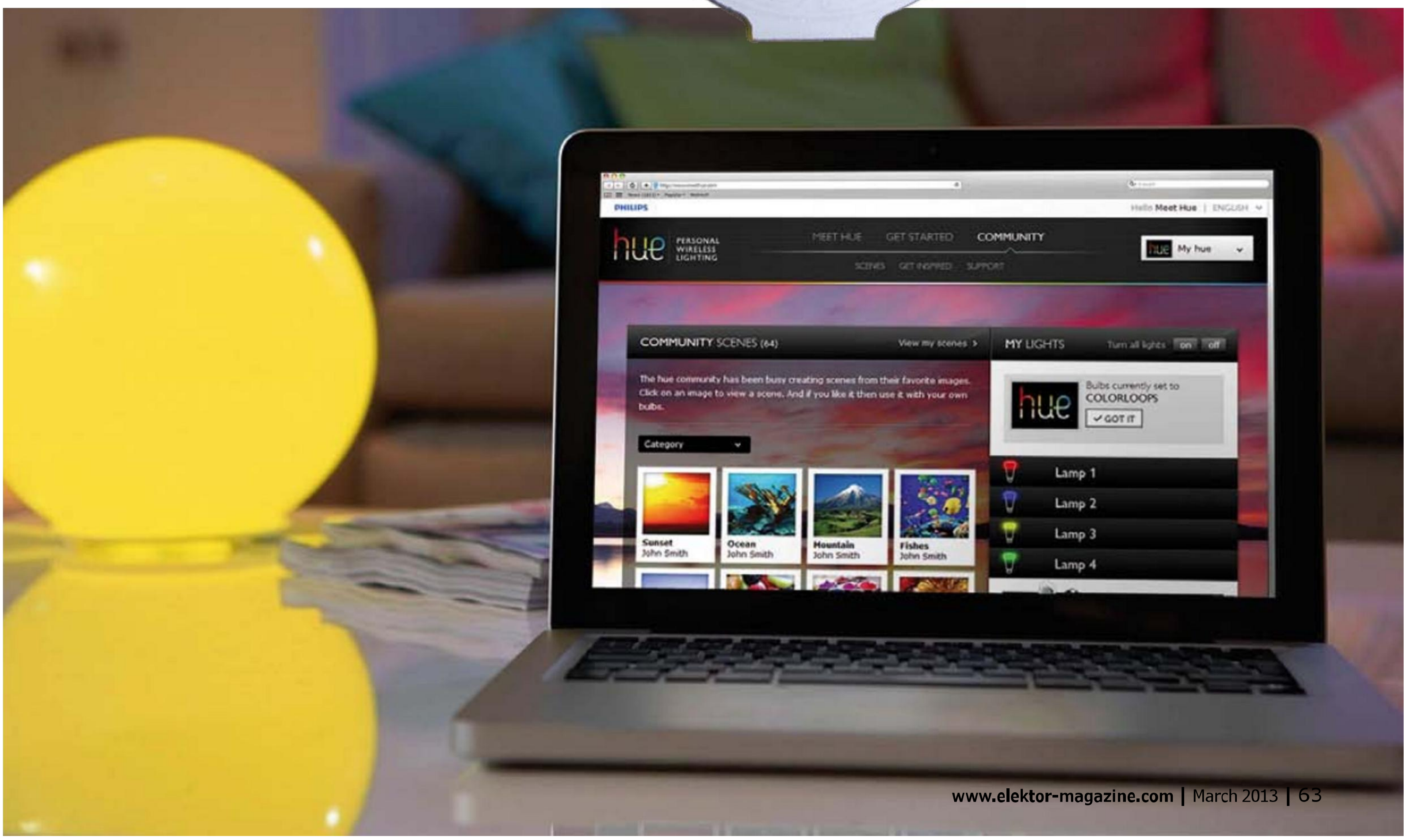


Figure 2.
Eleven Rebel LEDs make up the light source of *hue*.



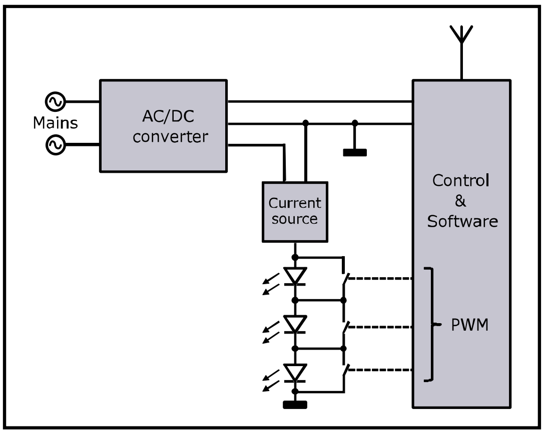


Figure 3.
Basics of the electronic innards of the bulb.

The flyback converter (**Figure 4**) converts the AC grid voltage to about 35 VDC, which is then used for powering the control circuitry and a current source driving the LEDs. The current source is a buck converter type, which has the advantage of operating without bulk capacitors at its output. Since the LEDs are series connected (all 11 of them) and PWM controlled, the generated current has to be constant while the voltage should be able to change quickly in order to guarantee a constant brightness without sudden dips or surges when a colour change command has been issued. The PWM signals generated by the microcontroller inside the bulb determine the brightness of each colour set of LEDs and thus the light colour and brightness.
There are two versions of the lamp, one suitable for 230 V and one for 120 V AC grids, the reason being that the transformer part in the flyback converter would be too bulky to fit in the case. Also, regulations regarding the permissible superimposition of disturbances on the AC grid actually prohibited the use of a single transformer suitable for all voltage regions.
The circuit *can* work when connected to a dimmer, but it can not (yet?) be dimmed by the dimmer. Operating the lamp this way is not recommended though, because the lifetime of the lamp can be reduced. But if the average voltage drops below a certain threshold, the lamp stops functioning completely.

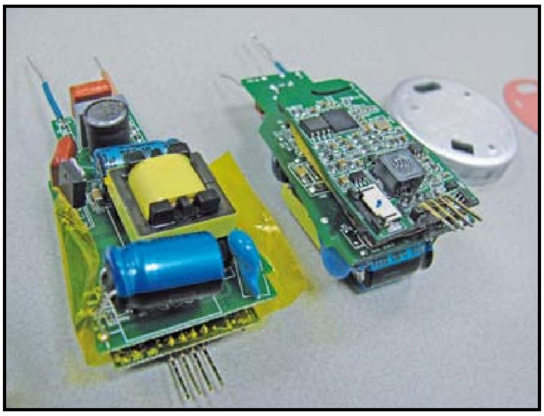


Figure 4.
At the left, the flyback converter side; at the right, the microcontroller side. More detailed photographs can be found online [2].

Built for hacking

Although Philips provides both Android and Apple apps for controlling *hue*, which are quite comprehensive and covered about 90% of the functions I could think of at the spot, development of your own app is actually encouraged, as the software is open source. Though there's not much to be hacked on the hardware side, the software is just begging for it.

