

WHAT MICRO FOR ME?

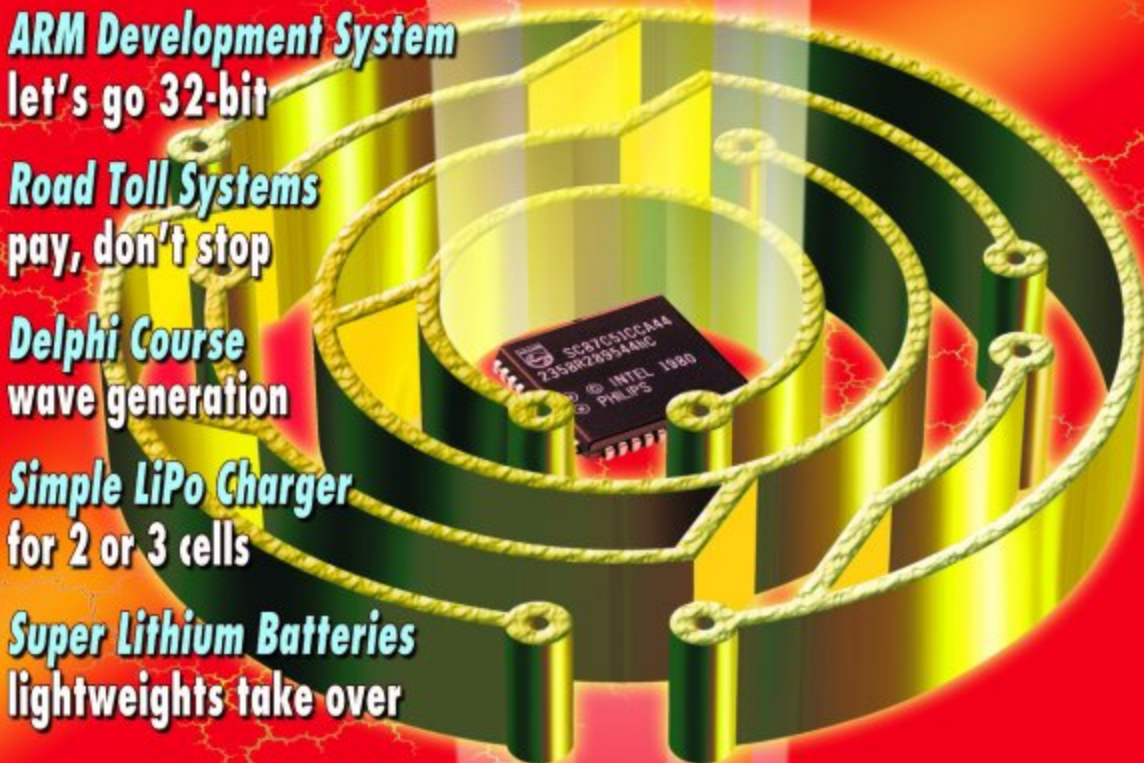
ARM Development System
let's go 32-bit

Road Toll Systems
pay, don't stop

Delphi Course
wave generation

Simple LiPo Charger
for 2 or 3 cells

Super Lithium Batteries
lightweights take over



SC-Analyser 2005
test transistors



Taken apart!
Mac Mini:
UNIX in a box



That's mIPS, not MIPS

This issue has a focus on microcontroller technology which has been a major subject in Elektor Electronics for 20-odd years already, not forgetting, of course, test and measurement, audio, RF, power supplies and what have you. With so much of "all the latest technology" being pushed on, across (and eventually off) my desk all the time, it is good to take a deep breath at times and reflect a moment on where it all came from. In this very issue we present 'ARMee', our first 32-bit microcontroller development system and it is no coincidence that this beast is solidly based on a plethora of support tools, some free, some commercial, but always just a few clicks away. What a contrast with the SC/MP ('scamp') Computer sprawling across the grim looking pages of Elektor in 1978. Youngsters will laugh their heads off: jokes about milli-instructions per second (mIPS); programming an 8-bit micro bit-by-bit, yes that's flipping eight miniature rocker switches to, say, 0101 0111 in order to prepare one byte '57' for storage in memory, while also knowing what the associated 'opcode' like MOVX or similar gibberish is supposed to do. Error reports, FAQs, whad'ye mean?

No wonder older programmers are still able, beer in hand, day or night, to tell you the bit configuration for any hexadecimal word you care to throw at them. With today's explosion of computing power, it all looks like futile knowledge. And yet, I doubt if those readers back in 1978 diligently assembling and programming their SC/MPs 'the hard way' got less fun and satisfaction from their project than anyone in the Internet age capable of pushing a few PC keyboard buttons and flashing a credit card. There is a microcontroller heritage and the investment of 27 years ago did not go to waste.

Jan Buiting – Editor



Volume 31, Number 342, April 2005

ISSN 0268/4519

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

Elektor Electronics is produced and published by Elektor Electronics (Publishing), P.O. Box 190, Tunbridge Wells TN5 7WY, England. Tel.: (+44) (0)1580 200657, fax: (+44) (0)1580 200616. Email: sales@elektor-electronics.co.uk.

The magazine is available from newsagents, bookshops and electronics retail outlets, or on subscription.

Elektor Electronics is published 11 times a year with a double issue for July & August.

Under the name *Elektor* and *Elektuur*, the magazine is also published in French, German and Dutch. Together with franchised editions the magazine is on circulation in more than 50 countries.

International Editor: Mat Heffels (m.heffels@segment.nl)

Editor: Jan Buiting (editor@elektor-electronics.co.uk)

International editorial staff: Harry Baggen, Thijs Beckers, Rolf Gerstendorf, Ernst Krempelsauer, Guy Raedersdorf.

Design staff: Karel Walraven (head of design), Ton Giesberts, Paul Goossens, Luc Lemmens (techdept@segment.nl)

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

Graphic design / DTP: Ton Gulikers, Giel Dols

Managing Director / Publisher: Paul Snakkers

Circulation Control: Margriet Debeij (m.debeij@segment.nl)

Subscriptions

Worldwide Subscription Service Ltd., Unit 4, Gibbs Reed Farm, Pashley Road, Ticehurst TN5 7HE, England. Telephone: (+44) (0)1580 200657, Fax: (+44) (0)1580 200616 Email: wwss@wwss.demon.co.uk Rates and terms are given on the Subscription Order Form

Head Office

Segment b.v. P.O. Box 75 NL-6190-AB Beek The Netherlands Telephone: (+31) 46 4389444, Fax: (+31) 46 4370161

Distribution: Seymour, 86 Newman Street, London W1P 3LD, England

UK Advertising

Huson International Media, Cambridge House, Gogmore Lane, Chertsey, Surrey KT16 9AP, England.

Telephone: +44 (0)1932 564999, Fax: +44 (0)1932 564998

Email: gerryrb@husonmedia.com

Internet: www.husonmedia.com

Advertising rates and terms available on request.

International Advertising

Klaas Caldenhoven, address as Head Office

Email: advertenties@elektuur.nl

Advertising rates and terms available on request.

Copyright Notice

The circuits described in this magazine are for domestic use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, software carriers and article texts published in our books and magazines (other than third-party advertisements) are copyright Segment. b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning or recording, in whole or in part without prior written permission from the Publishers. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature.

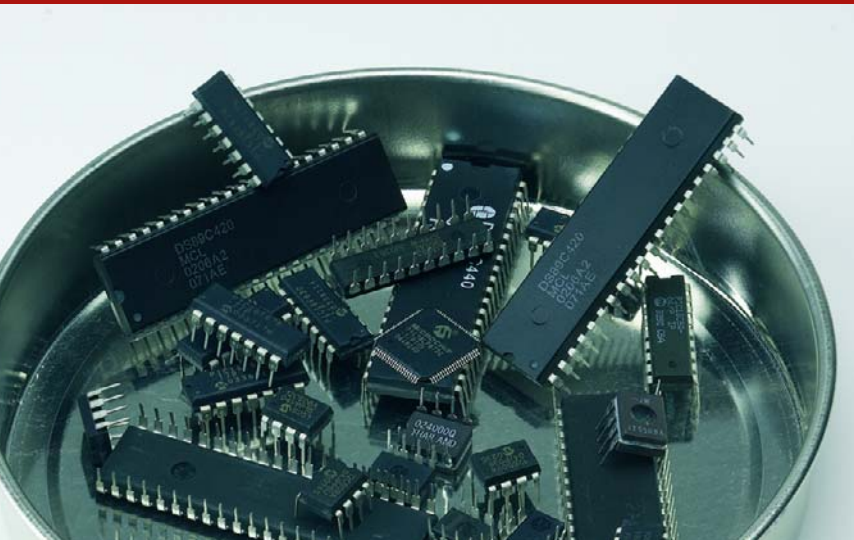
Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection.

The submission of designs or articles implies permission to the Publishers to alter the text and design, and to use the contents in other Segment publications and activities. The Publishers cannot guarantee to return any material submitted to them.

© Segment b.v. 2005

Printed in the Netherlands

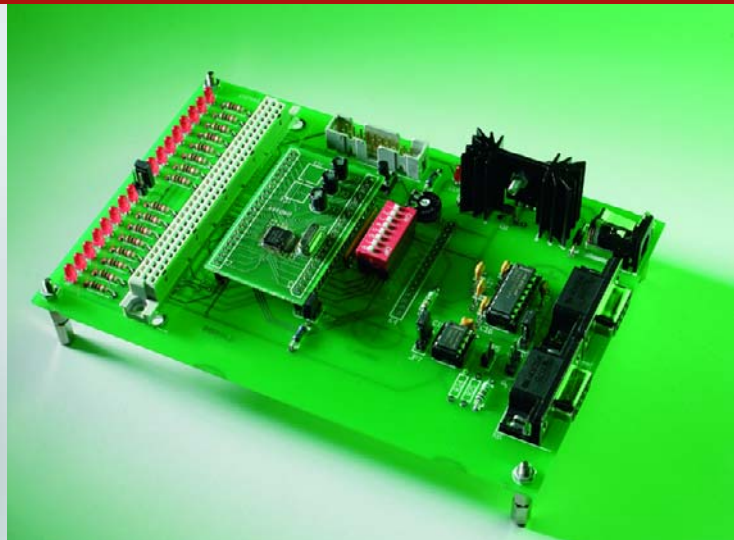
12



Made-to-Measure Processing Power

In the design stage for microcontroller circuits, professional and amateur designers often wrestle with the question of which microcontroller to use, since thousands of different types are commercially available. This article is intended to help guide you through the selection maze.

20



LPC210x 'ARMee' Development System (2)

Bored with PICs, AVRs and 8051s everyone else is doing? Last month we covered 32-bit ARM microcontrollers and the undeniable star from all the devices discussed in that article was the Philips LPC210x. This month the real thing is upon us: enter **ARMee**, an incredibly powerful ARM development board you can build and program yourself.

Informative Articles

- 12 Made-to-Measure Processing Power
- 42 Working with 32-bit Micros
(Keil ARM review)
- 44 Delphi for Electronic Engineers (4)
- 52 Super Lithium Batteries
- 70 E-Online: Pick your Micro
- 72 New Road Toll Systems
- 77 retronics: Elektor SC/MP Computer (1978)

Regulars

- 5 Foreword & Colophon
- 8 Mailbox
- 10 News: Mac Mini
- 67 New Products
- 78 Quizz'away
- 82 Readers Services
- 84 Sneak Preview
- 84 Index of Advertisers

CONTENTS

Volume 31
April 2005
no. 342

34



52



72



SC Analyser 2005

Super Lithium Batteries

New Road Toll Systems

Almost every electronic circuit contains bipolar transistors, FETs or diodes. Most electronics hobbyists have a supply of such components removed from old circuit boards. A tester that can be used to sort out the leads and measure the characteristics is thus a handy tool.

Modern mobile electronic gadgets demand ever more portable and powerful energy storage devices. Research by battery manufacturers all over the world moves on apace; new developments supplant old technologies, and are themselves almost instantly superseded by even newer technologies. The field is in a state of constant flux with new surprises every day.

Germany's sophisticated new network using GPS, GSM and infrared technology has only been up and running since the beginning of the year but the Department of Transport and UK Customs and Excise are watching closely and have announced their intention to introduce a similar system in the UK.

Construction Projects

- 20 LPC210x 'ARMee' Development System (2)
- 34 SC Analyser 2005
- 50 Brachion Detector
- 60 Simple LiPo Battery Charger
- 68 start here:
 - Spark Transmitter – Flashing Light with
 - Dimmer – Rev Counter for Mopeds

STOP PRESS...

Does alcohol
cause violent behaviour?

Brachion Detector – page 50

Free downloads are available!

On the restyled Elektor Electronics website, individual magazine articles can be downloaded at £1.00 each. Owing to an error in the website construction, £1.00 was also charged for pdf files of PCB layouts not published in the magazine and supplied free of charge till 2 February 2005. This has been reversed with immediate effect, the PCB layout files identified as such in the magazine being available **free of charge** again. We apologise unreservedly for any inconvenience caused.

Paul Snakkers, Publisher

211 Elektors for sale (2)

Further to an item in Mailbox, February 2005, Mr. Mike Pearson kindly advised us that his complete collection of magazines has been sold. We congratulate Mike on the sale and the buyer on obtaining a massive amount of information. We advise readers wishing to clear their stock of old magazines or books to do so using Marketplace on our website. [Editor]

Mobile telephone masts

Dear Editor, here's an alternative thought about the installation of mobile telephone masts near schools. As I understand it, the ERP (effective radiated power) of a handset is directly related to the signal reception quality. For example in a poor signal area the output power of the handset increases and conversely, in a high signal strength area the output power is reduced. This is not just while a call is being made but all of the time that the network is 'handshaking'. This is why the battery runs down quicker when the handset is located in poor reception areas. That being the case and considering that any adverse effects on a Child's health is likely to be caused by the close proximity of the handset to the head, it might be worth considering that a mast close to a school would cause the handsets within that area to be producing less radiation. I wonder if

anyone has undertaken any tests to see what the difference in ERP is under these conditions. Maybe the effects on health would be less if the areas around high numbers of handsets were served with good reception.

Martin Bushnell (UK)

Interesting thoughts Martin, let's hear what other readers have to say about the subject. This discussion may be continued in the Forum on our website.

Copybit Killer

Dear Jan — in an earlier instalment of Mailbox (can't remember which one exactly) I saw a reference to the January 1996 Copybit Inverter project, with an editorial remark saying that that particular design worked best. My Elektor year volume CD-ROMs tell that since 1996, two similar designs were published (in 1997 and 2002), so I wonder what your advice would be today.

Frank Budde (Netherlands)

The design from the June 2002



issue has more potential than the previous ones and is now recommended. You may download the DASP 2002 article from our website.

NiCd battery advantages (2)

Dear Editor — In your February 2005 edition Mr. Mark Vermeulen of the Netherlands has many favourable words about NiCd batteries for cordless tools and RC models. In my country (Denmark) it is very hard to find cordless tools without NiMH batteries, as there is a high penalty tax on NiCd cells. Stock is kept for existing equipment, but the mainstream is NiMH.

That goes for radio controlled airplanes and ship models too, whereas racing car modellers still seem to want the 100+ Amp capability of the NiCd cells for half a minute or so. The conclusion is that the battery manufacturers — under sufficient pressure — are able to make reliable and long-lasting batteries and rather cheap too.

Our discount chain stores sell 18-volt cordless drills for as little as 129 DKK, that is about £ 12 or 17 Euros.

Poul Børge Pedersen (Denmark)

Following this interesting viewpoint from Poul Børge and in the interest of our environment I am again inviting other readers to join the discussion, which may be continued in the Forum on our website.

USB/I²C Interface (2)

Dear Elektor — your 'USB/I²C Interface', EE December 2004, employs the same USB controller (TUSB3410) as the 'MultiProgrammer on USB' from the June 2004 issue. In the first article, the chip is connected to a 3.3-V supply, while a 5-V supply is used in the MultiProgrammer. Considering that an LC ver-

sion of the EEPROM is used, and having seen the maximum permissible values in the TUSB3410 datasheets (max. voltage on I/O pins = Vcc + 0.3 V), I believe your design is flawed. Incidentally, I found that the EEPROM on the MultiProgrammer board is connected to 5 V.

Rainer Bittendorf (Germany)



Correct, but supplying the EEPROM from 5 V is not a problem because the pull-up resistors will limit the current sufficiently. The same method is also seen in Texas Instruments' application notes.

VOX AC-30 query

Dear Jan — I've recently pulled a VOX AC-30 tube amplifier schematic off the internet, and was wondering if someone could answer a question for me. All the AC-30 amplifier schematics that I've come up with have EL84 tubes in the output position, however this particular schematic has EL34's in the output position. Is this a mistake, or has someone taken liberty to upgrade the schematic? Any help would be appreciated, thanks.

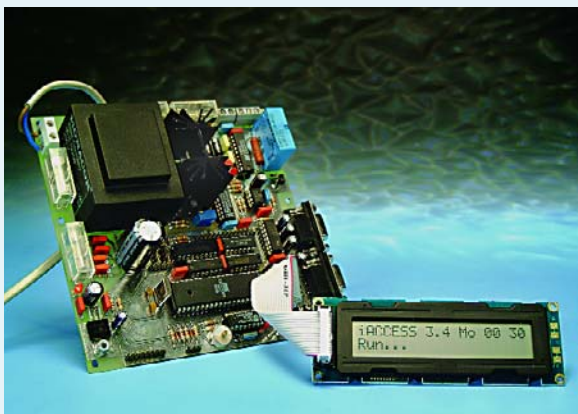
Greg (by email)

The VOX AC-30 amplifier is legendary for its unique sound. It is a collector's item and used by a surprising number of guitar players young and old, from N.E.R.D to Coldplay and New Order. I'm informed there are even DSP-

Is iAccess awake?

Dear Elektor staff — I built the iAccess system (February 2004, Ed.) but I've no idea what should happen when I switch it on? For example, how do I know the microcontroller is actually running?

Dietmar Terhörst
(Project Engineer Electronics, OPEL Germany)



The author/designer Ilse Joostens replies: At power-on or after a reset, the firmware pulls P0.2 (pin 37) on the processor low for about 0.1 second. This should happen even without an LC display module connected. If a buzzer is connected to Bz1 on the printed circuit board, you will hear a short beep since it is connected to P0.2. If you want to measure the level on P0.2 with a voltmeter without a buzzer being present, you need to connect a resistor of about 1 k between the 5-V supply voltage and P0.2 since port 0 of the processor has no internal pull-ups.

If P0.2 does not go low for a short time after reset or power-on, there is likely to be a problem with C3, C4, X1 or IC1 is not, or not properly programmed, with the firmware (IC1 might also be defective).

Be sure to mount crystal X1 about 0.5mm above the printed circuit board to prevent its metal case touching the solder pads on the component side. If necessary, using an oscilloscope you may check the presence of the clock signal on pin 18 (XTAL2) of the controller.

In the past, I noticed that the program Microflash necessary to program the controller with the firmware does not always function properly. Sometimes the controller gets partially programmed or even remains blank without any warning message from Microflash.

If you use Microflash, I would recommend programming the controller again using the ATMELISP software, which is a much better program. You can download it here: <http://www.itronics-design.be/downloads/atmelisp.zip>

Select the right COM port in the 'Parameter' menu and 'DK7JD' for hardware compatibility. More info can be found in the Atmel ISP help file.