The first 'disco' apps have already been spotted.
The system uses Zigbee Light Link, an open standard for interoperable and very easy-to-use consumer lighting and control products, with features like Internet ready, energy efficient, range of up to 1200 feet (400 m) outdoors and AES 128 encryption. Ever thought you would be able to give your light bulb a firmware update? With *hue* it's nothing out of the ordinary.

(120641)

Thanks to Aart Vroegop for his time and effort and the hospitality of Philips Lighting Eindhoven.

Internet Links

[1] www.meethue.com

[2] www.elektor-magazine.com/120641

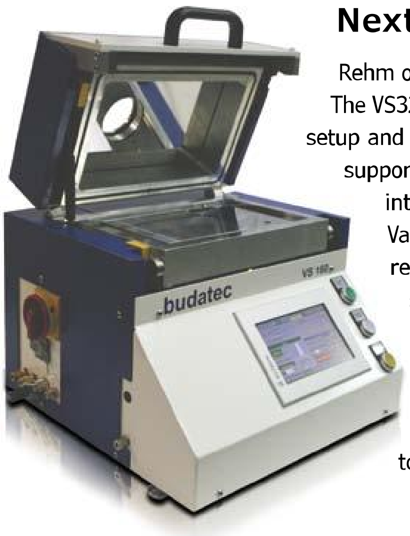
PC/104-*Plus* SBC with Intel® Atom™ N455

WinSystems announces their PPM-C393-S, a PC/104-*Plus* compatible single board computer (SBC) powered by an Intel® 1.66 GHz Atom™ processor. The PPM-C393-S blends high-integration I/O with PC/104-*Plus* expansion for a flexible yet cost-effective solution for demanding embedded applications. This combination provides designers' access to the low power performance of Intel Atom processors and to the thousands of PC/104, PC/104-*Plus*, and PCI-104 modules currently available worldwide. It is well suited for new designs and for

Extensive range of lighting components available from Avnet Abacus

Avnet Abacus is stocking a comprehensive range of power, interconnect, thermal management, device control and protection components designed specifically for solid-state lighting applications. These products complement and support the LED lighting portfolio of SILICA, another business unit of Avnet Electronics Marketing, which recently announced a pan-European franchise agreement with Philips Lumileds, a leading manufacturer of high-power LEDs. With the combined product portfolios of Avnet Abacus and SILICA, customers have access to a complete range of components for their solid-state lighting applications.
The Avnet Abacus lighting portfolio includes module-to-module, board-to-board, wire-to-board and wire-to-wire connectors with current and voltage ratings compatible with LED applications to enable space-efficient assembly and installation. These units are suitable for scalable applications such as architectural cove lighting, signage or emergency lighting.
For solid-state lighting power requirements, Avnet Abacus offers a broad range of constant-current and/or constant-voltage AC/DC modules from industry-leading manufacturers, suitable for both indoor and outdoor LED lighting applications such as street lighting, architectural lighting, signage and displays. Many products in the range offer high IP ratings and tolerances to both hot and extreme cold conditions. Available in a variety of formats including open-frame and sealed units, the product range provides high performances over a range of output voltages and currents. All of the modules come with stringent safety ratings and compliances. In addition, Avnet Abacus offers dedicated power management ICs and software to develop LED offline power supplies at universal input voltages from 85 to 265 VAC. Topologies include: boost, buck, buck-boost, and flyback for 1 W to 500 W and more power.

www.avnet-abacus.eu/lighting (130048-I)



Next-generation soldering solutions

Rehm offers three VS Series Vacuum Soldering system models. They're all small, all powerful, and all low cost. The VS320, the VS160UG, and the VS160S benchtop each feature fast heating and cooling rates with easy profile setup and editing plus data logging. The use of different gases as well as formic acid and microwave plasma is supported. All of the models offer a small footprint, and all are low cost to rapidly turn increased productivity into profitability.
Vacuum Soldering delivers a huge array of exceptional productivity and quality advantages by dramatically reducing voids in solder joints, crucial in the production of power electronics. It also supports processes such as plasma cleaning and gas exchange for advanced packaging through the controlled use of gases during the soft-soldering process at temperatures up to 450 degrees Celsius. It facilitates oxide-free and void-free connection of the chip to its substrate and allows flux free soldering and the use of solder preforms. Degassing and drying can be integrated into a single process. Vacuum Soldering delivers significantly increased yields through oxide-free processing and improving wetting to enhance solder filling.

www.rehm-group.com (130048-II)

upgrading existing applications. The PPM-C393-S' -40° to +85°C temperature operation and low power opens up applications for security, Mil/COTS, medical, transportation, data acquisition, and communications in a small, rugged, form factor proven in these industries.
The PPM-C393-S is based on Intel's single core 1.66 GHz Atom™ N455 and the ICH-8M System Controller Hub with up to 2 GB of DDR3 systems memory. The onboard I/O interface features a Gigabit Ethernet port, simultaneous CRT and LVDS flat panel video support, eight USB 2.0 ports, four serial COM ports, SATA controller, PATA controller for the CompactFlash socket, and stereo audio. The PPM-C393-S supports both PC/104 and PC/104-Plus expansion connectors to allow I/O modules to be added for even more I/O flexibility.
The PPM-C393-S requires only +5 volts and typically draws 2.5 A. It supports power savings modes which will reduce the standby current to <300 mA (S3 power state). The board is RoHS-compliant.
The PPM-C393-S supports Linux, Windows, and other x86-compatible real-time operating systems. Free drivers are available from the WinSystems' website.

www.winsystems.com (130048-V)

Differential pressure sensors support bypass configuration

Swiss sensor manufacturer Sensirion recently added new differential pressure sensors to its proven SDP600 series. The SDP601 and SDP611 sensors are differential pressure sensors specifically calibrated for measuring mass flow in a bypass configuration.

In a bypass configuration, an orifice or a linear flow restrictor is used to generate a differential pressure in a flow channel. The resulting pressure is measured over the orifice or the linear flow component. The difference be-tween the pressures before and after the orifice correlates to the volumetric flow in the channel, depending on the specific characteristics of the flow restriction component. The mass flow can therefore be calculated from the measured pressure drop (differential pressure) over the orifice.
A bypass configuration is highly suitable for applications where individually adapted flow channels are necessary or where small differential pressures must be measured with very high precision. Especially for HVAC applications, which often involve measuring large flow volumes, it is the ideal solution.
The sensors expand the broad product range of Sensirion's digital differential pressure sensors in the SDP600 series. Along with the other products in this series, they offer a digital I2C output and are fully calibrated and temperature compensated. Operating on the principle of calorimetric flow measurement, the CMOSens® differential pressure sensors achieve outstanding sensitivity and accuracy even at very low pressure differences (below 1 Pa). They also have very high long-term stability and are free from zero-point drift. Like all devices in the SDP600 series, the SDP6x1 sensors are available in two versions. The SDP601 is intended for direct threaded connection to a pressure manifold with O-ring sealing, while the SDP611 is designed for tube connection.

www.sensirion.com/en/sdp600 (130048-III)