Finally, I'd like to mention that if you used a 74LS126 instead of a 74HC126 or a 74HCT126 for IC10, the controller may remain in a reset condition after power up. In that case, you won't be able to program the firmware either. Please check the level at pin 9 (RESET) of the controller, it should be low during normal operation.

based VOX AC-30 simulators but personally prefer the 'real thing'. Anyone able to help Greg?

High-End Preamp

Dear Elektor — I have just started to gather components for the High-End Preamplifier I found in the April 2004 issue. Unfortunately I am unable to find a mail-order supplier for the rectifier diode you state in the parts list. Do you know of a source for the BY500, or failing that, an alternative?

Wim Oldemans
(Netherlands)

With the availability of our new website, this type of question is best asked in the Forum as other readers may have valuable information they would like to share on this topic. Meanwhile, you may use just about any diode rated at 100 V/3 A, for example the ubiquitous 1N5408 which happens to have the same dimensions as the BY500.

information. You can find the article by month/year of publication, title or production number.

Cuk Converter

January 2005, p. 26-32, 030346-1

Customers having received a PCB no. 030346-1 from The PCBShop are advised not to commence construction of the board because it contains several design errors. The faulty board may be identified by the absence of position C35 next to IC3.

Customers affected by this error are kindly requested to send an email stating order and invoice number to sales@elektor-electronics.co.uk. Eurocircuits / The PCBShop currently have the correct PCB design file and will send a free replacement board to customers having reported to the above email address.

MailBox Terms

- Publication of reader's correspondence is at the discretion of the Editor.
- Viewpoints expressed by correspondents are not necessarily those of the Editor or Publisher.
- Correspondence may be translated or edited for length, clarity and style.
- When replying to Mailbox correspondence, please quote Issue number.
- Please send your MailBox correspondence to: editor@elektor-electronics.co.uk or Elektor Electronics, The Editor, P.O. Box 190, Tunbridge Wells TN5 7WY, England.

CORRECTIONS & UPDATES

General

Readers are advised that the quickest way to check for corrections /updates for a project they have built is now by way of our website at

www.elektor-electronics.co.uk.

In the Magazine section of the website, corrections/updates will be included with other article

Mac mini

Unix in a Box

Dr. Thomas Scherer

Apple has managed to slim down the Mac to make a fully-featured computer that fits in an incredibly small box. So what is this 'micro' (in the fullest sense of the word!) computer capable of? What can we use it for? At such an attractive price, this Mac OS X machine could easily win over even the most hardened PC user.

Measuring just 165 mm (w) by 165 mm (d) by 50 mm (h) — about the same size as a pile of five CD cases — and weighing in at just 1.3 kg, the Mac Mini might at first glance be mistaken for a plinth for a vase or perhaps a designer food warmer. In reality it is actually a fully-fledged PC (and not much use as a food warmer: even under full load the Mac mini remains cool). In typical use under partial load it is almost inaudible and in standby mode it is completely silent.

All about it

All the interfaces are brought out to the rear of the enclosure (**Figure 2**), as is, perhaps rather less conveniently, the power switch. The row of slots forms the outlet vent for the fan, which draws in air from under the enclosure at the front, and at the left and right sides. The Mac Mini should therefore not be left on a soft surface, since this can block the air inlets.

Not only is the enclosure astonishingly small, it is also pleasing to look at and beautifully made. There are two main parts: an aluminium frame around 4 mm thick with a white plastic lid, and a base plate which carries all the innards of the machine including the connectors (**Figure 3**).

The two parts are held together by a large number of plastic hooks attached to the base plate,

which lock into recesses in the aluminium frame. Opening the case requires rather more than just a screwdriver, and Apple has certainly not designed the Mac Mini with the case-modding and homebrew fraternities in mind. The machine is clearly designed as a ready-to-go solution with a good out-of-the-box experience as the first priority. For the home user it's ideal for downloading to your iPod or for use as a multimedia workhorse, and it includes everything you need for these applications — and nothing more. Expansion and modifications are intended to be done by the manufacturer; alternatively the machine can be ordered with the desired expansion options, such as extra RAM, a DVD writer, and integrated WLAN and Bluetooth, already installed. *Elektor Electronics* readers need not be put off, however: it's possible to save a few pounds by, for example, expanding the RAM yourself. A spatula with a fine edge will open the case: more detailed instructions and advice on this can be found on the Internet.

What lies within

As you can see from **Figure 3**, Apple is not in the business of selling fresh air. The design of this ultra-compact computer takes advantage of techniques used in the manufacture of notebook computers.



Figure 1. The Mac Mini seen from the front. A typical Apple design: nothing to see except an indicator light, a slot for optical media, and the Apple logo.

The external power supply, which is about a quarter of the size of the Mac mini itself, can deliver up to 85 W, although in normal office-style use measurements show the complete computer only takes less than 20 W. The centrifugal fan also helps make this a very quiet and 'green' computer. The CPU, made by Freescale Semiconductor (formerly Motorola) plays a significant part in this too: under full load, its typical power consumption is only 18 W. The clock rate of 1.25 GHz might seem rather modest in comparison with the clock rates typically found on Intel processors, but it is deceptive: the processor has, for example, an unusually short pipeline, just seven stages long. Like the Pentium M series, this makes it particularly efficient when compared to current Pentium 4 devices with their 31-stage pipeline. The CPU also sports a so-called 'AltiVec' SIMD unit, which gives practically unparalleled performance in multimedia data processing applications. Further speed-ups are achieved using Quartz Extreme, a technology which moves as much as possible of the work involved in producing the graphical user interface into the graphics processor. In terms of raw processing power the system probably ranks at around 2.2 GHz to 2.8 GHz in AMD-style 'P4 equivalent units',

depending on the model chosen. Not the very peak of technology, but then the competition has nothing remotely comparable to offer in as small an enclosure. Also included is iLife, a software package that includes several programs: iPhoto, for enhancing and organising images; iMovieHD, for creating high-definition video clips; iDVD, for creating and burning DVDs; GarageBand, an incredibly powerful tool for creating and experimenting with music; and a free version of iTunes, for downloading and managing music files in MP3 and AAC formats. Also included is AppleWorks, an older program for word processing, spreadsheet calculations and vector drawing.

What else?

It is amazing that Apple can sell the machine at the price they do. Indeed, subtract the price of the operating system (£99) and the iLife multimedia package (£49), both of which are included, and Apple is selling the 1.25 GHz version of the hardware for only £191!

At this price, the keyboard and mouse are not included and must be purchased separately. However, any keyboard and mouse with USB connectors will do; you might prefer the Apple keyboard over one you can buy elsewhere, however, as the Apple layout is slightly different.



Figure 2. The Mac Mini seen from the rear. From left to right: power switch and inlet, network socket, modem socket, DVI-I socket, two USB 2.0 interfaces, 6-pin FireWire connector, 3.5 mm audio output and an oval slot for attaching a Kensington lock.



Figure 3. Inside the Mac Mini. At the top is the ultra-slim optical drive, behind which hides the 2.5" hard disk. Below, to the left, part of the centrifugal black CPU fan can be seen. Above and to the right is the main memory: a standard PC2700 DDR DIMM with 184 pins.

Not only is the machine suitable for home office use, it makes a reliable and virus-free surfing station: to date, viruses for OS X are almost unheard of. It could also form the heart of a quiet and stylish home entertainment system, and already 'media centre' applications are popping up on the Internet.

The machine is also suitable for use as a small all-round Internet server. Thanks to the underlying BSD kernel top-quality implementations of all the necessary programs and protocols are already

in place, including Apache, SMB and so on, allowing its use as a web, FTP, AFP or simple file server. Using 'Apple Remote Desktop' (VNC) the machine can be administered without its mouse or keyboard. When compared with the rather larger mini-ITX systems based around the Via EPIA CPUs, the Mac Mini offers more comfort and power and, moreover, is better value; and that's before you even consider the machine's looks.

(050033-1)

<http://www.apple.com>

Technical data and prices

CPU	MPC7447, alias G4 PowerPC; 512 KB L2 cache; 1.25 GHz or 1.42 GHz
RAM	256 MB PC2700 DDR; expandable to maximum 1 GB
Drives	DVD±R/CD-RW combo or DVD±RW; 2.5" hard disk, 40 GB or 80 GB
Interfaces	FireWire; 2 USB 2.0; modem; audio out; DVI-I (digital and analogue via adaptor)
Software	OS X 10.3.7; MacOS Classic; iLife '05 package

Miscellaneous External power supply; no keyboard or mouse supplied
 £339 for 1.25 GHz/40 GB; £399 for 1.42 GHz/80 GB;
 4x DVD±RW £70 extra; WLAN and Bluetooth £70 extra

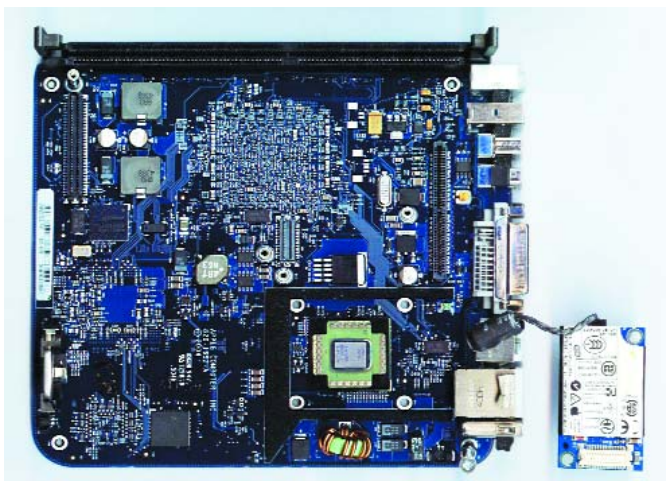


Figure 4. The motherboard seen from above. The RAM slot is along the top edge. Top left is the connector for the optional WLAN and Bluetooth card, while on the right is the IDE interface. In the bottom right corner is the CPU with the heatsink removed. (photo courtesy macnews.de)

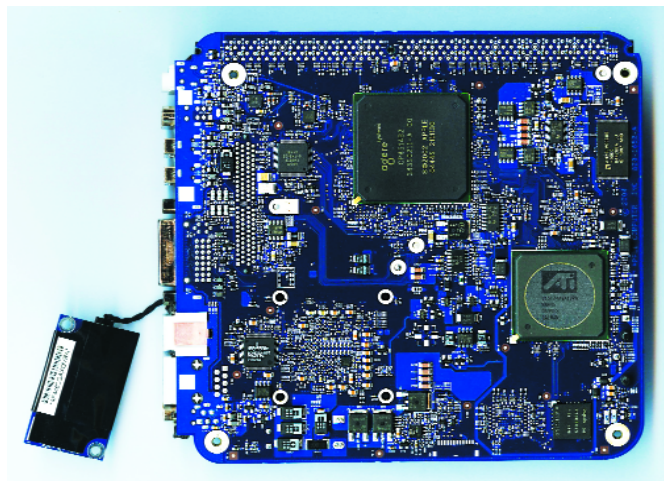


Figure 5. The motherboard seen from below. The large chip at the top is Apple-specific and does the same job as the north and south bridges in a PC. On the right is the ATI 9200 GPU, which has its own 32 MB of dedicated graphics memory. (photo courtesy macnews.de)

SAVING POWER *selecting the right microcontroller*

When the author of this article started studying microcontrollers around 25 years ago, the answer to this question was rather simple. There were only two or three manufacturers active in the field, and none of them offered more than two or three different types. Comparing the various types and determining which one best met your requirements was thus easy (and considering the state of affairs at that time, it was actually more a question of which one had the fewest shortcomings).

Nowadays we are overwhelmed by a flood of different types from dozens of manufacturers, all descended from various families that occasionally compete with their own stablemates. It's a hopeless task to try to directly compare the features of the various types of microcontrollers, since that would require poring through thousands of data sheets. You thus have to take a different approach, and the purpose of this article is to assist you in this by suggesting a few questions you should ask yourself. We also look at the answers you might expect to receive to these questions and now to decide which microcontroller families have the best features for your specific application. However, we don't want to mislead you: we can't give you any magic formula for selecting the most suitable microcontroller, for the simple reason that it's not possible. In fact, all types of microcontrollers can handle just about every sort of task, which is what makes them so powerful. It also means that what makes or breaks an application is not the pure performance figures of the microcontroller, but instead how you use it.

Survey

According to a recent American user survey conducted among major developers of microcontroller-based applications, the relative importance of the selection criteria can be rated as follows:

- availability of development software: 67%
- price: 51%
- number of I/O pins and available internal peripherals: 40%
- availability of developer kits: 35%
- compatibility of the generated code: 27%

You have probably noticed that this adds up to a good deal more than 100%, but the survey allowed multiple responses. The responses also give a good picture of the problems posed by such a question and how difficult it is to identify a predominant criterion.

Are you experienced?

The first question you must ask yourself is whether you already have any experience with a particular microcontroller family. If you do, you should concentrate your further interest on microcontrollers from that family unless you are not particularly satisfied with it. The advantage

of this approach is that you are already aware of the manufacturer's 'peculiarities', so you don't have to again learn all the details of an instruction set, addressing modes, register names, how to handle inputs and outputs, and so on.

If you have also invested in various development tools, such as a development environment, a programmer and perhaps even an emulator, you will probably be able to continue using them (although an upgrade may be necessary).

On the other hand, if you do not already have any experience with microcontrollers, your choice is completely open and you can use the selection criteria described below.

Assembly language or high-level language?

Although this question only involves the software, the answer is closely related to the selected hardware. Looking at it from the other direction, choosing a particular type of software can also result in selecting hardware that can work together with the software.

This question was irrelevant a few years ago, since at that time the amount of program memory in microcontrollers was so small that the only option was programming in assembly language, which is commonly but improperly called 'assembler'. If you program in a high-level language, you have to allow for the fact that your code will be translated into assembly language by a compiler and that each line of code in a high-level language usually generates many lines of code in assembly language. This means that if you program in a high-level language, you need a microcontroller with a relatively large memory capacity.

Although modern compilers have made enormous progress in optimising the generated code, they're still no match for 'hand-coded' assembly language, at least when the main activity of the code is manipulating bits in registers, as is typically the case for microcontroller applications. Compilers for high-level languages have problems dealing with such operations. For example, consider the instruction

```
output_D(i);
```

The purpose of this instruction is place the value of the 8-bit integer *i* on output port D of a microcontroller. On the better C compilers for current PICs translates this as follows:

```
MOVLW 00
BSF 03.5
MOVWF 08
BCF 03.5
MOVF 20,W
MOVWF 08
```


Figure 1. Although MPLAB is free, it is a full-fledged development environment with an editor and assembler, and it even includes a simulator. Everything works together perfectly.

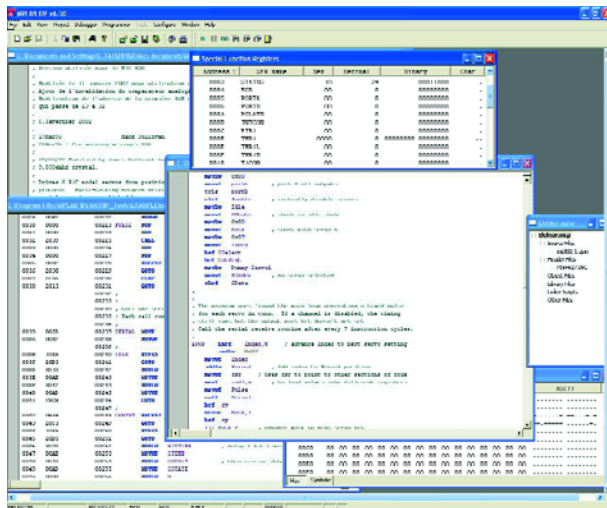


Figure 2. The PCWH development environment for the CCS C compiler for PIC microcontrollers is also a good example of successful integration.

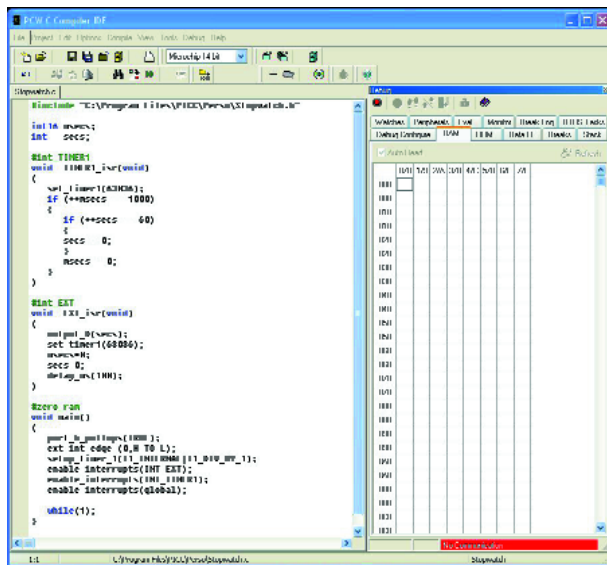
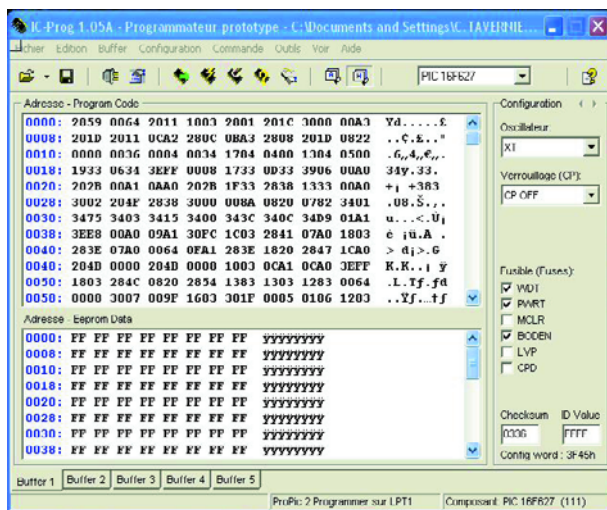


Figure 3. The well-known freeware IcProg (www.ic-prog.com) can be used to program various types of microcontrollers using very simple programmers.



A programmer coding in assembly language can do the same thing using only two instructions. In this particular (but realistic) example, using a compiler increases the memory requirement by a factor of three compared with using assembly language.

If your application requires making calculations using digits after the decimal point or in scientific notation, or if it employs complex mathematical functions such as trigonometric formulas, using a high-level language is practically unavoidable. Anyone who has ever computed a cosine in assembly language will doubtless agree with this!

The choice of programming language thus depends primarily on the sort of calculations that are made in your application. If all the program has to do is manipulate inputs and outputs and make simple calculations, assembly language is the right choice. On the other hand, if the program has to perform numerous calculations or use complex mathematical functions, you should choose a high-level language.

There is one exception to this rule: if you are experienced in using a high-level language, even though it may not be the most suitable language for microcontroller applications, you will probably want to use it in order to avoid having to learn how to program in assembly language. However, you should bear in mind that this will unavoidably result in reduced processing power for your application.

Before leaving the subject of software, we should note that with a few exceptions, the only high-level languages available for microcontrollers are Basic and C.

Even after you've chosen a programming language, you still don't know which microcontroller family you should select. However, one thing is already certain: if you want to use a high-level language, you will need a relatively large amount of program memory.

Development environment

The development environment includes several programs and interfaces, including:

- a text editor for writing programs;
- an assembler if you use assembly language, or a compiler if you use a high-level language, although a compiler is always accompanied by a matching assembler;
- a simulator, which is a program that can be used to simulate the program for testing;
- an emulator, which is a mix of hardware and software that can be used to test the program before it is programmed into the actual application environment;
- a programmer, for programming the microcontroller with the program you have developed.