Cypress TrueTouch™ Gen4 implements Multitouch in Fujitsu 4G smartphone

Cypress Semiconductor Corp. announced that Fujitsu Limited has selected the TrueTouch™ Gen4 touchscreen solution from Cypress to implement the touchscreen in the new Arrows V F 04E smartphone available from NTT DOCOMO. The new Fujitsu phone, which uses the Android operating systems and operates on the 4G LTE network, leverages the Gen4 solution's leading signal-to-noise ratio (SNR) to deliver highly responsive and accurate multitouch performance in any operating environment. Gen4 also provides industry-leading waterproofing capability, enabling accurate touch input and finger tracking in the presence of moisture from rain, condensation, or sweat. The phone features a dynamic 4.7-inch HD display that offers precise tracking of up to ten fingers with the Gen4 controller. The TrueTouch solution also provides a Charger Armor feature that enables mobile phones to operate in the presence of very noisy chargers. The Gen4 family also offers features that only TrueTouch can deliver, such as built-in waterproofing functionality that allows the product to meet IP-67 standards without extra sealants or shield layers.

www.cypress.com

Open Data: Hacking Democracy

By
Tessel Renzenbrink
(Elektor TTF Editor)

A group of hackers, programmers, researchers and policy makers met on the premises of the Lower House of the Dutch Parliament on September 8, 2012 to try to hack the parliamentary database. They were there on invitation of the outgoing Speaker of the Lower House, Mrs Gerdi Verbeet. The reason for the get-together was that the parliamentary database, Parlis, will be available as open data from now on. Open data is a big stride beyond open source. It is data that is freely available and can be used by everyone. There is also no fee for open data and no copyright, it is easy to find, and it is provided in machine-readable format.

Mieke van Heesewijk and Josien Pieterse are the joint founders and directors of the non-profit foundation Netwerk Democratie, a platform for democratic innovation. In cooperation with two other organizations, Hack de Overheid ("hack the government") and Open State, Netwerk Democratie organized the Apps for Democracy event. Mark Bastiaans, a researcher at the Dutch scientific research organization TNO [3], and a team of colleagues developed an application based on the parliamentary database. I talked with them about open data and how it helps to shape democracy in the twenty-first century.

The term "hacking" is used in this article in its original sense, which is "using innovation to create new applications for an existing system". The connotation of unauthorised penetration into systems, which later came be associated with the word, is not intended here.

As old as the Internet

Tessel: Is open data something new?

Mieke: Linking data sets is as old as the Internet, but the topic that's getting a lot of attention right now is open data and democracy. For a good while already, various programmers, universities, the TNO and activists have been advocating making public data open. The nice thing now is that we see growing support for open data in a broader circle, including government bodies and companies.

Josien: This is about public data. Data that is generated using public funds or data that concerns the public. The idea is that this data belongs to everybody because we have all contributed to collecting the data. This data becomes open when government bodies or companies put the data in an accessible location in a usable format. What's especially interesting is that this allows new combinations to be generated from the data streams, which leads to the creation of new knowledge and the discovery of new relationships.

Opening up the parliamentary database

Josien: Opening up Parlis, the parliamentary database, is a good example. It contains information of interest to citizens so they can keep track of the democratic process, such as the voting records of the parties, parliamentary motions and parliamentary questions. This information was already public, but it was released in PDF format. That's of little use to researchers, since it takes a lot of effort to access the information. Making the data available in machine readable form allows applications to be developed.

On invitation of Gerdi Verbeet, the former Speaker of the Lower House, Netwerk Democratie organized the Apps for Democracy event on September 8, 2012, in cooperation with other organizations. The event was held in the Parliament Building and consisted of a hackathon where programmers built applications to access the Parlis database. There were also workshops

on open data.
Mieke: A hackathon right in the Parliament Building is a world first. It's really nice to see programmers sitting in the Lower House hacking away. It also shows the nerve of the Lower House in opening their doors to hackers. Hackers are actually treated very poorly in the Netherlands, and you can see this fear of hackers with some of the political parties in the Lower House. They seem to feel that you should have as little to do with them as possible. This despite the fact that ethical hackers can be of tremendous benefit for transparency and security, particularly in this phase of democracy. The hackathon in the Lower House marked a turning point.

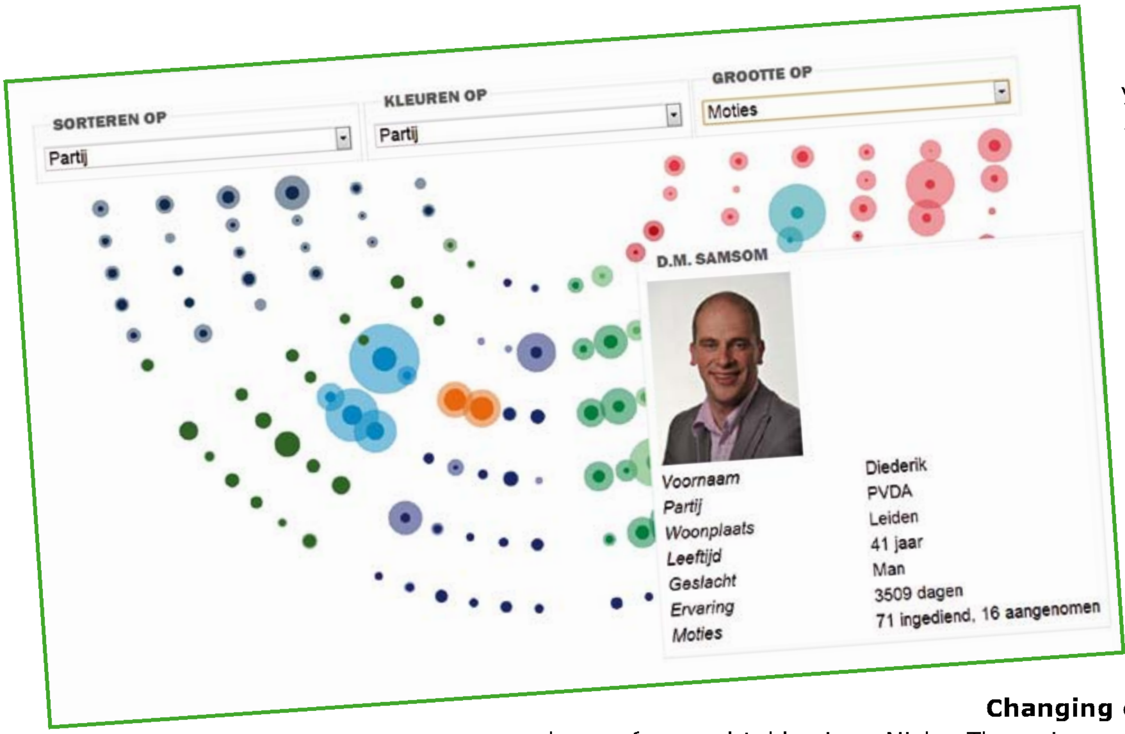
Hackathon
Mark: Together with a team of TNO employees, we looked at ways to visualize data from Parlis. We built an application for this during the hackathon. With our tool, you can arrange the members of Parliament in the Lower House in various groups based on specific parameters. For example, you can sort them by experience or by the number of submitted and approved motions. That way you can see at a glance which member of the House has submitted the most motions.
We added a data source of our own, where you can see which words are mentioned frequently in the media in connection with a particular member of Parliament. That's the nice thing about open data – that you can combine information from different sources.

Tessel: What requirements does open data have to meet so that developers can use it?
Mark: That depends on the type of developer. Tim Berners-Lee, the inventor of the World Wide Web, devised a five-star scale for open data. A single star is awarded to data with properties that meet the minimum requirements for open data: unstructured data that is published under an open license. Five stars are awarded to data that is annotated in a manner that allows it to be linked semantically to other data sets. This is called "linked open data". Data with five stars is better for a developer than data with just one star.
In theory, developers can extract information from unstructured data by using automatic analysis methods, but that is rather complicated and not very exact. A scanned PDF document is an example of unstructured data. It is actually an image, so you have to use optical character recognition (OCR) to extract the individual letters. Then you have to use text analysis to transform the letters into structured text. People have been working on OCR for a long time, and open source programs for OCR are available, but for the average developer it is easier if you have data that is directly machine readable, such as an Excel worksheet or a CSV file.
For me as a developer, it is also important that the meaning of the data is clear and that the semantics and relationships in the data are well defined. For building our application, we received a


Mieke van Heesewijk


Josien Pieterse


Mark Bastiaans



The TNO app allows the political scene in the Dutch Lower House to be visualized by dropdown selection menus. Users can also view relevant data on Members of Parliament, including experience, bills passed and rejected. Even the seating arrangement is shown with matching colours to identify the relative strength of political parties.

dump of several tables in the Parlis database. It is a relational database, which means that a column in one table is related to a column in another table. It takes a while to find out what the relationships are and what they actually mean. Fortunately, documentation about the exact relationships was provided with the data, and the organization had also generated a data model, but it still took a lot of detective work. This means that data owners must also supply metadata with the actual data, as otherwise developers won't use it and will most likely find some other data set.

Privacy and corrupted data

Tessel: Although many people have been advocating open data for a long time, government bodies were initially opposed to the idea. Now we've reached a turning point. Why is that?

Josien: It's naturally a difficult process. Lots of programmers say, "Hey, just open it up and see what happens". However, that's not how it works with the Lower House. They are very cautious about making things open.

Mieke: The Lower House is rightfully concerned about ensuring that only clean data is made open. That is the difficulty for institutions. Their position is that you can't put corrupted data sets online. They first want to get their internal information management in order, but even the best organizations can't do that.

Josien: Another reason is privacy. Letters are very interesting. Among other things, they let you see which organizations are doing lobbying, and that reveals power structures. However, this is only possible if it is clear who sent the letter, and that falls under privacy legislation. Consequently, this information is not available now, which is too bad because it is naturally very interesting information.

Mieke: Privacy and data corruption are often given as reasons, and they are legitimate concerns.

Changing circumstances

Mieke: The main reason that the idea finally got off the ground in the Netherlands is that the Ministry of Economic Affairs put their weight behind it. They realized that it's possible to make money with open data. Interesting new applications can be built by linking data sets with each other, and that is good for technological innovation and a knowledge-based economy. The parliamentary data is now hitching along for the ride.

You can see a change in attitude in the government. The government has less and less money available. That's why they need more and more help from citizens, entrepreneurs and all sorts of activists to get things done. They are looking for more collaboration with the social sector in order to achieve innovation. More and more government bodies are becoming aware of this. Ten years ago, a lot of data that was collected with public funds was farmed out to companies. Now the government realizes that there are problems because this caused the information to cut off from people who actually have a right to it. An example of this is postal code data. When the Dutch post office was privatized to form what is now TNT, it was given the right to manage the postal codes as a sort of dowry. However, there are lots of applications on the Internet that work with postal codes. Developers had to pay license fees to TNT in order to use up-to-date postal code data. The question is: who owns that data? After ten years of lobbying by a group called "Free the Postal Code", in 2012 the Dutch postal codes were finally released as open data.

A hackathon right in the Parliament Building is a world first

Josien: There was also another thing that happened: Gerdi Verbeet actually changed her position. She considers it important that citizens understand what happens in the political sector. At first her view was fairly traditional, but as a result of discussions with people who have open data high on their agenda, she came to realize that a transparent democracy also offers more opportunities for citizens to get involved in what the government does. She gradually came to see that making information open can play a part in this. That's why she started promoting it in the political world.

Tessel: Is the government on side now?
Mark: European legislation and regulations are being prepared to encourage governments to make data open. The details of the implementation are the only thing still being negotiated in Brussels. One thing is clear: things are heading in that direction top-down, but the material has not yet trickled down the chain. You see a lot of initiatives. An incredible number of hackers, enthusiasts and technically adept people are keen to get started. Things are moving at the top policy level and at the grass roots level, but we're still waiting for action at the intermediate level. This sort of change has to take place in the social realm, and that will take a bit longer than just banging a bit of code together and hacking a few systems.

(120741)

Hexadoku Puzzle with an electronic touch

No worries Elektor will run out of Hexadoku puzzles any time soon. Here's a fresh one for you to practice your combinational skills on. Enter the right numbers or letters A-F in the open boxes, find the solution in the grey boxes, send it to us and you automatically enter the prize draw for one Eurocircuits PCB voucher and three Elektor book 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 grey boxes**.

Solve Hexadoku and win!

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

Participate!

Before April 1, 2013,

send your solution (the numbers in the grey boxes) to

www.elektor.com/hexadoku

Prize winners

The solution of the December 2012 Hexadoku is: **621BA**. The Elektor £80.00 Voucher has been awarded to Manuel Aleejandro Masip Ruera (Spain). The Elektor £40.00 Vouchers have been awarded to Dominique Jacobs (Germany), Claude Ghyselen (France), and Reto Strub (Switzerland).

Congratulations everyone!

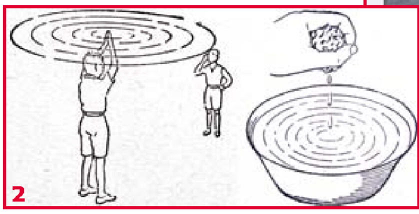
	F		8					9				7	4		B
0		9		F	5			B			2	D			3
	3	D	E		2						F			9	
4	A			6	7	9		8	3		E	C			
A		C			B	1	3							D	6
		3	2	4		C			8	0		1	9		E
					E				9				2		
E			6			2		1	7						
						3	7		A			B			1
		6				B				C					
8		7	D		9	5			E		1	3	C		
9	C							3	B	5			D		4
			3	D		6	9		4	E	7			2	F
	2			A	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

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

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

1

3



2

'Radiomann' Audion Kit (ca. 1956)

When boys' toys were educational

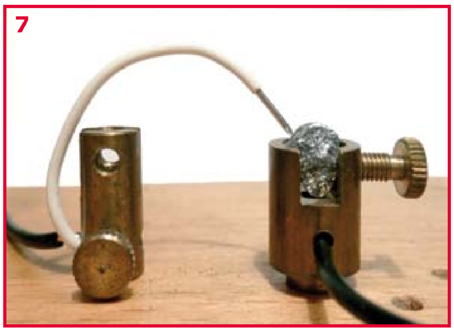
By **Peter Beil**
(Germany)

Do you remember back to your youth, when you took your first steps in the world of electronics? For me, and I guess many others, radio was a source of utter fascination back in the early 1950s. Making your own receiver was virtually impossible, because in my country at least mail order component dealers were few and far between.