Aside from the simulator and the emulator, which are optional, these elements are indispensable. Nowadays they are often packaged into what is called an 'integrated development environment' (IDE). These programs, most of which run under Windows, make a polished impression with their many windows, and they allow you to quickly jump from the editor to the assembler or compiler, or even to the simulator or emulator if they are integrated into the package. The screen shots made for this article show some examples of the Microchip MPLAB environment and the PCWH development environment for the C compiler from CCS.

With the exception of the emulator and the programmer, which partially consist of hardware that you have to pay for, most of these tools consist entirely of software. Some microcontroller manufacturers use a handy commercial trick here: they make the programs available free of

Don't program that μ C, bootload it!

Now that microcontrollers with a respectable amount of program memory are commercially available, a new concept has become popular: the boot loader.

A boot loader is a very small program that is programmed into the low or high memory of the microcontroller fitted in the circuit to be developed. This program can communicate with the development tools (used to write the program code) via a serial port or some other type of serial link, such as USB, I²C, or a CAN bus.

The boot loader program can interpret a certain number of commands related to reading, programming and erasing the program memory of the associated microcontroller. The working principle is basically fairly simple.

When the program being developed is ready for testing, communication with the boot loader is initiated. The boot loader then loads the program code into the program memory of the microcontroller, thus avoiding the need to use an external programmer.

After this operation is completed, the boot loader transfers control to the loaded program, and the user can then execute and test the loaded program. A new program can be loaded

using the boot loader as often as necessary (after the previous program has been erased).

The speed gain obtained by using this method is a factor of ten relative to the traditional method of programming, which naturally has a positive effect on the entire development process.

You have probably already realised that the microcontroller must meet a certain number of requirements to allow a boot loader to be used. In particular, it must have:

- sufficient program memory to hold the boot loader in addition to the program being developed,
- internal support for erasing and programming the program memory, and
- an RS232 port or other type of serial link, such as USB or CAN.

If these requirements are met, it is fairly easy to use a boot loader. There are presently numerous freeware programs available via the Internet for microcontroller families that support boot loading.

charge. That may seem strange, but the logical consequence of this way of doing business is that developers use these programs and are thus inclined to also use the microcontrollers from the same manufacturers.

Although these programs are free, they are certainly not second-rate software. In fact, MPLAB is one of the best programs the author is aware of. Beside the fact that this environment allows you to program in assembly language (it comes complete with an editor, assembler and simulator), it cooperates almost automatically with compilers and emulators from other manufacturers. They are so well integrated into the development environment that they seem to actually belong to it.

This 'generosity' with regard development tools is limited to programming in assembly language. Practically all current makers of compilers for high-level languages charge money for their products. Fortunately, many compiler suppliers (for Basic and C) make available demo versions that are adequate for hobbyists and allow professionals to quickly test their products.

The table accompanying this article lists all of the development environments available from microcontroller manufacturers. You can download them and try them out without having to use an actual microcontroller. That gives you an opportunity to find out which of them are the most user-friendly. If you wish to (or must) use a high-level programming language, it's a good idea to check whether there is an integrated compiler for that language, which further underscores the importance of demo versions.

Finally, there's the programmer, since sooner or later you will naturally want to program your microcontroller. The current trend in this area is toward in-system programming (ISP) or in-circuit serial programming (ICSP). These techniques allow the microcontroller to be programmed after it has been fitted to the circuit board, using a connection with only three or four leads. The required programmer is very simple, and suitable designs for various microcontroller families have already been published in *Elektronik*. The availability, price and/or difficulty of building the programmer are thus factors you should take

into account when selecting a microcontroller family.

Another recent development is the 'boot loader', which is certainly a handy tool in the development stage of a program. We have dedicated a separate inset to this subject. However, boot loaders can only be used with microcontrollers that can program their own program memory. Manufacturers are slowly but surely employing this technique, but many of them have not yet dared to take this hurdle. In conclusion to this discussion of development tools, we can say that for a hobbyist or small company that cannot afford to make a large investment, the availability of a simple programmer or one that can be built at a low cost, together with free development tools, will be the decisive factor in selecting the microcontroller family. Once again, we would like to point out that the accompanying table can help you considerably in making your selection.

8, 16 or 32 bits?

Microcontrollers just keep on growing with regard to processing power. The 4-bit types have just about disappeared from the market, and current microcontrollers can process 8-bit, 16-bit, or even 32-bit data. However, you shouldn't draw the conclusion that 32-bit types are necessarily the most powerful and you're archaic if you use an 8-bit microcontroller.

In fact, if your program only controls a few inputs and outputs and doesn't use complex computations or large databases, an 8-bit type is a better choice than a 16-bit or 32-bit model. That may sound illogical, but the following example can quickly show that it isn't. If your application uses a serial RS232 link, the data are treated as 8-bit words. That's because ASCII characters are transmitted via this link, and as you know, ASCII characters have a size of seven or eight bits (standard or extended set). A 32-bit microcontroller is at a disadvantage in such a situation, since it has to process the 8-bit data as 32-bit words, which means 24 bits have to be masked out. If you have to make mathematical or scientific calculations in your application, a 16-bit or 32-bit microcon-

Selection table. This table lists the various families from the major manufacturers. The 'Core' column indicates the type of architecture or processing unit that forms the basis of the microcontroller. The 'Free IDE' column lists the names of the development environments that can be downloaded from the related manufacturer sites.

Manufacturer	Internet	Data bus	Family	Architecture	Core	Free IDE
Analog Device	www.analog.com	8 bits	ADUC8xx	CISC	8051	–
		32 bits	ADUC7xx	RISC	ARM7	–
Atmel	www.atmel.com	8 bits	AT89xxx	CISC	8051	independent programs
		8 bits	TS87xxx	CISC	–	independent programs
		8 bits	AVR	RISC	–	AVR studio
		16 bits	AT91xxx	RISC	ARM7/9	–
Cirrus Logic	www.cirrus.com	32 bits	EP73xxx	RISC	ARM7	–
		32 bits	EP93xxx	RISC	ARM9	–
Cygnal	www.silabs.com	8 bits	C8051F	CISC	8051	–
Freescale (ex. Motorola)	www.freescale.com	8 bits	HC05	CISC	6800	–
		8 bits	HC08	CISC	6809	Code Warrior
		8 bits	HC11	CISC	6809	–
		16 bits	HC12	CISC	–	–
		16 bits	HCS12	CISC	–	Code Warrior
		16 bits	HC16	CISC	–	–
		16 bits	56800	CISC	–	–
		32 bits	68K	CISC	68000	–
		32 bits	ColdFire	CISC	–	–
		32 bits	MAC7100	RISC	ARM7	–
Fujitsu	www.fujitsu.com	8 bits	F2MC–8	CISC	–	–
		16 bits	F2MC–16	CISC	–	–
		32 bits	FR	RISC	–	–
Infineon	www.infineon.com	8 bits	C5xxx, C8xxx	CISC	8051	–
		16 bits	C16xxx	CISC	–	–
		16 bits	XC16xxx	CISC	–	–
		32 bits	TCxxx	CISC	–	–
Intel	www.intel.com	8 bits	MCS251	CISC	8051	–
		16 bits	MCS96/296	CISC	–	–
Maxim (Dallas)	www.maxim-ic.com	8 bits	DS80Cxxx	CISC	8051	–
		8 bits	DS83Cxxx	CISC	8051	–
		8 bits	DS89Cxxx	CISC	8051	–
		16 bits	MAXQ	RISC	–	–
Microchip	www.microchip.com	8 bits	PIC 10, 12, 16, 18	RISC	–	MPLAB
		16 bits	dsPIC	RISC	–	MPLAB

troller is naturally the best choice, since it can process floating-point calculations much faster than an 8-bit counterpart.

You might be inclined to choose a compromise solution by selecting a 16-bit microcontroller. However, in light of present developments we don't recommend this. This is partly because a more and more 32-bit microcontrollers are becoming available at reasonable prices, and partly because 8-bit microcontrollers are constantly becoming more powerful and less expensive. This makes 16-bit types something of an 'odd man out'. If you want to say farewell to your 8-bit microcontroller because it isn't powerful enough for a particular application, you're better off to move directly to a 32-bit microcontroller, since nowadays the price difference relative to 16-bit types is almost negligible.

ROM, OTPROM, EEPROM and Flash

All of these terms stand for different types of memory that sometimes form a specific data component of the microcontroller. It goes without saying that microcontrollers with EEPROM or Flash memory are preferable for one-off use, because they are the only types that can be programmed, erased and reprogrammed electronically, up to several thousand times if necessary.

After your prototype has been developed using a microcontroller with EEPROM or Flash memory, it is financially

attractive to use microcontrollers with one-time-programmable memory (OTPROM) for small-scale mass production. These devices can be programmed using the same tools as types with erasable memory, but they cannot be erased afterwards. They are usually cheaper than their namesakes with EEPROM or Flash memory, which are intended to be used in the development stage.

Note: versions with different types of memory are not always available for all types of microcontrollers from a particular manufacturer. You should thus always consult the manufacturer's website to check the availability of various types of memory for specific microcontroller types. You should also be aware that there are microcontrollers designed for use in large-scale mass production, which have ROM that can only be programmed using a mask. However, you can forget about using such types unless you need several thousand identical devices.

Inputs, outputs and internal peripherals

One of the attractive aspects of microcontrollers is that they allow the external circuitry to be kept to a minimum or even eliminated entirely. If you know what your application needs, then all you have to do is make a list of the following items:

Manufacturer	Internet	Data bus	Family	Architecture	Core	FreeIDE
NS	www.national.com	8 bits	COP8xxx	CISC	–	Webench
		16 bits	CR16Cxxx	CISC	–	–
		16 bits	CP3000	RISC	–	–
Philips	www.semiconductors.philips.com	8 bits	P8xxx	CISC	8051	–
		16 bits	Xaxxx	CISC	–	–
		32 bits	LPC2000	RISC	ARM7	–
Rabbit Semiconductor	www.rabbitsemiconductor.com	8 bits	Rabbit2000	CISC	–	–
		8 bits	Rabbit3000	CISC	–	–
Renesas	www.renesas.com	8 bits	740	CISC	–	–
		16 bits	H8	CISC	–	HEW
		16 bits	H8S	CISC	–	HEW
		16 bits	M16C	CISC	–	–
		16 bits	7700	CISC	–	–
		32 bits	H8SX	CISC	–	–
		32 bits	Super H	CISC	–	HEW
ST	www.stm.com	8 bits	ST5	CISC	–	Visual FIVE
		8 bits	ST6	CISC	–	–
		8 bits	ST7	CISC	–	STVD 7
		8 bits	ST9	CISC	–	STVD 9
		16 bits	ST9	CISC	–	STVD 9
		16 bits	ST10	CISC	–	–
		32 bits	ARM7	RISC	ARM7	–
Texas Instruments	www.ti.com	8 bits	MSC12xxx	CISC	8051	–
		16 bits	MSP430	CISC	–	Eclipse
		32 bits	TMS470	RISC	ARM7	–
Toshiba	chips.toshiba.com	8 bits	870	CISC	–	–
		16 bits	900/900H	CISC	–	–
		32 bits	900/900H	CISC	–	–
Uvicom (ex. Scenix)	www.ubicom.com	8 bits	SXxx	RISC	–	–
Zilog	www.zilog.com	8 bits	Z8xxx	CISC	Z80	–
		8 bits	Z8Encore !	CISC	Z80	–
		8 bits	eZ80Aclaim	CISC	Z80	–

- number of parallel inputs and outputs
- number and size (8, 16, or 32 bits) of necessary timers
- availability of one or more PWM ports
- availability of an analogue/digital converter, including resolution and number of inputs
- availability of a synchronous serial port or SPI
- availability of interfaces for specific busses, such as CAN, I²C, USB, etc.
- availability of specific interfaces, such as for driving a dumb LCD

After you have put together this list, you can simply compare it with the data for the microcontroller families that remain after selection using your previous criteria. In light of the large number of microcontrollers on the market, you're bound to find a match in one of the many available families.

If your application makes high demands on the microcontroller, your choice will almost automatically fall on a 'heavy-duty' microcontroller, which may well be too expensive for your application. Naturally, you can also look for a compromise solution. For instance, if you need an A/D converter and it isn't available in the desired family, it may be possible to use an external converter with an I²C or SPI interface. If you need a USB port, you'll quickly notice that your choices are limited to the latest types, which also have all sorts of bells and whistles that you probably don't need. Fortunately, excellent

external USB controllers are also available, and they can be easily interfaced to the microcontroller you have selected.

Speed and current consumption

Microcontroller clock rates just keep on rising. The first types worked at a sedate 1 MHz, but nowadays you can find types with clock rates of 100 MHz. In our eternal chase for better performance, we often tend to automatically select the microcontroller with the highest clock speed. Although a faster clock in principle yields a faster microcontroller, it's a good idea to take a broader perspective here.

For a variety of reasons, you can't blindly take the clock rate as an indication of the effective speed of a microcontroller. For one thing, the clock rate is often divided internally, so the actual working frequency is almost never the same as what the marking on the IC might suggest. Depending on the architecture of the microcontroller (RISC or CISC), instructions are executed in a single clock cycle (RISC) or in several clock cycles (CISC), where 'several' can amount to ten or more. As a concrete example, an 8051 operating at a clock speed of 30 MHz has lower performance than an Atmel AVR operating at a clock speed of only 10 MHz (both running the same program, of course). The 8051 has a CISC architecture and needs 12 clock cycles to execute some of its instructions,

About the author

Christian Tavernier is a consulting electronic engineer associated with a university and a legal expert in the fields of informatics and electronics. He has written numerous books and articles and has specialised in the area of microcontrollers for 25 years already, so we are pleased to be able to draw on his experience for this article.

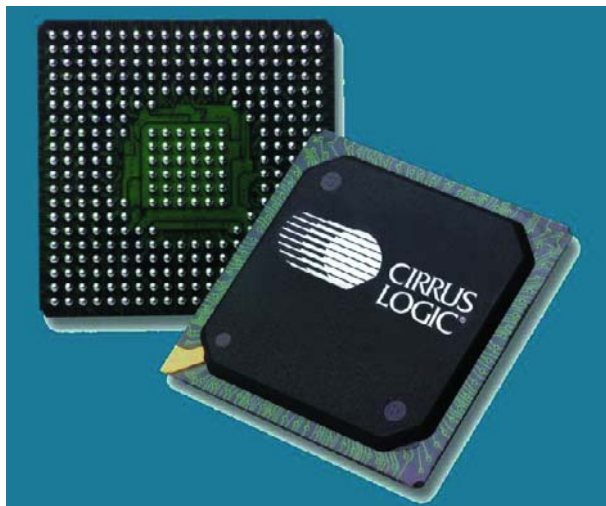


www.tavernier-c.com

contact@tavernier-c.com

while the AVR has a RISC architecture and executes practically all instructions in a single clock cycle. Increasing the clock speed also increases the amount of electromagnetic interference that is radiated, and this interference is also more difficult to control at high frequencies. All electronic products are now required to comply with the European EMC directives, so this aspect cannot simply be ignored.

Finally, you shouldn't overlook the issue of power consumption. The faster the microcontroller operates, the more current it consumes. The 'speed demons' should thus be reserved for applications where the power source is not a problem. Of course, all modern ICs are implemented in MOS technology, but that is not a guarantee for low current consumption. You should bear in mind that CMOS devices only consume current during switching – and the faster the clock rate, the faster they switch. If current consumption is an important consideration in your application, you should pay attention to the clock rate, and you should also select a microcontroller with a standby or sleep mode. In these modes the microcontroller is idle, which means the current consumption is as much as 1000 times less than in the normal operating mode. You may be wondering what purpose it serves to have a microcontroller that isn't doing anything. The answer is that it simply waits until it is 'woken up', so it can do something. This situation often occurs in many



applications involving communication with a human being. For instance, the only thing a programmable thermostat does is to measure the temperature every few minutes and wait to see whether the user does anything with the keypad. It can thus operate in the sleep mode for the vast majority of the time, which in fact is what most such devices actually do. That's why they can operate for more than a year on a single penlight battery.

Lest you forget: the package

Finally, you've found the microcontroller of your dreams, and now you're ready to start the circuit board layout. You feverishly start searching through the data sheet for the IC footprint, which you totally ignored up to now... and that's when you hit the snag. Particularly if you have selected a recent microcontroller or one with an abundance of internal peripherals, you can usually forget about the old, familiar DIL socket, since the number of leads for certain types of microcontrollers can amount to more than 100. There's thus a very good chance that your microcontroller is only available in a PLCC or PGA version. That's not such a disaster, since sockets are available for these packages. But if the package is difficult to fit using manual soldering, you have a problem. The first-generation SMDs, with a lead pitch of 1.27 mm, can be soldered with a bit of patience and a fine-tip soldering iron, but the latest SMDs with a lead pitch of 0.64 mm are practically impossible to solder without using a soldering machine.

And even if you're still willing to tackle the challenge, manual soldering is completely impossible with the increasingly popular BGA packages. If this term doesn't mean anything to you, just imagine a small, square ceramic plate covered with a large number of hemispherical bumps, and you'll understand the problem. Before making a final choice, it's thus a good idea to have a good look through the manufacturer's data sheets to see which packages are available for the selected types. At present (early 2005), this problem is limited to the most recent types of microcontrollers.

Ghostware microcontrollers

This doesn't mean microcontrollers born in ancient Scottish castles, but simply microcontrollers that only exist on paper. Motorola (which is now called Freescale) once went totally overboard in this regard with the 68HC05 family, which promised quite a lot on paper but actually consisted of only one or two types.

This is a secondary issue for a large company, since if you want to order several thousand microcontrollers from the catalogue, the manufacturer will actually supply them. But if you only want a few pieces, or maybe even just one, it's a good idea check whether there is a distributor who actually has them in stock (and in the right package) before starting your design.