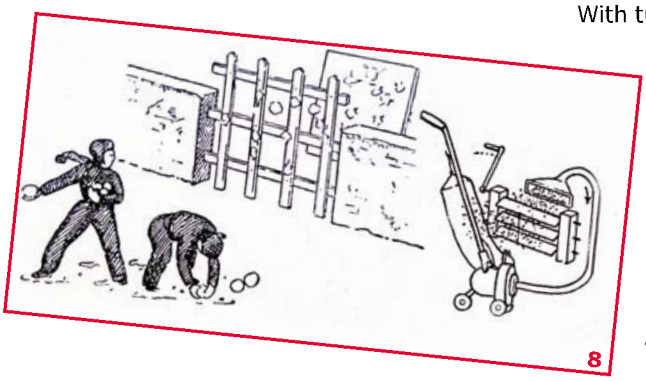
Filling this gap in the market came the so called 'Experimenter Set' offered by the Kosmos publishing house in Stuttgart, Germany, which opened a door to the electronics of the time for interested juveniles and budding engineers. And even if its visual look-and-feel (**Figure 1**) was a little austere, a wealth of 80 experiments was nevertheless possible with this outfit.

It's worth remembering that semiconductor technology was still in its infancy at this time and the vacuum tube dominated the majority of electronic circuits. With tubes one could not only amplify but also control, regulate or rectify. Television already existed at that time but held practically no significance for hobbyists or indeed boys.

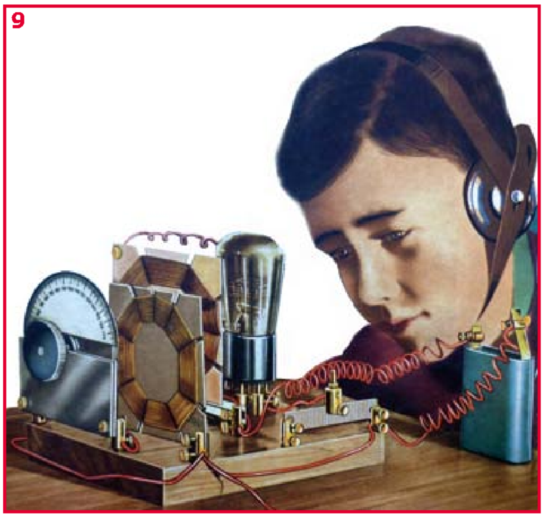
Radiomann (Radioman in English) provided an almost fun and games method of gaining an insight into wave theory, audio and radio frequency or vacuum tube technology (**Figure 2**). Everything was made wonderfully simple: the capacitor was just a piece of plastic with a piece of metal foil glued back and front (**Figure 3**), and you made a resistor yourself using a thick pencil stroke on a piece of card (**Figure 4**). I have since checked the value of that



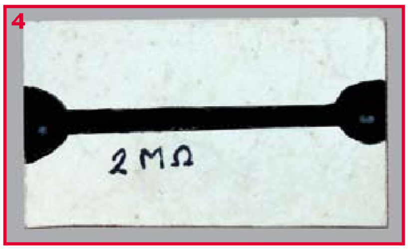
7

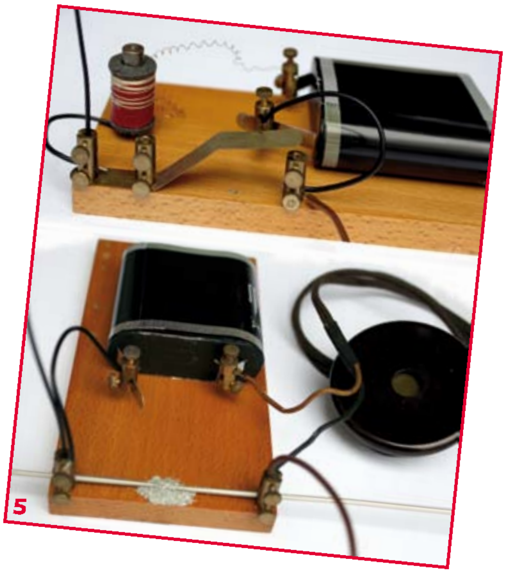


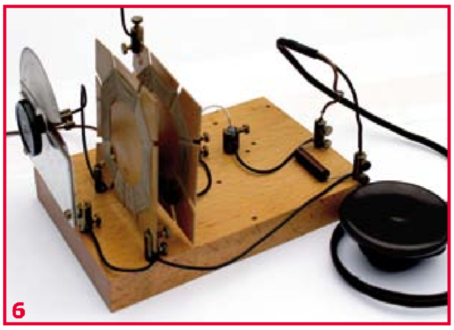
8



9


4


5


6

'resistor' and it was almost exactly two megohms. It was painted with black drawing ink and 'fine tuned' with a pencil marking over the top.

The experiments were built on a wooden breadboard that used plug-in brass clips that you placed in pre-drilled holes. In this way you learnt in the 'coherer' example what induction was, and produced your first 'wireless' transmission (**Figure 5**). For people unfamiliar with the coherer, this was a simple indicator of electromagnetic waves, usually consisting of two electrodes and iron filings in a glass tube, but 'tubeless' here.

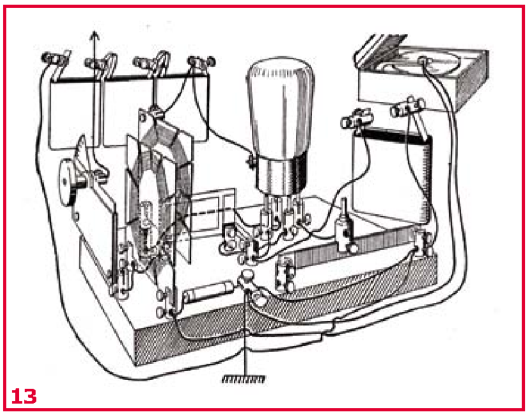
In those days a 'crystal detector' with negative feedback emerged as the appropriate 'receiving device' (**Figure 6**). The germanium diode was unknown then, and accordingly we built a proper 'semiconductor' out of a galena crystal and a sharply pointed piece of wire, experiencing straightaway the inevitable problem of finding the optimum junction position ('sweet spot') (**Figure 7**). Close to a medium wave transmitter you could even drive a small loudspeaker with this detector!

Initiation into the secrets of the vacuum tube took place on the basis of exercises that were entirely comprehensible, even for juveniles. So the operation of the anode (a.k.a. plate) was compared to a vacuum cleaner; electron flow on the grid to snowballs thrown through a grating and/or with an adjustable sunblind or louver (**Figure 8**).