Conclusion

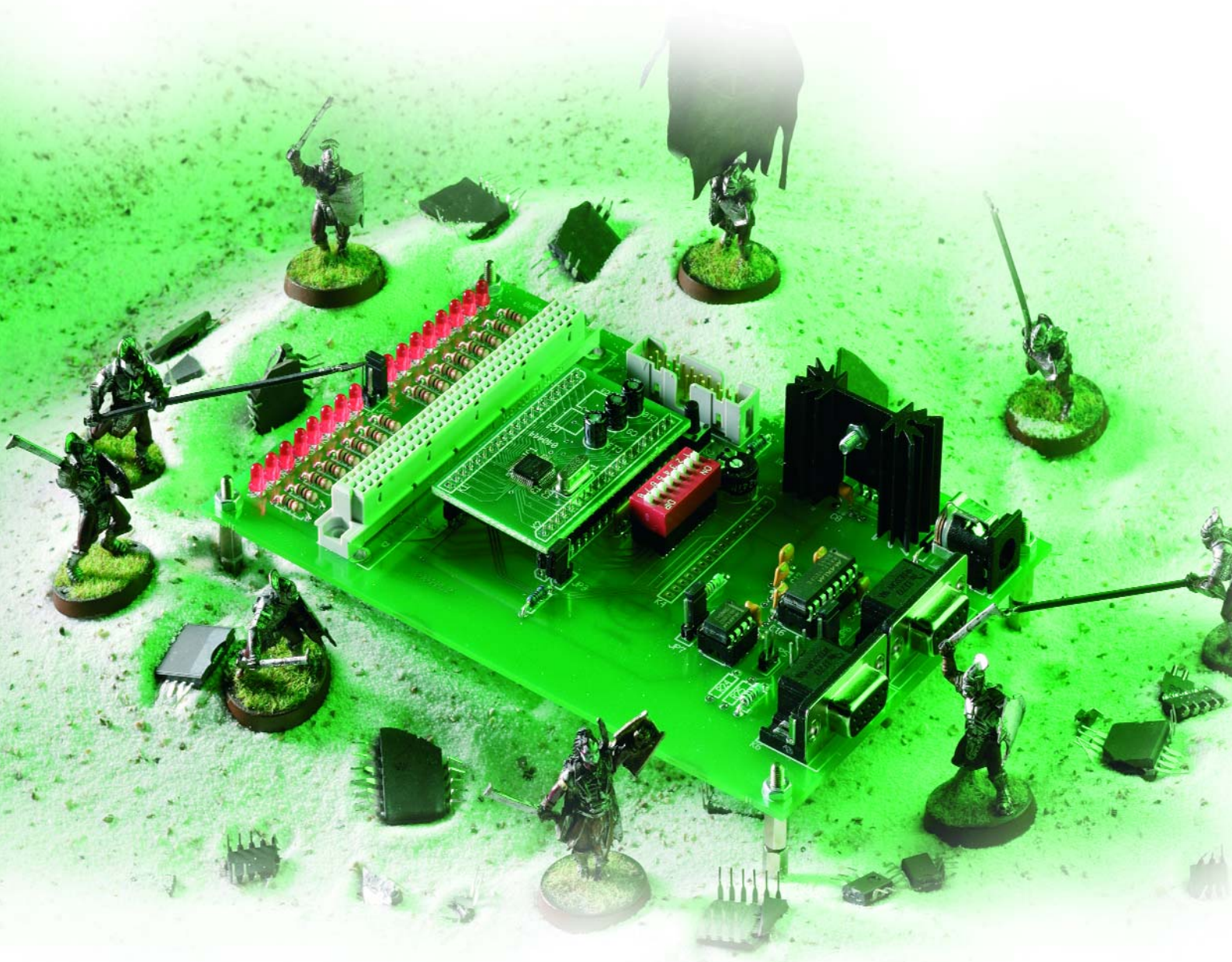
At the beginning of this article, we warned you that it wouldn't conclude with the sentence, 'And now here's your microcontroller...'. Still, we hope that after this discussion you will be able to ask and answer the right questions on your own, after which you can make an initial selection using the accompanying table.

(050003-1)

LPC210x 'ARMee'

Part 2: build and program the ARMee board

Tony Dixon



Development System

Bored with PICs, AVR's and 8051's everyone else is doing? Last month we covered 32-bit ARM microcontrollers and the undeniable star from all the devices discussed in that article was the Philips LPC210x. This month the real thing is upon us: enter **ARMee**, an incredibly powerful ARM development board you can build and program yourself.

Over the years *Elektor Electronics* have published a number of microcontroller development boards and this article is the latest in a long line of such articles. What's different about this article and this design is that instead of being based on an 8-bit device such as a PIC [1], AVR [2] or 8051 [3] controller, this design will use a **32-bit ARM microcontroller**: the LPC210x from Philips Semiconductors. This device has just about everything you could want from a microcontroller, see the '**ARMee & LPC210x Main Features**' inset. However, we should also mention that it lacks any ADC interface and an external bus interface. But then the LPC210x is a true microcontroller and has no external bus interface to extend the memory capabilities using external Flash or SRAM chips, or add new peripherals.

Hardware

The circuit diagram of the development system is shown in **Figure 1**. The motherboard may be powered from a standard DC mains adapter capable of supplying between 9 V and 15 V DC. An adapter capable of supplying around 500 mA will be more than sufficient for powering the development system board and any other reasonable amount of hardware connected to the development board. On the processor daughterboard, voltage regulators provide the various volt-

ages required by the LPC210x: 1.8 V for its CPU core and 3.3 V for its peripherals and I/O. The LPC210x micro has 5-V tolerant I/O and is capable of directly driving 5-V TTL logic.

Like many modern processors the LPC210x provides a reset circuit internally, so an external pull-up resistor is all that is required to finish the design. Pushbutton S2 is available to allow for manual resets to occur.

RS232 serial communication is provided at connector K5 and the interface is built around IC1, a MAX3232 RS232 Transceiver chip providing two driver and two receiver circuits. The MAX3232 is powered from the 3.3-V supply and an on-board charge pump provides the +12 V and -12 V needed by the RS232 interface. The board has the option of a second RS232 interface at connector K6 provided by the second RS232 driver and receiver circuits of IC1. Jumpers JP6 and JP7 should be fitted to the correct position to enable the second RS232 port.

If two RS232 interfaces are not required then connector K6 can provide a Modbus pin-compatible RS485 interface. The RS485 interface is driven by IC2, a MAX3082 RS485 Transceiver chip from Maxim, which is also powered from the 3.3-V supply. Jumpers JP6 and JP7 should be fitted to the correct position to enable the RS485 port

instead of the second RS232 port.

As already stated, the RS485 interface is Modbus pin-compatible. However, Modbus uses pin 5 of the 9-way sub-D connector as a signal line while for RS232 pin 5 is connected to signal ground. To overcome this problem, jumper JP4 also needs to be fitted correctly to select either pin 5 as an RS485 signal or as RS232 signal ground.

Connector K4 provides the option of linking a standard alphanumeric LCD to the system board. The LCD interface shares some of the same LPC210x I/O pins with the 8-way DIL switch. To use the LCD, every switch position of the DIL switch must be set to Off. A preset P1 is used to adjust the LCD contrast. The LCD R/W signal pin 5 of K4 is tied to 0 V thus preventing a program to read back any data from the LCD. Therefore a program should use a simple delay to allow the LCD sufficient time to process its last command. Digital I/O lines P0.22 and P0.23 control the LCD RS and E signals respectively. If the LCD module has an integrated backlight it may be powered through current limiting resistor R22.

An array of 16 LEDs (D1-D16) is provided on the board. The LEDs are connected to the first 16 digital input / output (I/O) lines of the LPC210x and can be used to indicate status or monitor peripheral activity such as activity on the UARTs. LED D1 is connected to

ARMee & LPC210x Main Features

LPC210x (2104/2105/2106):

- 16/32-bit ARM7TDMI-S processor core
- 128 kB program Flash memory
- 16/32/64 kB SRAM data memory
- ISP and IAP
- digital I/O; 2 UARTs
- SPI and I²C interfaces
- timer capture and PWM outputs
- JTAG debugging interface

ARMee Development System:

- direct lineage with earlier *Elektor Electronics* development systems

- detachable processor module available ready-stuffed with LPC2106
- LEDs for viewing port status
- 8-way DIL switch for switch inputs
- connector for linking to a standard alphanumeric LCD module
- real-world connectivity via two 9-way sub-D connectors: 2 x RS232 or 1 x RS232 and 1 x Modbus pin-compatible RS485
- DIN41612 I/O expansion connector on motherboard
- 160x100 mm Eurocard
- Free, multi-platform GNU GCC 'C' compilers
- Wide selection of free and commercial ARM development software

P0.00, LED D2 to P01 and so on. It should be noted that the LEDs also share the LPC210x I/O pins with other functions such as the UART0 and UART1.

The LEDs are divided into two groups of 8. The first group, D1-D8 are enabled by jumper JP1 connecting them to 0 V and the second group D8-D16 are enabled by jumper JP2 again connecting them to 0 V.

An 8-way DIL switch, S1, shares the same LPC210x I/O pins as the LCD interface connector and should only be used if no LCD module is fitted to the system. You may use this switch for debugging purposes or use it as a configuration switch.

On the motherboard a DIN41612 A+C connector, K2, is available for the connection of external circuitry to the development system. This connector carries the voltage supply rails +5 V and +3.3 V, clock and reset signals and all 32 I/O signals from the LPC210x. The connector also carries signals for separate I²C and SPI interfaces. It should be noted that on the LPC210x an I/O pin may provide more than one function, in this case the I/O pin may be shared with the UART, I²C, SPI or other signals. It should also be noted that the on board LEDs, 8-way DIL switch and LCD connection also share I/O signals on the DIN41612 connector.

Printed circuit boards

The PCB artwork for the motherboard and detachable processor board is shown in **Figure 2**. The processor daughterboard is avail-

able ready-built as item **040444-91** (with LPC2106 fitted), or bare as item **040444-1**; see Readers Services pages or go to our Online Shop.

The LPC210x is a 48-pin surface mount device (SMD) and will need to be soldered directly to the PCB. The bare

PCB **040444-1** comes pre-tinned. Position IC1 carefully onto the SMD pads, then using a fine tipped soldering iron, solder each corner of the IC. If you are happy with the IC's position then continue and carefully solder the rest of the IC pins. Solder carefully, but don't worry if excess solder connects the

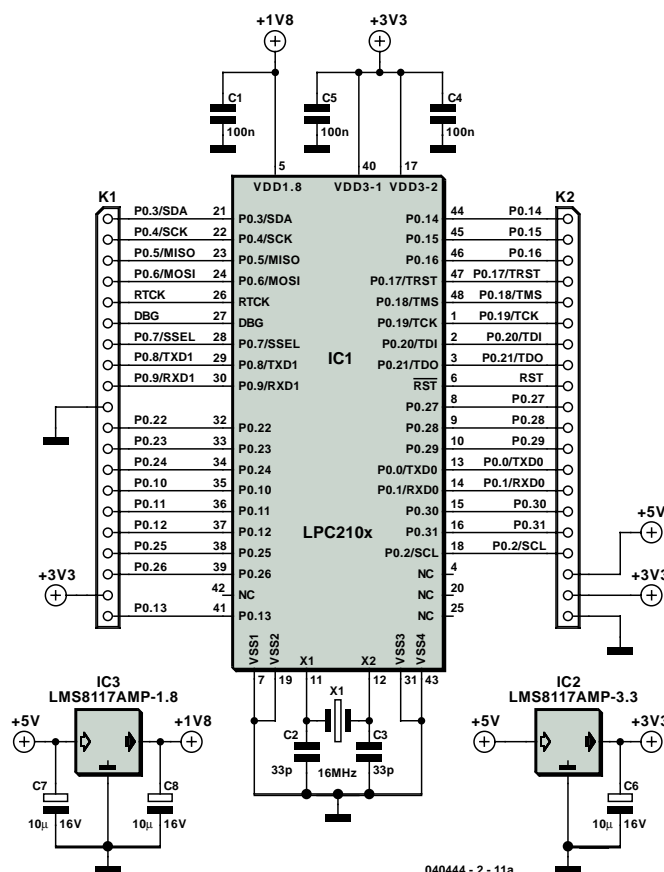


Figure 1. Circuit diagram of the ARMee Development System. Note that the processor module is detachable!

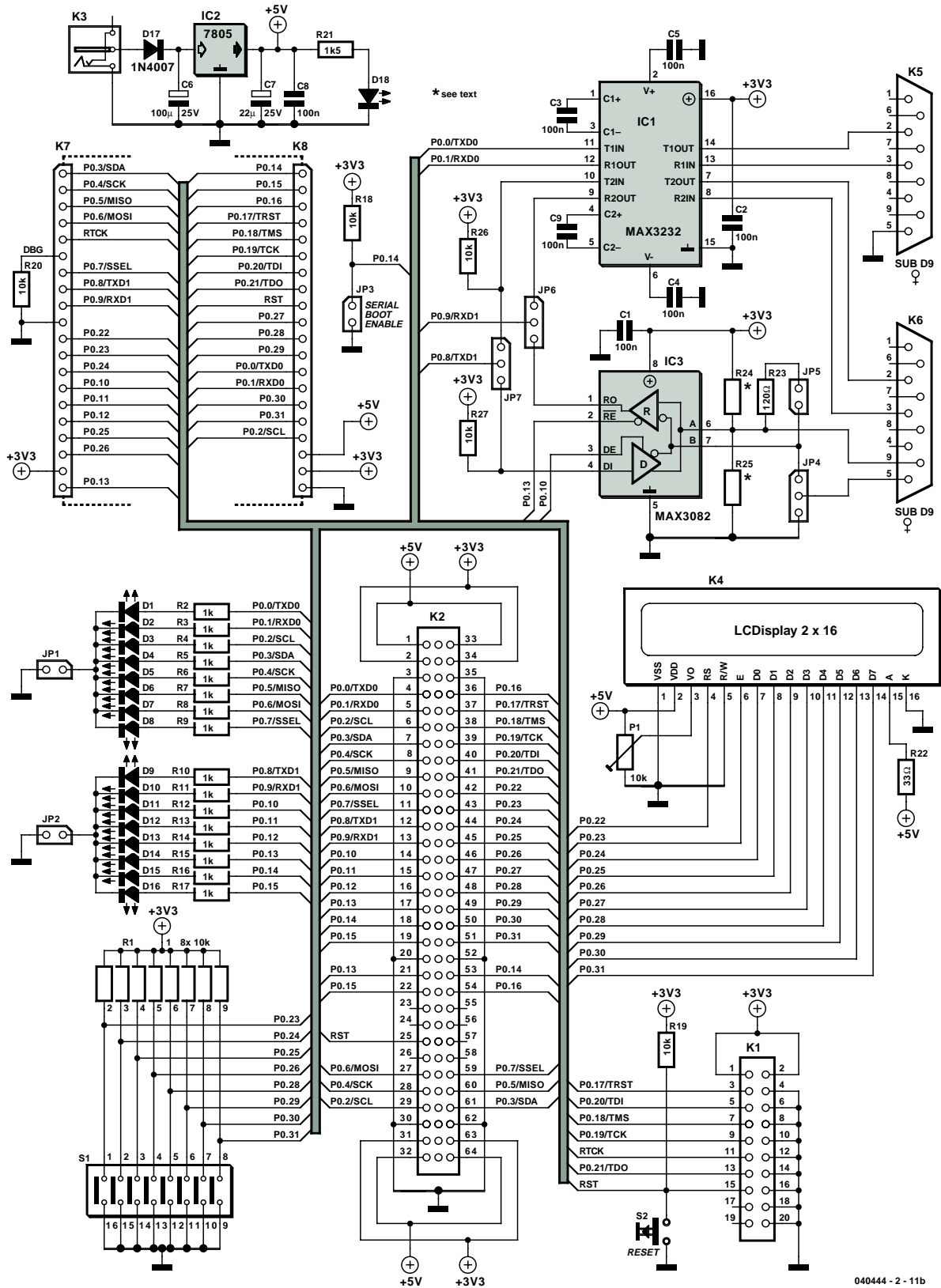
pins. Using some fresh desolder braid apply to each solder bridge and heat with a soldering iron for a moment, allowing the braid to absorb the solder from the bridge. Once the bridge is removed carefully resolder with the fine tipped soldering iron. The bare motherboard PCB is available ready-made from our Readers Services

and online Shop as item no. **040444-2**. When fitting components to the PCB, do observe the polarities of the diodes, the electrolytic capacitors and the LEDs. IC1 and IC3 on the motherboard can be fitted into sockets. With all components are fitted do one last visual inspection and compare the fruit of your efforts with our working proto-

type shown in **Figure 3** before applying power to the board.

Programming Tools

The LPC210x is easily programmed using the C programming language. A number of commercial C-Compilers are available for the ARM architecture (see



web links). Those users who do not want to purchase a commercial C-compiler can use the GNU C-Compiler. This is an open-source C-compiler that is considered by many as good as most of the commercial offerings. The downside of using the GNU compiler is that you don't have the polished installation process of a commercial compiler. The LPC210x IC itself can be pro-

grammed using the on-board JTAG or the internal Serial Boot Loader.

Go JTAG!

The JTAG interface can be used for more than just programming the chip. It can be used to debug a program during execution on LPC210x. To use the JTAG interface, a JTAG debugging interface

module is required to connect between the LPC210x development system and a host PC. There are several low cost commercial JTAG interface modules available, as well as several non-commercial interpretations. These can be found though a simple search on the web, or by checking out the web links provided at the end of this article. If JTAG debugging is not required then

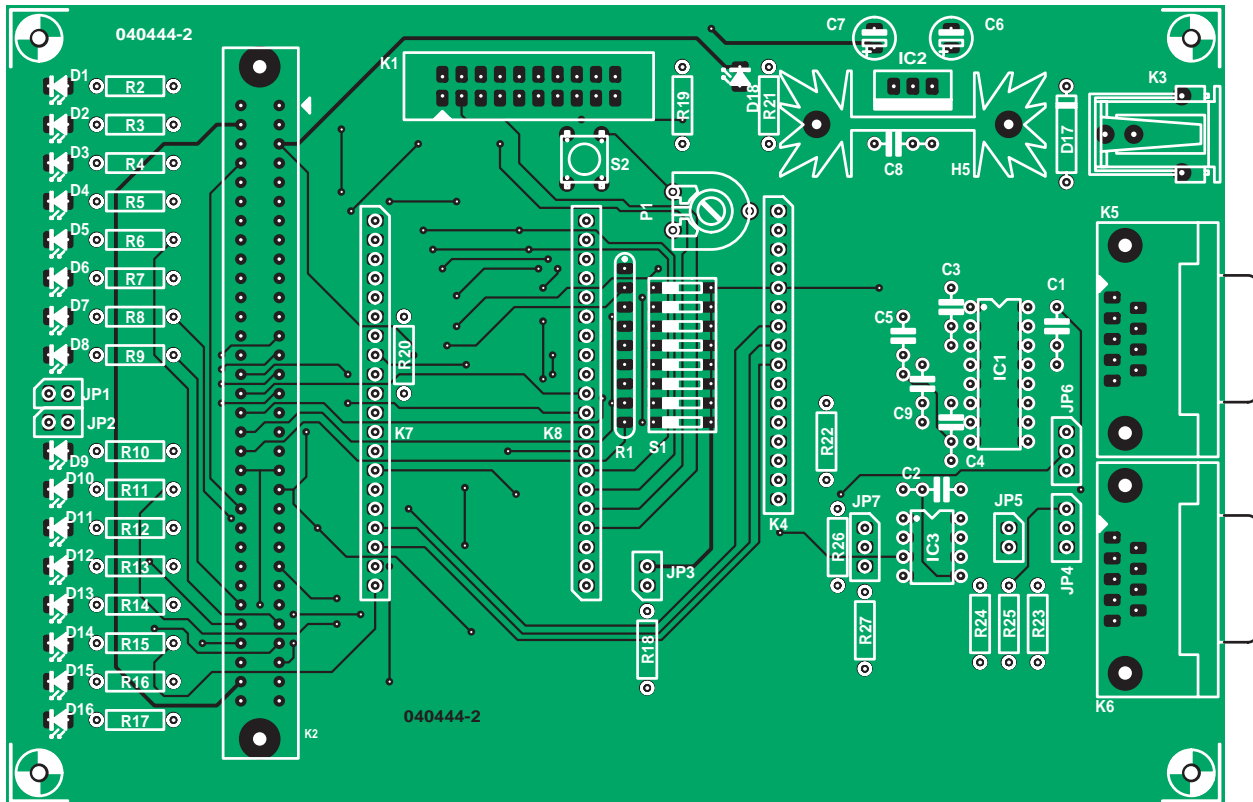


Figure 2a. PCB design for the Eurocard-size ARMee motherboard.