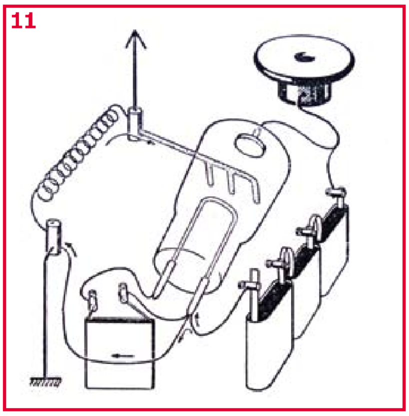
My vacuum tube has unfortunately gone missing during the last 60 years. All that remains is the illustration on the box lid (**Figure 9**). Originally it would have been a type RE074d from Telefunken, a so-called space charge grid tube. Once this became no longer available in those post-war years it was substituted by the custom-designed type DM300 (**Figure 10**). As this was actually no longer a space charge grid tube, we cunningly made a normal tetrode out of it by reversing the grid connections. In these applications it worked without any problems, despite the (technically unfavourable) direct current filament voltage.

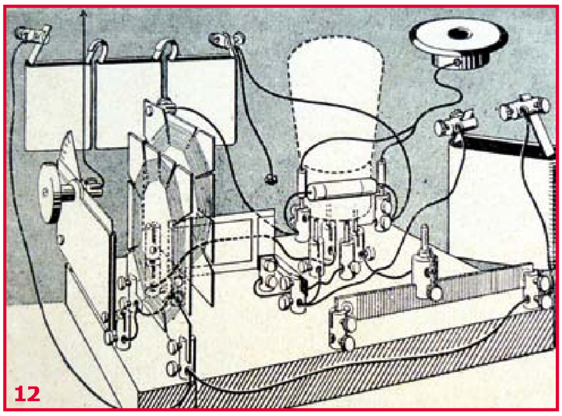
The vacuum tube was a pattern with an unusual filament voltage of 3.5–4 volts and a double grid. The plate voltage used was a harmless 12–20 volts. Incidentally this was feasible only because the second grid was connected to the plate voltage. The whole business was powered by (then commonplace) 4.5-volt flat 'lantern' batteries (zinc-carbon; IEC 3R12) (**Figure 11**).

The experiments culminated in the construction of an 'audion'


13


10


11


12

with reaction, which provided pretty good selectivity (**Figure 12**). By connecting up the domestic record player (if you had one), and the receiver 'reverse-connected', you could transmit to your broadcast receiver over an awesome distance of around 10 feet (**Figure 13**). Apparently some mischief also took place, as this experiment was not included with follow-up models.

The person who conceived this and other 'experimenter sets' to was the honorary Dr Wilhelm Fröhlich, by profession a teacher at a school in the Lake Constance district. His original wish was to bring technology closer to his students with simple physics class experiments.

During the 1960s the Radiomann was updated to an extent with an EF 89 (6DA6) pentode tube and a transistor. Subsequently the well-known electronics writer and Elektor contributor Burkhard Kainka adapted the set for more sophisticated technologies.

The model described here dates from around 1956, and at the time cost the equivalent of $6.76 ($57 today) plus $2.73 ($23 today) for the tube, which was sold separately. In those days, that was a substantial amount of money, requiring much arm-twisting on my part but providing many hours of pleasure for me. Today the Franckh-Kosmos publishing company is still very much at the cutting edge of learning and experimentation materials right up to the Arduino microcontroller platform.

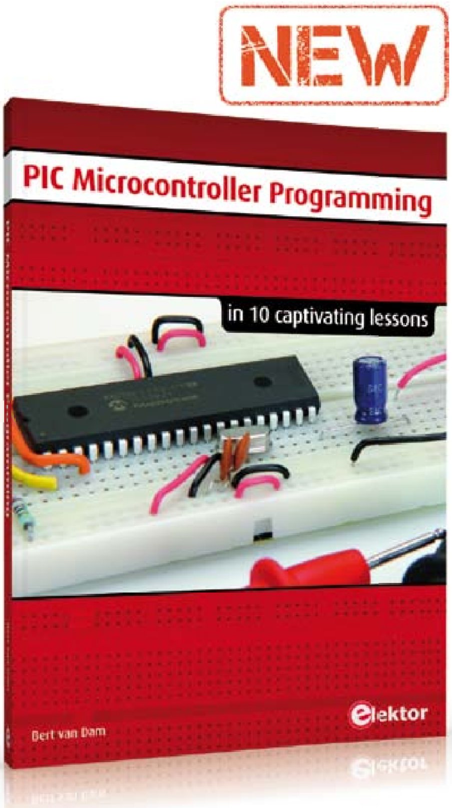
Even today there remains plenty of information about the old Radiomann on the Web. Type "Kosmos Radiomann" into your favourite search engine, you will be amazed.

(120650)

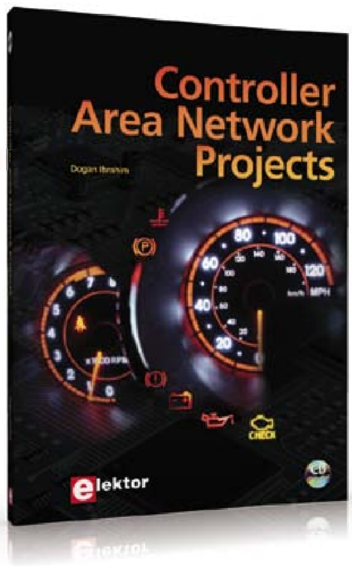
Original drawings reproduced with the kind permission of Franckh-Kosmos publishing company, Stuttgart. Photos by the author.

Retronics is a monthly section covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com

•Store

1 

2 

3 

4 

Limited Time Offer for GREEN and GOLD Members!
14% DISCOUNT
www.elektor.com/march

10 captivating lessons

1 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 • US $47.60

More than 75,000 components

2 CD Elektor's Components Database 7

This CD-ROM gives you easy access to design data for over 11,100 ICs, 37,000 transistors, FETs, thyristors and triacs, 25,100 diodes and 2,000 optocouplers. The program package consists of eight databanks covering ICs, transistors, diodes and optocouplers. A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data.

ISBN 978-90-5381-298-3 • £24.90 • US $40.20

Free mikroC compiler CD-ROM included

3 Controller Area Network Projects

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters.