COMPONENTS LIST

ARMee motherboard

Resistors:

R1 = 8-way 10kΩ SIL array
 R2-R17 = 1kΩ
 R18, R19, R20, R26, R27 = 10kΩ
 R21 = 1kΩ5
 R22 = 33Ω
 R23 = 120Ω
 R24, R25 = see text
 P1 = 10kΩ preset

Capacitors:

C1-C5, C8, C9 = 100nF
 C6 = 100µF 25V, radial
 C7 = 10µF 25V, radial

Semiconductors:

D1-D16, D18 = LED, low current, 3mm
 D17 = 1N4007
 IC1 = MAX3232CPE
 IC2 = 7805
 IC3 = MAX3082CP

Miscellaneous:

JP1, JP2, JP3, JP5 = 2-way pinheader with jumper

JP4, JP6, JP7 = 3-way pinheader with jumper

K1 = 20-way boxheader

K2 = 2x32-way DIN41612AC connector
 K3 = mains adapter connector, PCB mount

K4 = general purpose LCD module, 2x16 characters

K5, K6 = 9-way sub-D socket (female), angled, PCB mount

K7, K8 = 20-way SIL socket

S1 = 8-way DIP switch

S2 = pushbutton, PCB mount, e.g., DTS6 Heatsink, Fischer SK104 25,4 STC

PCB, order code **040444-2**

Disk, project software (LEDTest), order code **040444-11** or Free Download

the LPC210x can be programmed using the built-in serial boot loader. This allows the 5 digital I/O lines used by the JTAG interface to be used for something else.

Serial Boot Loader

To use the serial boot loader jumper JP3 needs to be fitted. JP3 pulls P0.14 to

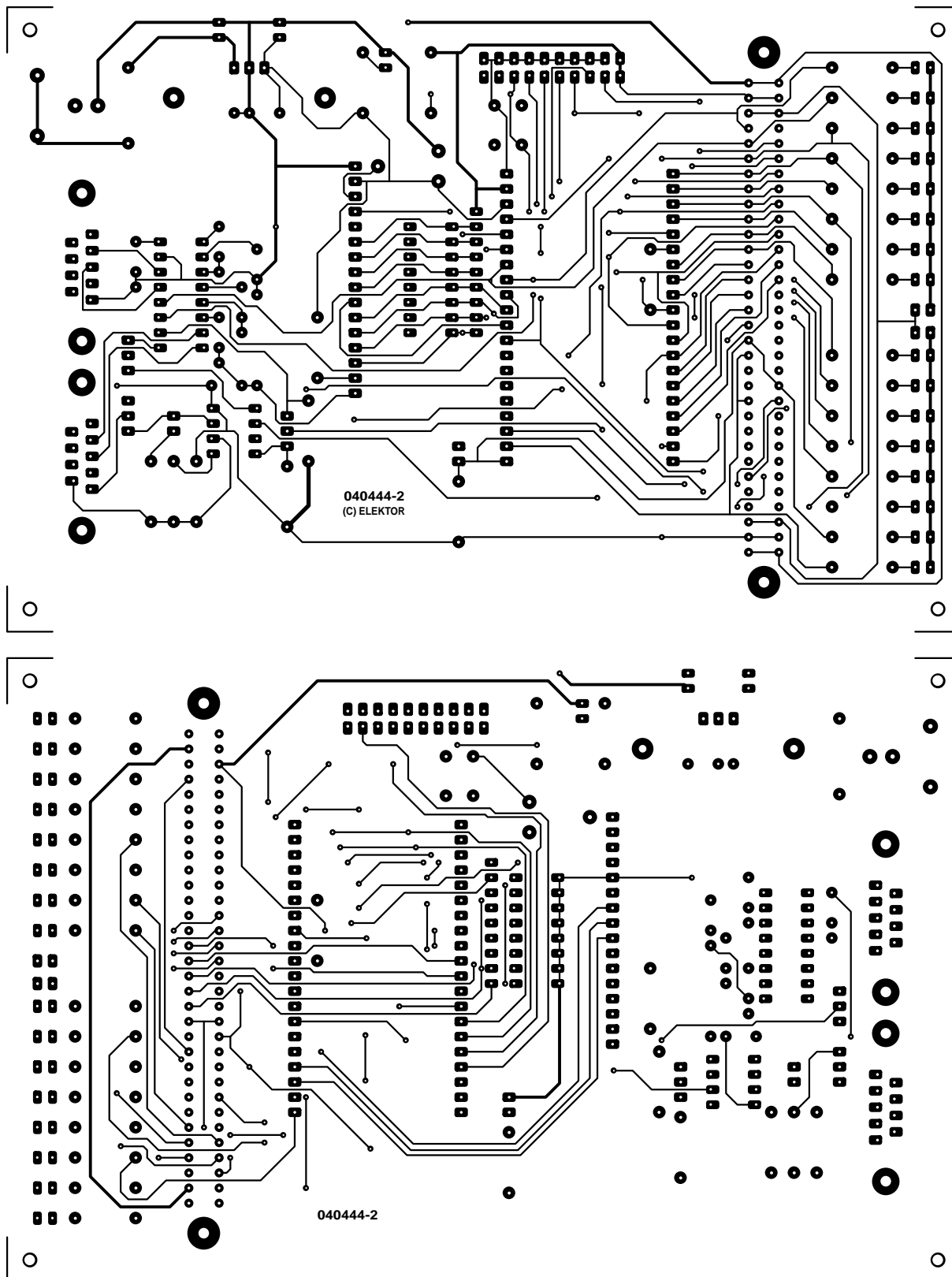
ground and releasing the LPC210x from reset forces the LPC210x to execute an internal serial boot loader program from the top 8 k of its Flash memory.

Using a free PC program provided by Philips Semiconductor (**Figure 4**) and a serial cable connected between a host PC and UART0 of the LPC210x (RS232 Port 1 on the ARMee board)

allows the programming of the LPC210x to take place.

Using GNU GCC with ARM Processors

“Why do we want to use the GNU GCC compiler?” you may ask? Well for a start the tools by virtue of their beginnings and the method of their commu-



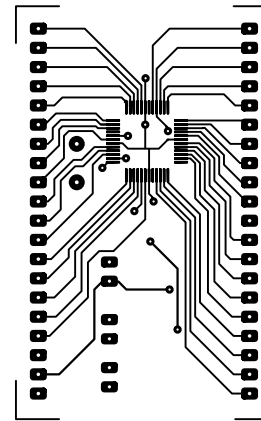
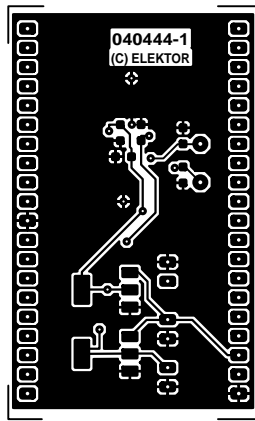
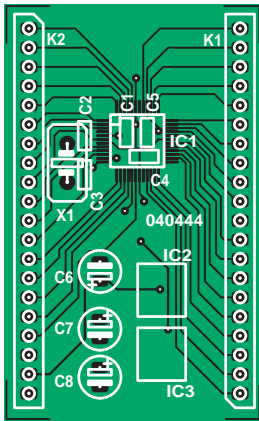


Figure 2b. PCB design for the detachable LPC210x processor board.

COMPONENTS LIST

LPC210x daughterboard

Capacitors:

C1,C4,C5 = 100nF, shape 0805
 C2,C3 = 33pF, shape 0805
 C6,C7,C8 = 10µF 16 V, radial

Semiconductors:

IC1 = LPC210x (2104/2105/2106)
 IC2 = LMS8117AMP-3.3
 IC3 = LMS8117AMP-1.8

Miscellaneous:

K1,K2 = 20-way SIL pinheader
 X1 = 16MHz quartz crystal
 PCB (bare), order code 040444-1
 PCB (ready populated, with LPC2106),
 order code **040444-91**

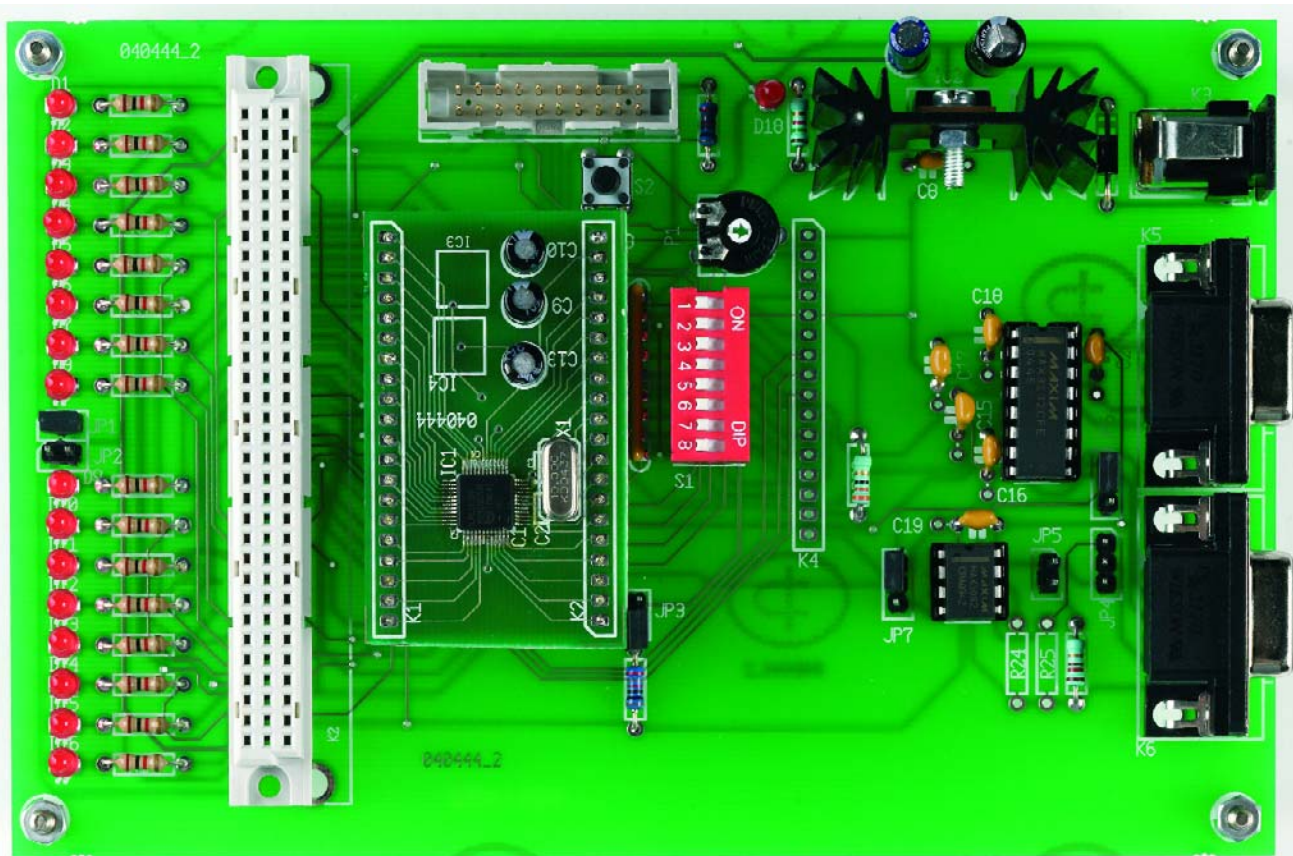


Figure 3. Our finished and tested prototype of the ARMee Development Board. The processor module can be migrated to the target application!

nal user support they are essentially provided for free at the Free Software Foundation (FSF) web site. These tools are made available through the GNU project under the GNU general public license. Historically, the software from GNU/FSF was aimed at the UNIX operating system and GNU is short for 'GNU's Not Unix', while GCC is short for 'GNU Compiler Collection'.

GNU GCC can be found on dozens of different host platforms from Linux, Microsoft Windows, UNIX and Solaris and many others. GNU GCC has been targeted to produce executables for many different microprocessor architectures ranging from Intel/AMD x86,

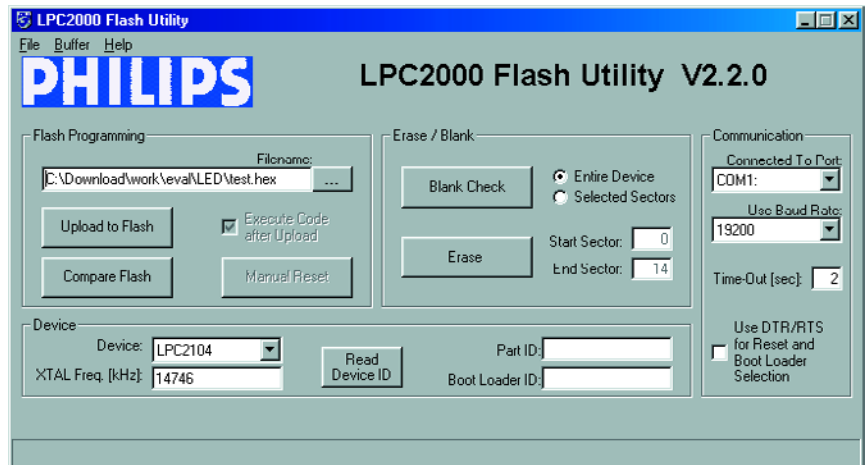


Figure 4. Flash Downloader from Philips.

Simple Input / Output Bus – SIOB

The Simple Input / Output Bus (SIOB) is about as simple as an expansion bus can be. 32 digital I/O signals are brought to a DIN41612 board connector and made available as an expansion port. The digital I/O lines may be either inputs or outputs dependant on the type of expansion interface required. The programmer of the development board is responsible for configuring the data direction register of the microcontroller according to their digital I/O requirement.

In addition to the digital I/O signals the SIOB also offers serial expansion in the form of a SPI and I²C interface. A small number of SPI chip select signals are also present on the bus. The presence of the SPI and I²C interfaces will allow the development system to be expanded in many ways. For example a system could be expanded with I²C based Analogue to Digital (ADC) and Digital to Analogue (DAC) converters or SPI driven digital I/O.

The bus is completed by the +5 V and +3.3 V power signals, clock signal, $\overline{\text{RESET}}$ and RESET.

One final note on the SIOB, the signal pins of the LPC210x are not only used for digital I/O but have other functions such as UART, I²C, SPI, PWM, Timer Capture, JTAG debugging etc. Where a signal pin is used to say drive a UART interface it will no longer be available as a digital I/O signal and thus no longer be available for the SIOB. If a system requires the use of the UART, JTAG debugging and LCD interfaces then there will be very few digital I/O signals available for the SIOB. In such cases system expansion is best done through the I²C or SPI interfaces available on the SIOB.

Motorola 68x00, PowerPC, ARM7 and ARM9. GNU GCC also covers several microcontroller architectures, some of these being TI's MSP430, Atmel's AVR and Motorola's 68HC11.

Another good reason is bugs, software bugs. The GNU tool chain gets updated regularly, meaning any reported problems are fixed relatively fast.

Your next question might be "What's the downside of using GNU GCC?" Well, for a start it's the staggering range of hosts and targets available. You have to check if GCC is available for your host platform, next check if your target processor is available and finally decide if you want to build (i.e., compile) the GCC yourself or want a pre-built binary. However, for the most part you will find the most common target architectures available for both Win32 and Linux hosts as both source code and binaries. But one word of warning, don't expect to find a fancy

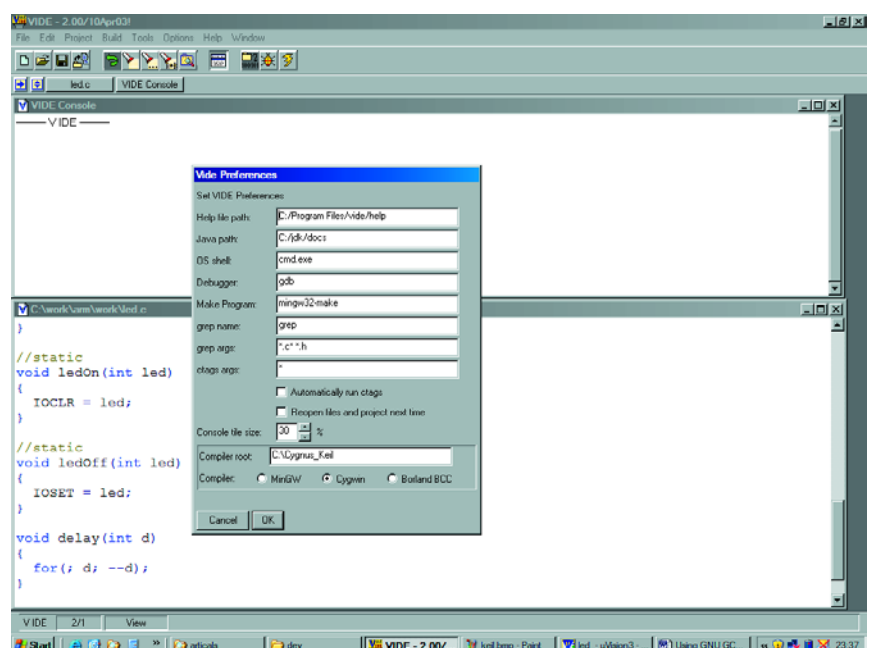


Figure 5. VIDE Programmer's Editor.

More ARM binutils.

addr2line	Converts addresses into filenames and line numbers.
ar	Archiving utility (creating, modifying and extracting from archives)
gprof	Displays profiling information.
nlmconv	Converts object code into an NLM.
nm	Lists symbols from object files.
ranlib	Generates an index to the contents of an archive.
readelf	Displays information from any ELF format object file.
size	Lists the section sizes of an object or archive file.
strings	Lists printable strings from files.
strip	Discards symbols.

installation program or an Integrated Development Environment (IDE) — this is 'storm trooper' DIY development. If you want a fancy installer and IDE check out one of the companies that offer pre-configured, ready to go versions of the GNU tools. GNU-X tools from Microcross and GNUPro from Redhat are two examples of commercial GNU tools.

If you wanted to build your own binaries then download them from the

FSF/GNU web site and follow the instructions provided on the site. However, we will take one of the many pre-built binaries and use it to compile our C code with.

For our example we will use the **GNU GCC version 3.4 Win32 pre-built binaries** from www.gnuarm.com. You can select the Linux host if you wish. Other pre-built binaries are available on the web, Google a search with [GNU], [ARM] and [binaries] and see how

many hits you get. So let's download the binaries from GNUARM and install them on our PC.

GNU C Compiler

Go to the GNUARM sub-directory you installed, look down the list of program files and you will see a program called *arm-elf-gcc.exe*. Similarly, if you were using a microprocessor based on the SH3 architecture you would expect to find *sh3-elfs-gcc.exe* in the sub-directory.

Now, *arm-elf-gcc* is a GNU GCC compiler targeted for the ARM architecture using the Extended Linker Format (ELF) object file format. If you were using the Common Object File Format (COFF) you would find *arm-coff-gcc.exe*. ELF and COFF are two different formats of file structure that the GCC can save the compiler object output to.

Our first task is to make sure the GCC compiler has been correctly installed. A quick test is to display the version message from the GCC compiler. Open a Command Window or MS-DOS window and type the command *arm-elf-gcc -v*. If you don't see the GCC version message, then you most likely need to add the GCC directory to our environment path.

To use the compiler, open a DOS window and type

```
arm-elf-gcc -c ledtst.c
```

The actual program is found in the **'LEDTest – an example' inset**. An archive file containing the assembly code, linker and compiler result files is available as a **free download** with this article (**040444-11**).

The GCC compiler has a raft of options we can invoke, too many to list in this article! Some of the main command line options are:

- `-c` for compile only and do not link the object file;
- `-Wall` to enable every warning and error conditions the compiler will produce;
- `-o filename` to change the file name of the object file produced;
- `-S` to produce an assembly listing interspaced with the original C code;
- `-g` to include debugging information.

The command

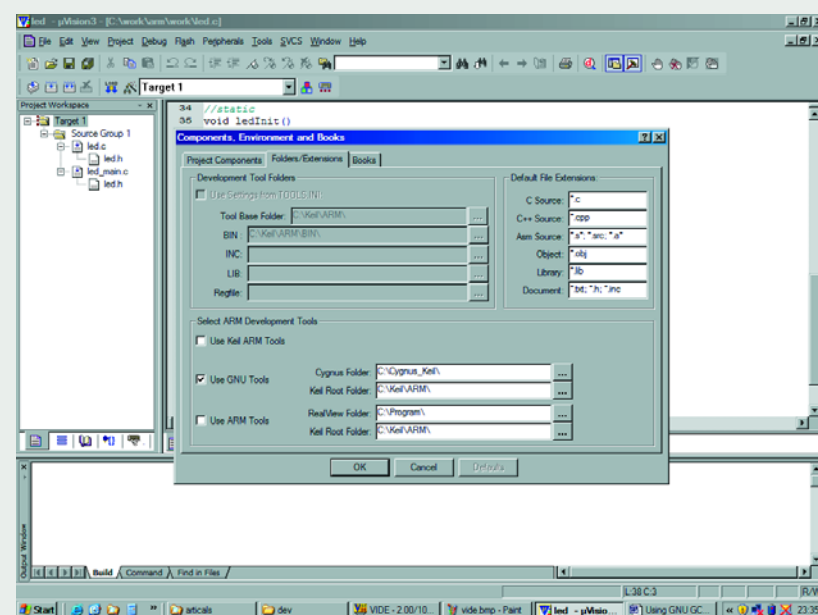
```
arm-elf-gcc -Wall ledtst.c -o ledtst
```

will faithfully compile *ledtst.c* with all compiler warnings enabled and pro-

Commercial offerings

If you are happy to provide a few personal details to Keil you can freely download their uVision IDE and a pre-built copy of GNU tools. These tools are unrestricted for generating code with the only limitation (at this time) is that the debugger is limited to 16 k program size. The Keil tools are based around their uVision 3 IDE, the **screendump** shows the IDE configured for GNU GCC.

Redhat and Microcross provide pre-built GNU tool chains for your chosen target and host platform. Rowley Associates also provide a commercial development environment based on the GNU tool chain but using their own libraries and IDE.



Configuring Keil uVision IDE for GNU GCC.

duce an object file called *ledtst.o*. When we invoke GCC compiler, it normally does the preprocessing, compilation, assembly and linking. However we can tell the compiler not to link any of the object files by using `-c` option. Where this is done we can use the GNU linker to combine our object files into a single output file.

GNU Linker

The object files produced during the compilation stage are by themselves incomplete. We still need to resolve internal variables and function references for the object files and this is where the linker comes in. The linker's job is to combine the object files and resolve any unresolved symbols and function references.

The command line GNU linker *ld* (or *arm-elf-ld*) takes the names of the object files to be linked together as arguments, as well as any options and outputs a single relocatable object file. If we had several files to compile and link we might type:

```
arm-elf-gcc -c module1.c
arm-elf-gcc -c module2.c
arm-elf-gcc -c module3.c
```

```
arm-elf-ld module1.o module2.o module3.o -o prog1
```

There are no memory addresses assigned to the code and data sections in the relocatable object file. To do this we need a locator program that will assign physical memory addresses to each code and data sections. The GNU linker has a built-in locator function that can locate the code and data sections of the program to specific areas of memory, which results in an absolutely located binary image.

We can now download this binary image to our development system.

The GNU linker includes a scripting language that can be used to control the linking process and the exact order of the code and data sections within the relocatable program. We can also assign memory address to the code and data sections. We call a script file by using the command line option `-T` followed by the name of the script file, for example:

```
arm-elf-ld -Tlpc2106-rom.ld
```

GNU Make

The make file is used to mechanise the compilation process. In essence, the make utility follows a user defined set of rules and these rules are processed

in sequence. For instance, instead of typing the four command lines for the linker example shown previously we would type and save these commands into separate text file *makefile* or another suitable filename. To compile this example we would type:

```
make makefile
```

The Make utility will determine what source code files have changed since the last time it was run and will only compile those changed files. You may also include conditional compilation choices by making use of makes' pre-processor functions.

Debugging

The GNU GCC tool chain also comes with a command line debugging tool called GDB.

GNU GDB is one of those tools whose operation and large number of options could fill an entire book, let alone a single article. We do not intend to use GDB with the ARMee Development Board, instead you will be using the Flash downloader provided Philips (Figure 4). Therefore, for this article we skim over most of GDB functionality but those readers wanting more information on GNU GDB are referred to the web where lurks Bill Gatliff's excellent introduction.

In essence GDB is a source debugger. It is connected to a target board through a serial connection where it communicates and interacts with a monitor type program on the target board. To launch GDB type:

```
arm-elf-gdb ledtst
```

Once GDB is running you will be working within the GDB console where the debugging commands will be typed. Next, you can load your program to the target board by using the command

```
(gdb) load
```

Once loaded, you can step through your source code, set break points, investigate what functions were called to get to the current source line or use one of the other debugging options GDB offers.

Other tools

The GNU tool chain also includes a number of other tools, the *binutils*, which we can use during our development. Two of the most useful tools are *objcopy* and *objdump*.

The *objcopy* (or *arm-elf-objcopy*) utility is used to translate object files from one format to another format. For example, we can translate a file from binary image into s-record format.

```
arm-elf-objcopy -O srec ledtst.o ledtst.s19
```

while the *objdump* (or *arm-elf-objdump*) utility can be used to disassemble the object files. With the disassembled file you can see where the linker has located the various text, data and bss sections of our object file.

```
arm-elf-objdump. —disassemble ledtst.o
```

You may also add source code to the

Start-up file

To use our compiled C program we first need to initialise the processors stack pointers and memory settings. A separate object file called the start up file usually does this. Typically a start up file will be called *startup.s*, *boot.s* or *crt0.s*. The start up file is linked to the rest of the compiled object files during the program linking stage. Check the example files from the LPC2000 group on Yahoo for *crt0.s* or *boot.s*, which are typical examples of ARM start up files.

The start up code must provide the following:

1. Disable all interrupts.
2. Copy initialised variables from ROM to RAM.
3. Zero uninitialised data variables.
4. Allocate space for and initialise stack area.
5. Initialise the stack.
6. Create and initialise the heap area.
7. Enable interrupts.
8. Jump to main ().

disassembled output by adding the option `-S` to the command line. If you want to view the symbol table of our object file, simply include the `—syms` option to the command line.

Some of the GNU other *binutils* are listed in **Table 1**.

IDE

When installed, GNU doesn't provide us with any form of integrated development environment (IDE). To resolve this we could install VIDE (**Figure 5**), one of the more popular IDEs that can be used with the GNU tool chain.

VIDE provides us with a programmers text editor and a GUI for running the GNU tools. Other IDEs, both free and commercial, are available for us to use with our GNU tools. Do a search of the web to see what's out there, or use the Forum on our website to get in touch with fellow ARM twisters.

(040444-2)

References:

1. PICee Development System, Elektor Electronics February 2003.
2. AVRee Development System, Elektor Electronics March 2003.
3. 89S8252 Flash Microcontroller Board, Elektor Electronics December 2001.

For further reading

1. Programming Embedded Systems in C and C++, Michael Ball; O'Reilly (www.oreilly.com)
2. An Introduction to GCC, Brian J. Gough, foreword by Richard M. Stallman; pdf copy at <http://www.network-theory.co.uk/gcc/intro/>

Websites:

- www.geocities.com/tonydixon2k1/index.html (ARMee support website)
- www.gnu.org (Main GNU web site)
- www.fsf.org (Free Software Foundation web site)
- www.gnuarm.com (GNU Binaries for ARM processors)
- <http://groups.yahoo.com/group/lpc2000/> (Philips LPC2000 ARM Discussion Group)
- <http://groups.yahoo.com/group/gnuarm/> (GNUARM Discussion Group)
- www.objectcentral.com/videl.htm (VIDE Windows IDE)
- www.billgatloff.com (Excellent GNU tutorials by Bill Gatloff)
- <http://www.dreamislife.com/arm/> (LPC2106 tutorials by Senz)
- www.keil.com (GNU GCC and Commercial C-Compilers)
- www.rowley.co.uk (GNU GCC and Commercial C-Compilers)
- www.redhat.co.uk (Commercial GNU GCC tool chains)
- www.microcross.co.uk (Commercial GNU

LEDTest - an example using the ARMee board

In the embedded development community the first program you usually write with a new compiler or target platform is a LED-flashing program. This is as ubiquitous as the "hello world" program is for PC development and we will be no different. Our example program will consist of a C program that will flash one the LEDs on the ARMee development board. See Listing 1!

Include the Philips ARM LPC2100 C header file, ARM start up file (boot.o) and GCC linker script (lpc2106-rom.ld) in the same directory as our LED test program. From the command line prompt, type the following GCC commands:

```
arm-elf-gcc -c ledst.c
arm-elf-ld -Tlpc2106-rom.ld -nostartfiles, -nostdlib -s -o led boot.o ledst.o
arm-elf-objcopy —output-target ihex test test.hex
```

We should now have a compiled and linked hex file ready for sending to the ARMee Development board. To save retyping the GCC commands, we could type the above GCC commands into a text editor and save them as a batch file ledst.bat or create a Makefile.

Finally, instead of using the GDB debugging tool included with GNU GCC we will opt to use the simple Flash download provided by Philips Semiconductor, Figure 4. Select the filename of our LED hex file and upload to the target board.

Once uploaded, press reset on the ARMee board and LED D1 should start to flash.

Congratulations! You've have successfully compiled, linked and uploaded our test program using the GNU GCC compiler and tools.

```
#include "LPC2000.h"

void Delay (unsigned long a)
{
    while (--a!=0);
}

int main (void) {

    /* System Init */
    /* Init PLL */
    SCB_PLLCFG = 0x23;
    SCB_PLLFEED = 0xAA; SCB_PLLFEED = 0x55;

    SCB_PLLCON = 0x01;          /* Enable PLL */
    SCB_PLLFEED = 0xAA; SCB_PLLFEED = 0x55;

    while ( !(SCB_PLLSTAT & PLOCK) ); /* Wait for PLL to lock */

    SCB_PLLCON = 0x03;          /* Connect PLL as clock source */
    SCB_PLLFEED = 0xAA; SCB_PLLFEED = 0x55;

    /* Init MAM & Flash memory fetch */
    MAM_MAMCR=0x2;             /* MAM = flash */
    MAM_MAMTIM=0x4;
    SCB_VPBDIV=0x1;           /* PCLK = CCLK */

    /* Init GPIO */
    GPIO_IODIR = 0x00000001;    /* P0.0 as output */
    GPIO_IOSET |= 0x00000001;  /* LED off */

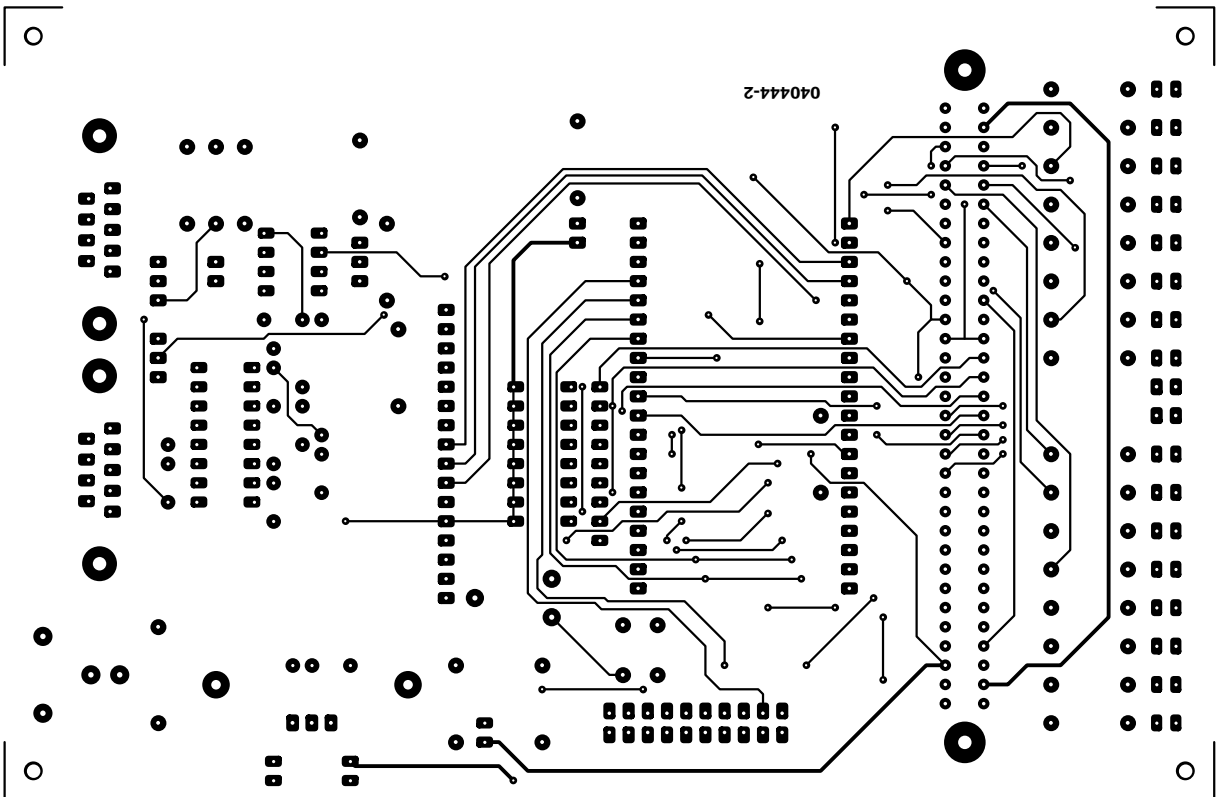
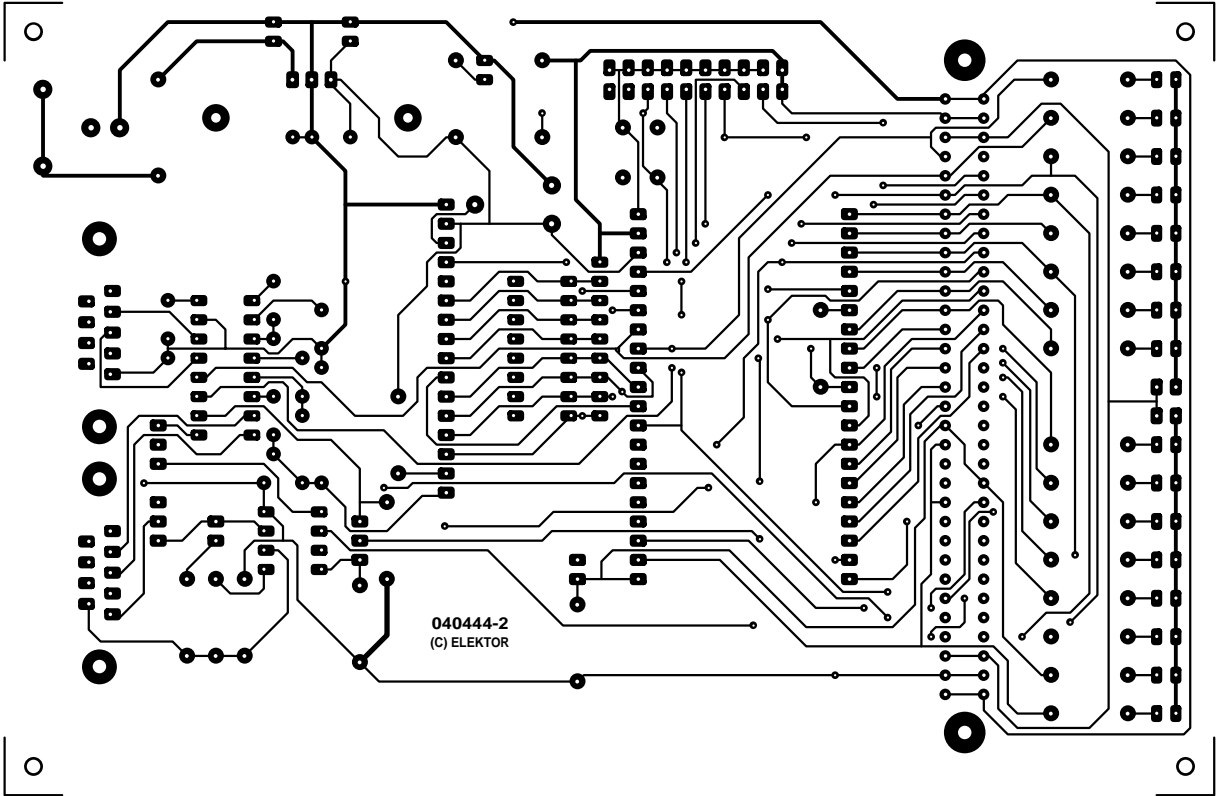
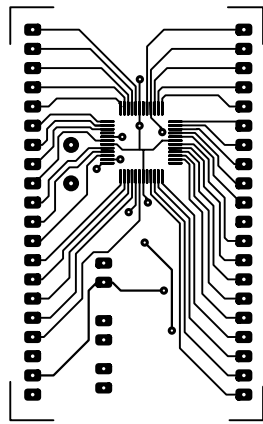
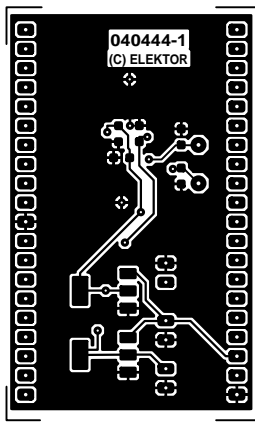
    /* main loop */
    while (1) {

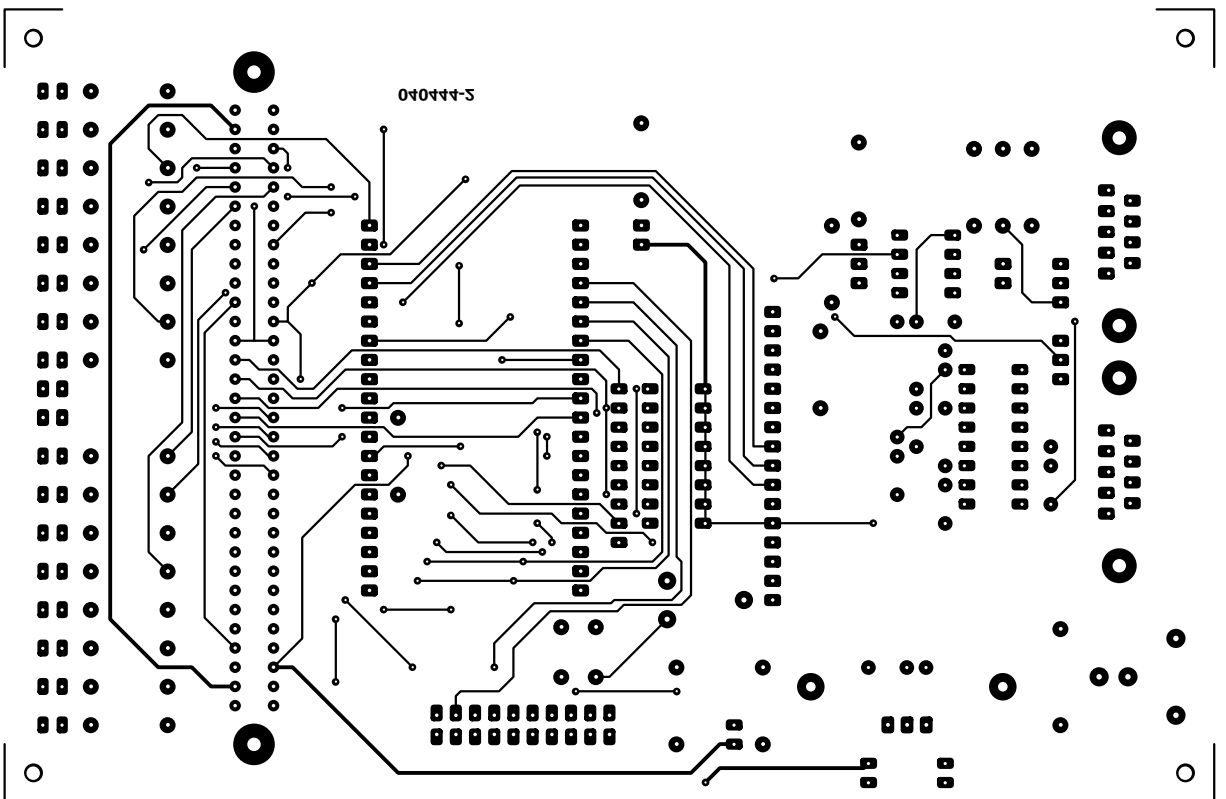
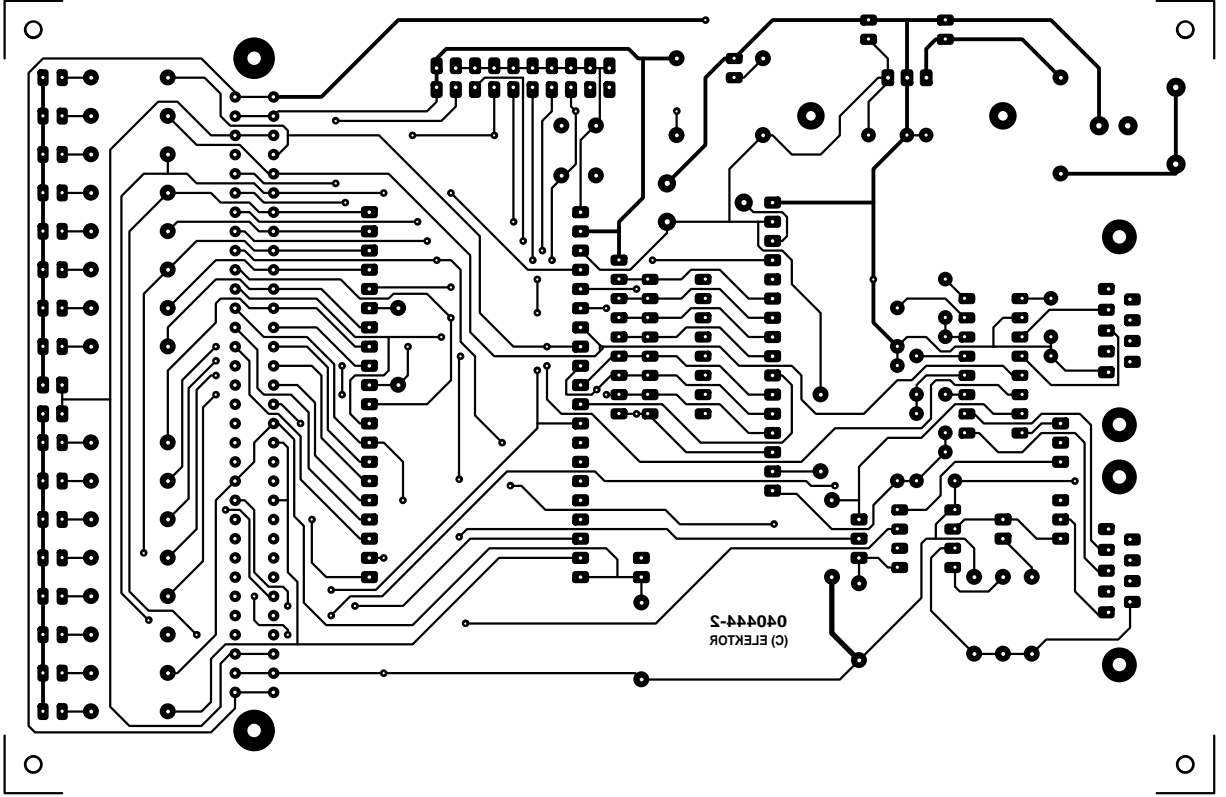
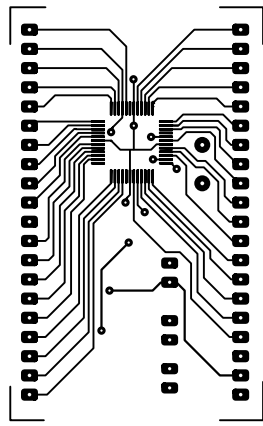
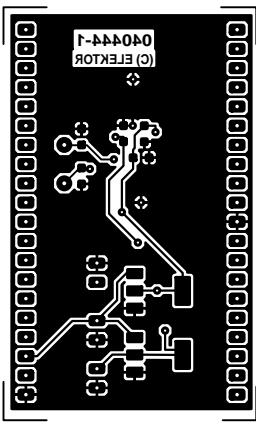
        GPIO_IOSET |= 0x00000001; /* LED off */
        Delay (1000000);

        GPIO_IOCLR |= 0x00000001; /* LED on */
        Delay (2000000);

    }

} /* End of main () */
```