260 pages • ISBN 978-1-907920-04-2
£29.50 • US $47.60

Meet BOB

4 FT232R USB/ Serial Bridge/BOB

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

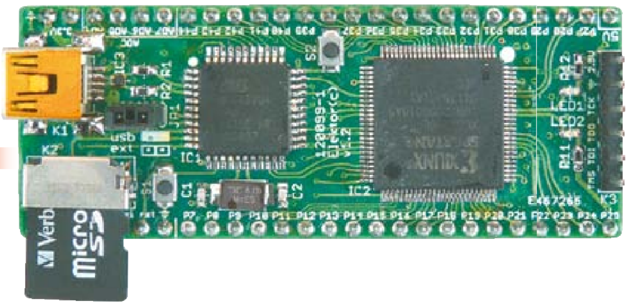
PCB, assembled and tested
Art.# 110553-91 • £12.90 • US $20.90

LabWorX 2

5 Mastering Surface Mount Technology

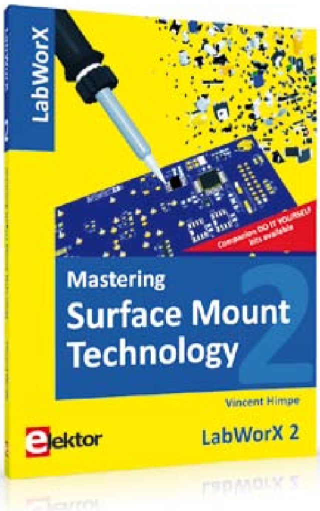
This book takes you on a crash course in techniques, tips and know-how to successfully introduce surface mount technology in your workflow. Even if you are on a budget you too can jumpstart your designs with advanced fine pitch parts. Besides explaining methodology and equipment, attention is given to SMT parts technologies and soldering methods. Many practical tips and tricks are disclosed that bring

6

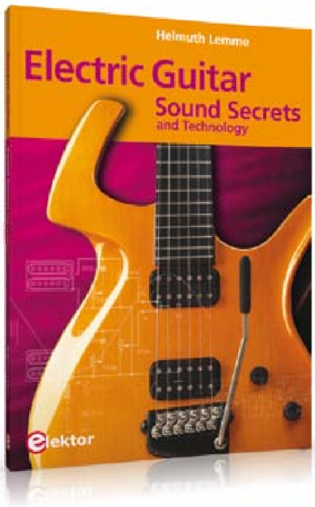


8

5



7



9

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 • US $47.60

Taming the Beast

6 FPGA Development Board

FPGAs are unquestionably among the most versatile but complex components in modern-day electronics. An FPGA contains a maze of gates and other circuit elements that can be used to put together your own digital circuit on a chip. This FPGA development board (designed in the Elektor Labs) shows how easy it is for any electronics enthusiast, whether professional or amateur, to work with these programmable logic devices.

Module, ready build and tested
Art.# 120099-91
See www.elektor.com/fpgaboard

Sound Secrets and Technology

7 Electric Guitar

What would today's rock and pop music be without electric lead and bass guitars? These instruments have been setting the tone for more than forty years. Their underlying sound is determined largely by their electrical components. But, how do they actually work? This book answers many questions simply, in an easily-understandable manner. For the interested musician (and others), this book unveils, in a simple and well-grounded way, what have, until now, been regarded as manufacturer secrets. The examination explores deep within the guitar, including pickups and electrical environment, so that guitar electronics are no longer considered highly secret. With a few deft interventions, many instruments can be rendered more versatile and made to sound a lot better – in the most cost-effective manner.

287 pages • ISBN 978-1-907920-13-4
£29.50 • US $47.60

Programming step-by-step

8 Android Apps

This book is an introduction to programming apps for Android devices. The operation of the Android system is explained in a step by step way, aiming to show how personal applications can be programmed. A wide variety of applications is presented based on a solid number of hands-on examples, covering anything from simple math programs, reading sensors and GPS data, right up to programming for advanced Internet applications. Besides writing applications in the Java programming language, this book also explains how apps can be programmed using Javascript or PHP scripts. When it comes to personalizing your smartphone you should not feel limited to off the shelf applications because creating your own apps and programming Android devices is easier than you think!

244 pages • ISBN 978-1-907920-15-8
£34.95 • US $56.40

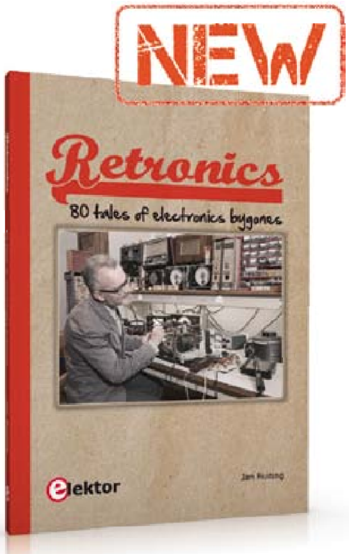
Elektor Linux Board

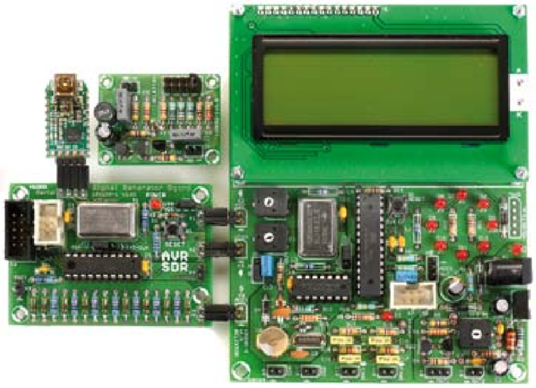
9 Embedded Linux Made Easy

Today Linux can be found running on all sorts of devices, even coffee machines. Many electronics enthusiasts will be keen to use Linux as the basis of a new microcontroller project, but the apparent complexity of the operating system and the high price of development boards has been a hurdle. Here Elektor solves both these problems, with a beginners' course accompanied by a compact and inexpensive populated and tested circuit board. This board includes everything necessary for a modern embedded project: a USB interface, an SD card connection and various other expansion options. It is also easy to hook the board up to an Ethernet network.

Populated and tested Elektor Linux Board
Art.# 120026-91 • £57.80 • US $93.30

•Store





The world's first book with NFC technology integrated inside

10 Catch the Sun

The oldest known contactless connectivity technology dates back 2000 years to the Han dynasty in China. In that era, the Kongming lantern was invented: a small hot air balloon used primarily for transmitting military signals. The Kongming balloons have today been replaced by chips. Near Field Communication, or NFC, provides wireless connectivity over short distances based on semiconductor technology. This book links both technologies together.

Catch the Sun is the world's first book with NFC semiconductor technology integrated inside, while the content of this high-tech book is about the beautiful magic of low-tech ballooning. The book has multiple NFC chips inside that allow the book to connect to the internet, simply by touching an NFC-hotspot in the book with your NFC-enabled smartphone or tablet.

128 pages • ISBN 978-9-07545-861-9
£35.50 • US $57.50

80 tales of electronics bygones

11 Retronics

Quite unintentionally a one-page story on an old Heathkit tube tester in the December 2004 edition of Elektor magazine spawned dozens of 'Retronics' tales appearing with a monthly cadence, and attracting a steady flow of reader feedback and contributions to the series. This book is a compilation of about 80 Retronics installments published between 2004 and 2012. The stories cover vintage test equipment, prehistoric computers, long forgotten components, and Elektor blockbuster projects, all aiming to make engineers smile, sit up, object, drool, or experience a whiff of nostalgia. Although vastly diff erent in subject matter, all tales in the book are told with personal gusto because Retronics is about sentiment in electronics engineering, construction and repair, be it to reminisce about a 1960s Tektronix scope with a cleaning lady as a feature, or a 1928 PanSanitor box for dubious medical use.

193 pages • ISBN 978-1-907920-18-9
£24.80 • US $40.00

A whole year of Elektor magazine onto a single disk

12 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.