SC Analyser 2005

Semiconductor Device Tester

Michel Waleczek

Almost every electronic circuit contains bipolar transistors, FETs or diodes. Most electronics hobbyists have a supply of such components removed from old circuit boards. A tester that can be used to sort out the leads and measure the characteristics is thus a handy tool.



If the markings on the device can still be read, it's usually possible to look up its technical characteristics in a data book. But if the markings are vague or completely missing, you can only guess. Using this particularly handy tester, you can quickly identify the most commonly used types of semiconductor devices (bipolar transistors, JFETs, MOSFETs and diodes), including SMD components. Besides identifying the leads, the tester calculates various parameters, such as H_{FE} for bipolar transistors, V_{TH} , I_{DSS} and R_{DSON} for JFETs, the threshold voltage for MOSFETs, and the forward current/voltage and leakage current for diodes. All of this information is clearly displayed on an LCD (compatible) screen.

Operating principle

Each of the three leads of the unknown device is connected to ground or +5 V via resistors with known values. The following resistance values can be used: 100 Ω , 1 k Ω , 5.6 k Ω , and 100 k Ω . For each configuration, three voltages are measured using a PIC16F876 microcontroller.

The microcontroller always starts with a quick, rough measurement to determine whether the device is a bipolar transistor. This is done by connecting two of the three transistor leads to ground while the third lead is connected to +5 V via a 5.6-k Ω resistor. The microcontroller measures the voltage across the resistor and stores the measured value. Two more measurements of this sort are made with different lead sequences, with each measurement being made at the junction of the 5.6-k Ω resistor and the transistor. This yields three values that have a specific relationship to the transistor type.

Table 1 shows the values that should theoretically be measured for NPN and PNP transistors. Here the minus sign corresponds to a connection to ground via a 100- Ω resistor, and the plus sign stands for a connection to +5 V via a 5.6-k Ω resistor. An NPN transistor gives two values of approximately 5 V and one of approximately 0.7 V, while a PNP transistor gives a single value of 5 V and two values of 0.7 V.

The first test is also sufficient to identify the base lead of the transistor, since it is the lead whose value differs from the values for the other two leads. Once

Table 1. Initial measurements				
Type	E	B	C	measured value
NPN	-	-	+	5 V
	+	-	-	5 V
	-	+	-	0.7 V
PNP	-	-	+	0.7 V
	+	-	-	0.7 V
	-	+	-	5 V

the transistor has been identified in this manner, it is tested for current gain. As the positions of the emitter and collector are not known, the current gain is measured for each of the two possible combinations. The ultimate value is taken to be the greater of the two measured values.

If the measured voltages do not match any of the combinations in Table 1, the device is subjected to special tests for other types of components (MOSFET, diode and JFET). To test whether the device is a MOSFET, the current gain is measured in a similar manner for all six of the possible lead arrangements. However, some bipolar transistors also yield results that differ from those shown in Table 1. This primarily occurs with transistors having a protection diode between the collector and the emitter. For such transistors, the current gain is also measured for all six of the possible arrangements.

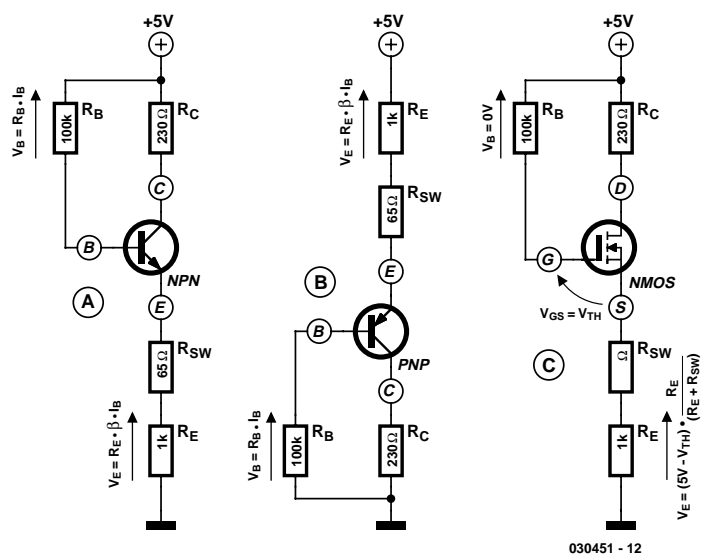


Figure 1. Configurations for measuring the value of β for a transistor and the threshold voltage of a MOSFET.

Limitations

To avoid possible misunderstandings, we must say up front that the SC Analyser 2005 cannot be used to measure thyristors or Darlington transistors.

Measuring current gain for bipolar transistors

The measurements described above have identified the base lead of the transistor, and now the other two leads are identified by connecting the transistor in a common-collector configuration if it is a bipolar transistor, or in a source-follower configuration if it is a MOSFET (see Figure 1).

The gain of the transistor is determined by measuring V_B and V_E . The formulas for this parameter are:

$$V_E = R_E \times (\beta + 1) \times (V_B \div R_B)$$

$$\beta = [(V_E \times R_B) \div (V_B \times R_E)] - 1$$

The circuit can measure gain values over a range of 5 to 999. An N-channel MOSFET (Figure 1c) can be distinguished from a bipolar transistor by the fact that its gate current is practically zero. In this case, the threshold voltage corresponds to the voltage $V_{CC} - V_E$ (for an N-channel type). To be properly identified by the tester, the MOSFET must have a threshold voltage less than 4.5 V, and it must be an enhancement type (which is almost always the case). The other type of MOSFET, which is

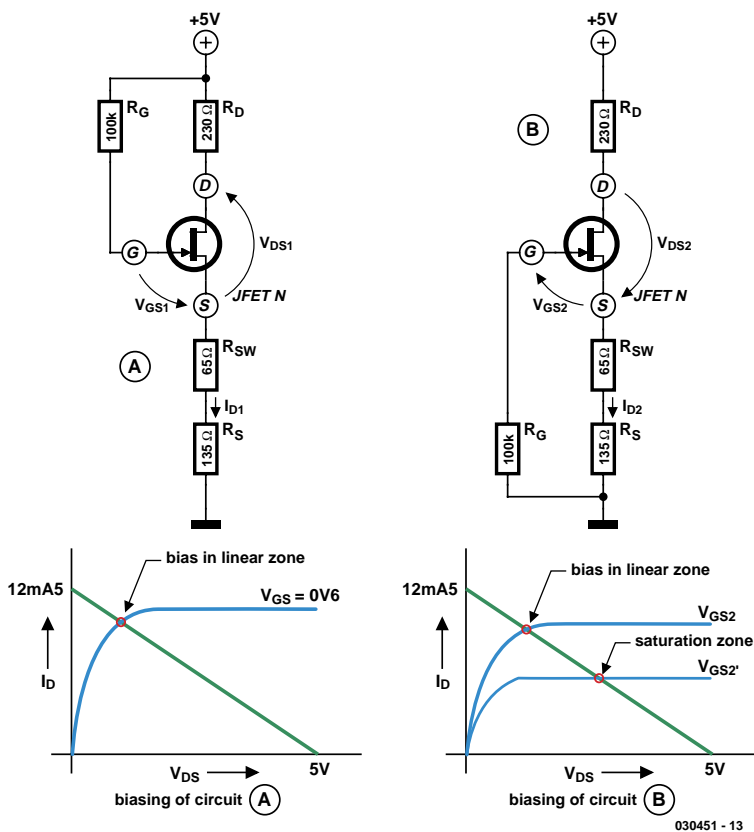


Figure 2. Configuration for measuring JFET parameters.

called a depletion type, is scarcely used any more.

Measuring FET parameters

Field-effect transistors (FETs) can be characterised by a certain number of parameters, but here we are only inter-

ested in three of them: V_{TH} (gate-source threshold voltage), I_{DSS} (drain saturation current), and R_{DSON} (resistance in the full 'on' state). Determining these parameters is more complicated than simply measuring the current gain of a bipolar transistor. The topology of the circuit is hardly suitable for making direct measure-

ments, so an indirect method based on a mathematical model of the FET (the Schichman-Hodges model) is used to determine the values of the above-mentioned three parameters by calculation. Of the three principal parameters of this static model, which is primarily used in Spice simulation, it turns out that only two are actually necessary here, since the third parameter (λ) has hardly any effect on the ultimate calculation (see inset).

In order to determine the values of these two parameters, it's necessary to examine two specific operating points in order to obtain a set of two equations. In the first case, the FET is biased in its linear operating region (**Figure 2a**) by using resistor R_G to force V_{GS1} to approximately 0.6 V. Some FETs with low saturation currents can be biased into their saturation region in this configuration, and in such cases a variation on the circuit shown in Figure 2 is used, but this is not described any further here. The first configuration yields the first set of values for V_{DS1} , V_{GS1} and I_{D1} . The second operating point is obtained by configuring the transistor to operate as a current source (**Figure 2b**), which yields a second set of values (V_{DS2} , V_{GS2} and I_{D2}).

Now things become a bit more complicated, because the second operating point can lie in the linear region or the saturation region. The operating region cannot be determined until the threshold voltage V_{TH} is known. There's thus no other choice than to calculate the value of V_{TH} for each type of operating region and then choose the proper value from the two results by checking the corresponding operating regions. The resistances R_{SW} of the analogue switches are shown in the schematic diagram in **Figure 3**. They have a value of approximately 65 Ω . The exact values are determined using an automatic calibration procedure. The drain resistance, which is approximately 230 Ω , consists of a 100- Ω resistor, the resistance of the analogue switch (65 Ω), and the output resistance of the micro-processor (65 Ω).

Another factor that must be borne in mind is that most FETs are symmetric, which means that the drain and source are interchangeable. It is thus impossible to tell these two leads apart, so the tester can only actually identify the gate lead. The drain and source leads are indicated according to the configuration used for calculating the transistor parameters. Interchanging the

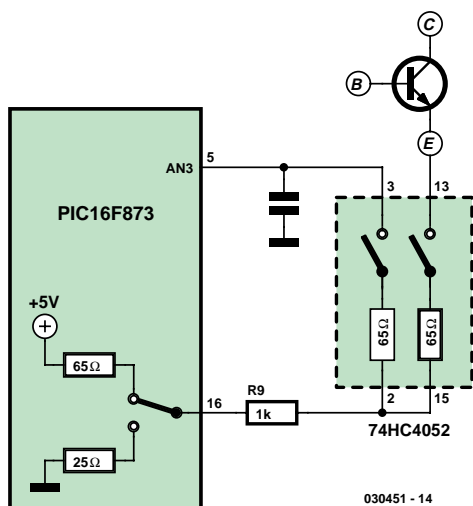


Figure 3. The Kelvin circuit avoids errors due to the internal resistance of the 74HC4052.

source and drain leads will not change the information shown on the display of the SC Analyser 2005, but the results of the calculation always correspond to indicated lead arrangement.

Schematic diagram

The complete schematic diagram of the circuit is shown in **Figure 4**. The circuit draws approximately 6 mA (except for the backlight, which needs around 20 mA), and it is powered by a 9-V battery. A 78L05 voltage regulator

combined with three decoupling capacitors reduces the voltage to exactly 5 V in order to power the PIC16F876, the LCD (with backlight), and the three 74HC4502 multiplexers. The display module communicates with the microcontroller in 4-bit mode via five Port C leads and lead RA5 of Port A. The PIC16F876 is clocked at approximately 1 MHz by the R10/C4 network. The PIC16F876 differs from the 'classic' 16F84 by having an on-board A/D converter, which is used in this circuit. The Reset pin can be tied

directly to +5 V, since the microcontroller has an automatic reset function. The three control signals for the transistor being tested come from RC4, RC5 and RC6 in Port C. These signals, which have levels of 0 V or 5 V, are routed to the three test terminals via three analogue multiplexers in order to insert resistors with specific values between the microcontroller outputs and each of the test terminals. The resistance values are determined by selection signal pairs RB4 and RB5 for the right-hand signal (J1), RA2 and

Measuring FETs with the Schichman-Hodges model

The static characteristics of a JFET can be represented by a set of formulas that express the drain current as a function of the voltages V_{GS} and V_{DS} of the FET. There are two different formulas, since the model distinguishes between the linear region $0 \leq V_{DS} \leq (V_{GS} - V_{TH})$ and the saturated region $V_{DS} \geq (V_{GS} - V_{TH})$, where V_{TH} is the threshold voltage of the FET.

The full model employs three parameters: β , λ , and V_{TH} . Parameter β is related to the saturation current and the threshold voltage, while parameter λ represents the channel-length modulation factor and is neglected in our calculations ($\lambda = 0$). The mathematical model that we use thus appears as follows:

For the linear region, defined as $0 \leq V_{DS} \leq (V_{GS} - V_{TH})$, the following formula applies:

$$I_D = \beta \cdot V_{DS} \cdot [2 \cdot (V_{GS} - V_{TH}) - V_{DS}] \cdot (1 + \lambda \cdot V_{DS})$$

For the saturation region, defined as $V_{DS} \geq (V_{GS} - V_{TH})$, the following formula applies:

$$I_D = \beta \cdot (V_{GS} - V_{TH})^2 \cdot (1 + \lambda \cdot V_{DS})$$

As we have seen, in the first configuration the FET is in the linear region with an adequately large I_{DSS} , so the first formula becomes:

$$I_{D1} = \beta \cdot V_{DS1} \cdot [2 \cdot (V_{GS1} - V_{TH}) - V_{DS1}] \quad [1]$$

In the second configuration, the FET can be in the linear mode with:

$$I_{D2} = \beta \cdot V_{DS2} \cdot [2 \cdot (V_{GS2} - V_{TH}) - V_{DS2}] \quad [2]$$

or it can be in the saturated mode with:

$$I_{D2} = \beta \cdot (V_{GS2} - V_{TH})^2 \quad [3]$$

In this manner, we obtain a set of two equations in which V_{TH} and β are the unknowns. Depending on the operating region of the FET, we must use either formula [2] or formula [3].