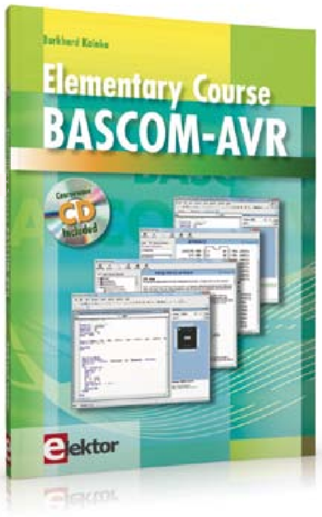
ISBN 978-90-5381-273-0 • £23.50 • US $37.90

Package Deal: 12% off

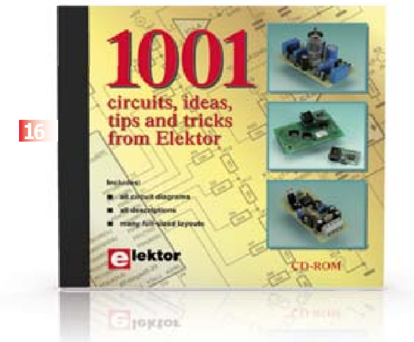
13 AVR Software Defined Radio

This package consists of the three boards associated with the AVR Software Defined Radio articles series in Elektor, which is built around practical experiments. The first board, which includes an ATtiny2313, a 20 MHz oscillator and an R-2R DAC, will be used to make a signal generator. The second board will fish signals out of the ether. It contains all the hardware needed to make a digital software defined radio (SDR), with an RS-232 interface, an LCD panel, and a 20 MHz VCXO (voltage-controlled crystal oscillator), which can be locked to a reference signal. The third board provides an active ferrite antenna.

Signal Generator + Universal Receiver + Active Antenna: PCBs and all components + USB-FT232R breakout-board
Art.# 100182-72 • £99.90 • US $133.00

15


17


16


18


110 issues, more than 2,100 articles

14 DVD Elektor 1990 through 1999

This DVD-ROM contains the full range of 1990-1999 volumes (all 110 issues) of Elektor Electronics magazine (PDF). The more than 2,100 separate articles have been classified chronologically by their dates of publication (month/year), but are also listed alphabetically by topic. A comprehensive index enables you to search the entire DVD. What's more, this DVD also contains the entire 'The Elektor Datasheet Collection 1...5' CD-ROM series.

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

Free Software CD-ROM included

15 Elementary Course BASCOM-AVR

The Atmel AVR family of microcontrollers are extremely versatile and widely used. In Elektor magazine we have already published many interesting applications employing an ATmega or ATtiny microcontroller. The majority of these projects perform a particular function. In this book we focus more on the software aspects. Using lots of practical examples we show how, using BASCOM, you can quickly get your own design ideas up and running in silicon.

224 pages • ISBN 978-1-907920-11-0
£34.95 • US $56.40

Circuits, ideas, tips and tricks

16 CD 1001 Circuits

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Circuits issues 2001-2010 of Elektor, supplemented with various other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & modelling, home & garden, high frequency, power supply, robotics, test & measurement and of course a section miscellaneous..

ISBN 978-1-907920-06-6 • £34.50 • US $55.70

140 Minutes video presentation and more

17 DVD Feedback in Audio Amplifiers

In this Masterclass we address several aspects of feedback in audio amplifiers. The focus of this Masterclass, although not entirely math-free, is on providing insight and understanding of the issues involved. Presenter Jan Didden provides a clear overview of the benefits that can be obtained by feedback and its sibling, error correction; but also of its limitations and disadvantages. Recommended to audio designers and serious audio hobbyists!

ISBN 978-907920-16-5 • £24.90 • US $40.20

Counter for alpha, beta and gamma radiation

18 Improved Radiation Meter

This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive samples. The photodiode has a smaller sensitive area than a Geiger-Müller tube and so has a lower background count rate, which in turn means that the radiation from a small sample is easier to detect against the background. A further advantage of a semiconductor sensor is that is offers the possibility of measuring the energy of each particle.

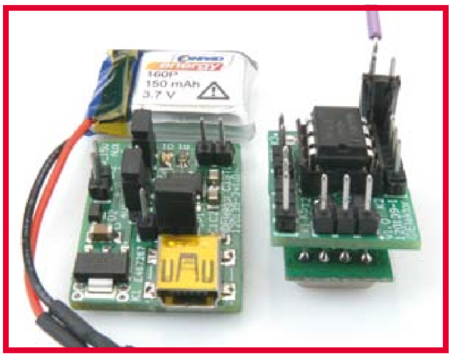
Kit of parts incl. display and programmed controller
Art.# 110538-71 • £35.50 • $57.30

Further information and ordering:

www.elektor.com/store

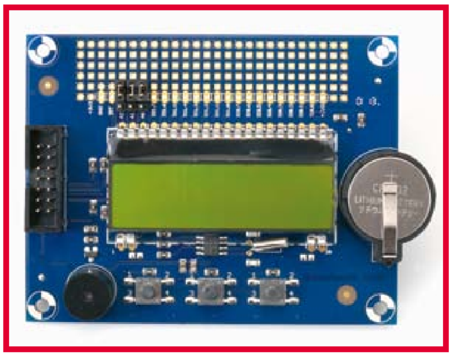
Elektor
78 York Street
London - W1H 1DP
United Kingdom
Tel.: +44 20 7692 8344
Email: order@elektor.com

NEXT MONTH IN ELEKTOR MAGAZINE



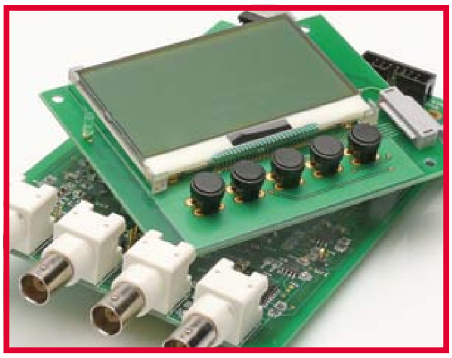
Lost Model Finder

Nope, this circuit does not help you locate Naomi Campbell or Kate Moss. We're talking about a handy device to help you find your crashed model aeroplane in a field. A small transmitter is housed in the aircraft, and its beacon signals are picked up by a receiver using a directional antenna. Thanks to the use of commercial, type approved ISM band transmitter and receiver modules, the circuit is easy to build.



Embedded Linux Extension Board

By now the Elektor Linux Board has captured thousands of Elektor readers. In the next edition we present an extension board that contains some nifty hardware add-ons. There are three push buttons, an LCD with 2 x 16 characters, a buzzer, a real-time clock with battery and a port expansion with 16 additional digital inputs/outputs. Moreover, there is also a 0.6 x 2.4 inch (1.5 x 6 cm) prototyping area to build your personal extensions.



500 ppm LCR Meter Construction

We appreciate that the schematics and various design considerations of the brand new 500 ppm LCR Meter launched in this edition may take some time to digest properly, say, a month. So, next month we cheerfully continue with the construction of the meter. A precision instrument like this one of course requires extra care in design and the quality of the components used.

Note: due to lack of space the Thermo Book project could not be included in the current edition.

Article titles and magazine contents subject to change; please check www.elektor-magazine.com

Elektor USA April 2013 edition published March 19, 2013.