If we assume that the FET is operating in the linear region, formulas [1] and [2] yield:

$$V_{THLIN} = C - (V_{GS2} - V_{DS2})^2 / (V_{DS2} - V_K) \quad [4]$$

where

$$V_K = (I_{D2} \div I_{D1}) \cdot V_{DS1}$$

and

$$C = V_{GS2}^2 + V_K \cdot (V_{DS1} - 2 \cdot V_{GS1}) \quad [5]$$

Consequently, we can calculate the constant B as follows:

$$B = 2 \cdot (V_K - V_{GS2}) \quad [6]$$

If the operating point of the FET is in the saturated region, formulas (1) and [3] yield a second-order equation:

$$V_{TH}^2 + B \cdot V_{TH} + C = 0$$

We now have two solutions for V_{TH} :

$$V_{TH1} = -B + \sqrt{B^2 - 4 \cdot (C \div 2)} \quad [7]$$

and

$$V_{TH2} = V_{THSAT} = -B - \sqrt{B^2 - 4 \cdot (C \div 2)} \quad [8]$$

Only one of these solutions is physically possible, and this is determined by evaluating the two expressions. All that's left to do now is to determine V_1 and V_2 in order to figure out the actual operating region of the FET in the second configuration:

$$V_1 = V_{GS2} - V_{THLIN} - V_{DS2}$$

$$V_2 = V_{GS2} - V_{THSAT} - V_{DS2}$$

If $V_1 > 0$ and $V_2 < 0$, the FET is operating in the linear region and $V_{TH} = V_{THLIN}$.

If $V_1 < 0$ and $V_2 > 0$, the FET is operating in the saturated region and $V_{TH} = V_{THSAT}$.

If the FET is operating in the saturated region, the I_{DSS} parameter corresponds to the drain current for $V_{GS} = 0$. For V_{DS} , we take the value corresponding to the boundary of the saturation region, which is $V_{DS} = V_{GS} - V_{TH} = -V_{TH}$.

Substituting for V_{DS} in equation [2] or [3] yields:

$$I_{DSS} = \beta \cdot V_{TH}^2$$

The value of can be easily calculated by rearranging formula [1] as follows:

$$\beta = I_{D1} \div \{ V_{DS1} \cdot [2 \cdot (V_{GS1} - V_{TH}) - V_{DS1}] \}$$

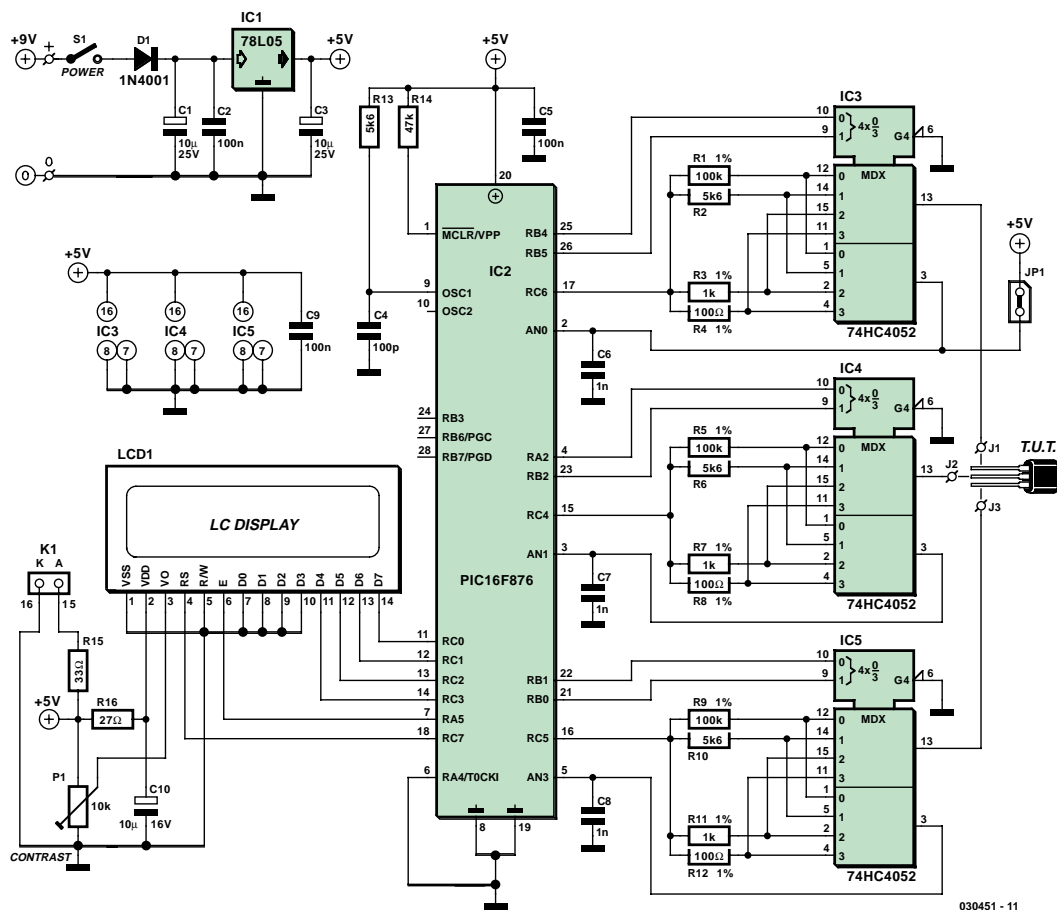
The resistance R_{DSON} corresponds to the slope at the origin of the characteristic curve:

$$V_{DS} = f(I_D) \text{ if } V_{GS} = 0$$

Finally, formula [1] yields the following results when V_{DS} approaches 0 V:

$$I_D = -2 \cdot \beta \cdot V_{DS} \cdot V_{TH}$$

$$R_{DSON} = dV_{DS} / dI_D = -[1 \div (2 \cdot \beta \cdot V_{TH})]$$



030451 - 11

Figure 4. Complete schematic diagram of the SC Analyser 2005.

RB2 for the centre signal (J2), and RB1 and RB2 for the left-hand signal (J3). The voltages present on the test terminals are measured by the PIC15F876 via analogue inputs AN0, AN1 and AN3.

In order to compensate for the effects of the internal resistances of the analogue switches when making current measurements, these measurements are made using a second switch in each 74HC4052 instead of directly at the leads of the unknown transistor. The operation of this arrangement is shown schematically in **Figure 3** with resistor R9 connected in the circuit. The internal resistance of the microcontroller output, which is around 30 Ω, must also be taken into account. Finally, three 1-nF capacitors provide a certain amount of filtering for the measured signals.

Software

The software is written entirely in assembly language and fills a large chunk of the memory space of the PIC16F876. Approximately half of the

occupied space is used for calculating parameter values for FETs. If you would like to program your own microcontroller, you can download the necessary hex code from the *Elektor Electronics* website (www.elektor-electronics.co.uk). The file is found with the pdf download for this article (April 2005). For those of you without Internet access it is also available on floppy disk from Reader Services, the order code is **0304051-11**. Naturally, you can also order a fully programmed microcontroller from Reader Services (order no. **030451-41**).

Construction

The PCB layout and component layout are shown in **Figure 5**. Ensure that the IC sockets, electrolytic capacitors, 5-V regulator and the four ICs are fitted the right way around.

The 1% resistors can be replaced by 5% metal-film resistors carefully selected using an accurate and reliable multimeter. The analogue switches must be 74HC types, since the internal resistance of the normal 4000 series is too

large for this application.

The display module can be fitted on the copper side of the circuit board. To make it easy to connect the display, there is a single-row 16-way pinheader (male) on the circuit board, which mates with a corresponding 16-way female connector on the display circuit board. Here we chose to use a modern type of display called 'PLED' (see the 'OLED and PLED' inset), but you can also use any desired LCD module based on (or compatible with) the Hitachi HD44780, although the pin arrangement may differ from the arrangement for the display used here. Note that the pin arrangement of the display used here is rather unusual, with pins 15 and 16 for the backlight being located next to pin 1.

Beside the test leads with alligator clips, there is also a special test circuit board that is connected to the main circuit board and can be used to simplify testing SMD devices (diodes and transistors). Use three short, flexible leads to connect the SMD holder to the main circuit board, and ensure that the proper lead sequence is maintained (as

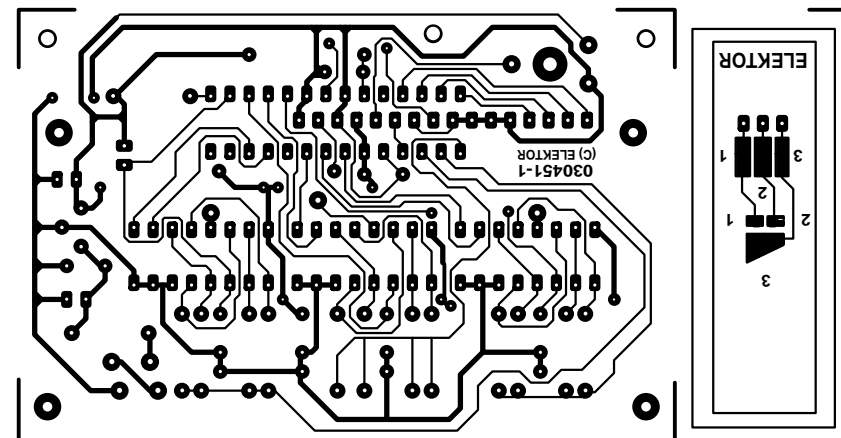
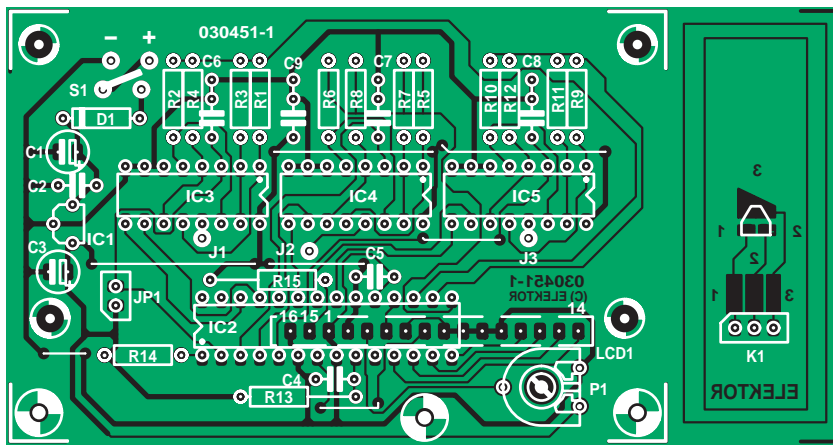


Figure 5. Track and component layouts for the associated printed circuit board. SMD components can be conveniently tested using the small circuit board.

seen from above: contact 1 to J1, contact 2 (middle) to J2, and contact 3 to J3). Although this hardly matters for the measurements, it is important for

identifying the leads. **The middle contact must always be connected to J2.** Naturally, you can also use flexible leads with miniclips. The circuit board

is designed to fit in a standard enclosure with a 9-V battery compartment.

Calibration

After the tester is switched on, a welcome message appears and displays the software version (*SC-Analyser 2005 Elektor Rev. 1.0e*). If you don't see this message, try adjusting contrast control P1 to improve the situation.

The first thing you should do is to calibrate the internal resistances of the analogue switches. If these values are known as exactly as possible, the measurements of the various transistor parameters will be more accurate. If this calibration is not performed, a default value of $65\ \Omega$ is used for each switch. Each time the tester is switched on, the software checks to see whether the calibration has taken place, and if it hasn't, the message 'Cal error' is displayed for approximately one second.

The calibration procedure is easy and runs automatically. To perform the procedure, fit a jumper in position JP1 while the tester is switched off and short all three test leads together, and then switch the tester on. The message 'Cal Remove jumper' will be displayed on the tester. Now you should remove the jumper, and the calibration procedure will start. Three resistances are measured one after the other, and their values are displayed successively. Next, the message 'Short $R_{SH}\ XX\ \Omega$ ' will appear. After this you can disconnect the three test leads from each

COMPONENTS LIST

Resistors:

R1, R5, R9 = $100\text{k}\Omega$, 1%
 R2, R6, R10, R13 = $5\text{k}\Omega$
 R3, R7, R11 = $1\text{k}\Omega$, 1%
 R4, R8, R12 = $100\ \Omega$, 1%
 R14 = $47\text{k}\Omega$
 R15 = $33\ \Omega$
 R16 = $27\ \Omega$
 P1 = $10\text{k}\Omega$ preset

Capacitors:

C1, C3 = $10\ \mu\text{F}$ 25V radial
 C2, C5, C9 = 100nF
 C4 = 100pF
 C6, C7, C8 = 1nF MKT
 C10 = $10\ \mu\text{F}$ 16V radial

Semiconductors:

D1 = 1N4001
 IC1 = 78L05
 IC2 = PIC16F876-20/SP (programmed,

order code **040409-41**)
 IC3, IC4, IC5 = 74HC4052 (HC only!)

Miscellaneous:

S1 = on/off slide switch
 LCD1 = standard LCD with 2×16 characters, e.g., ASI-G-162FS-GF-EWS/W (with backlight) or LCD 162C BL (P-LED with active backlight)
 16-way SIL pinheader with mating connector, or flatcable for connection to display
 JP1 = 2-way pinheader with jumper
 3 mini croc clips*
 3 DIL16 IC sockets
 1 DIL28 IC socket
 Enclosure, e.g., Hammond 1591BTBU
 9-V battery with clip-on leads
 5 wire links
 PCB, order code **030451-1** (see Readers services or www.elektor-electronics.co.uk)
 Disk, PIC hex code, order code **030451-11** or free download from www.elektor-electronics.co.uk

* see text

μC configuration

The microcontroller must be configured with the following options:

- oscillator in RC mode
- watchdog timer (WDT) disabled
- 'timer reset on power on' enabled
- brownout reset disabled
- EEPROM protection disabled
- writing to Flash memory disabled
- debug mode disabled
- code protection disabled

other, and the tester will automatically change to the test mode and display the message 'No component *-*-*'. Calibration can be performed whenever desired by repeating the above procedure (switch off the power, fit the jumper, and switch the power on again).

If you experience any problems, check that the five wire bridges are in fact properly fitted and soldered in place. Also check to make sure that the supply voltage is present at the IC sockets.

Operation

After the welcome message showing the software version (*SC-Analyser 2005 Elektor Rev. 1.0e*), the display should show the following information (**Photo 1**).

The first line shows the transistor type and the value of one of the characteristic parameters. The second line shows the lead arrangement of the transistor and, if relevant, the value of a second parameter.

Bipolar transistor

The first line of the display shows the polarity of the transistor (PNP or NPN), the type of semiconductor material (silicon or germanium), and the parameter H_{FE} . The second line shows the lead arrangement of the transistor and the collector current. The current has a value of approximately 1.5 to 4 mA, depending on the current gain. The SC Analyser 2005 displays the type of semiconductor material ('Ge' for germanium or 'Si' for silicon) according to the measured value of V_{BE} (**Photo 2**).

If the transistor is partially or completely shorted, the associated leads are marked with an 'X' by the SC Analyser 2005. A short is considered to be present if the resistance measured between two leads is less than 50Ω (**Photo 3**).

FETs

The values of the parameters V_{TH} , I_{DSS} and $R_{DS(ON)}$ are shown approximately every two seconds. The drain and source leads are determined arbitrarily by the SC Analyser 2005, but they do correspond to the configuration actually used to determine the parameters. If you swap the drain and source leads of the transistor under test (TUT), you can obtain the values for the reverse configuration, but since field effect transistors have symmetric structures,

OLED and PLED

Organic LED (OLED) and polymeric LED (PLED) are recent developments in LED technology. The OLED effect was discovered in the early 1980s by Eastman Kodak, but it has only recently been put to practical use in commercial applications such as PDAs and MP3 players. The Kodak LS633 digital camera was one of the first devices to be fitted with an OLED screen. Some of the MP3 players from Packard Bell also have an especially nice OLED display.

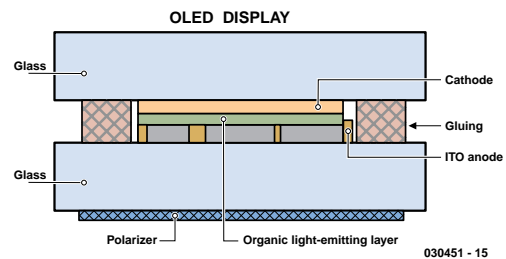
OLEDs are based on LCD technology. A sandwich formed of several layers of organic film is located between two charged electrodes: a metal cathode and a transparent anode. There are four organic films in total: a hole injection layer, a hole transport layer, an emission layer and an electron transport layer. As soon as a voltage is applied, the positive and negative charges combine in the emission layer to generate electroluminescent light. In contrast to LCDs, which need an external source of light (backlight), OLEDs actively emit light.

An even more recent development is the PLED, in which an undoped polymer is sandwiched between two electrodes. The polymer lights up when a voltage is applied. This device can produce a full range of colours and is relatively inexpensive compared with other technologies, such as LCD and OLED.

The advantages of PLEDs are that they require only a small amount of power for high brightness, they are relatively light, robust and fast, and they operate over a relatively large temperature range.

Comment from our design staff:

As you have probably realised, we are quite enthusiastic about the visual properties of the PLED display used in this project. However, we are highly disappointed with the inadequate design of this display. It generates huge current spikes with amplitudes up to several hundred milliamperes, which create a lot of noise on the 5-V supply line. In our opinion, this is caused by a poor hardware design for the display (Version A), with inadequate attention being given to the switching times required by the display drivers. Due to this problem, we were forced to include an extra RC network



*Cross-section of an OLED display.
The structure of a PLED is similar.*

in the +5-V supply.

We do not (yet) know whether PLED displays from other manufacturers suffer from the same shortcoming. If you can get your hands on a different brand of PLED display, it would be interesting to just measure what sort of current pulses it draws.



there shouldn't be any large differences in the measured values.

Considering the method used to make the measurements, the accuracy of the values provided by the SC Analyser 2005 is more than adequate. For I_{DSS} it is $\pm 5\%$, and for $R_{DS(on)}$ it is $\pm 5\%$. However, the measured value of the threshold voltage can differ from the true value by 0.5 V for certain transistors, although in general an accuracy of 50 mV is achieved. A particular type of FET from one manufacturer may yield a value that is very close to the true value, while the same type from a different manufacturer yields a value that differs by several hundred millivolts. In practice, the accuracy is related to the accuracy of the Schichman-Hodges model and reveals the limitations of this model (refer to the inset on this subject).

Threshold voltage

The threshold voltage is negative for N-channel FETs and positive for P-channel FETs. The maximum value is limited to ± 20 V by the software. The resolution is 10 mV for values up to ± 9.99 V and 100 mV for larger values (Photo 4).

Saturation current

The saturation current (with the gate shorted to the source) is shown here. The value of the saturation current ranges from 0 to 99.9 mA. The resolution is 10 μ A for currents up to 10 mA and 100 μ A for currents up to 100 mA (Photo 5).

Drain-source resistance

This is the drain-source resistance when the FET is fully 'on' with $V_{GS} = 0$ V. The measurement has a range of 0–999 Ω (Photo 6).

MOSFETs

The displayed voltage corresponds to the threshold voltage of the MOSFET for a drain current of approximately 2 mA. The measurement range is 0–4.5 V, with a resolution of 10 mV. For proper identification of the device as a MOSFET, the gate leakage current must not exceed 0.5 μ A (Photo 7).

Diodes

Diodes can be connected between the left-hand and right-hand terminals of the tester. The SC Analyser 2005 indicates the positions of the anode and cathode. Three screens are displayed in sequence at an interval of approximately 2 seconds. The first screen

shows the voltage and the current through the diode via a resistance of approximately 400 Ω . The maximum current is thus limited to around 12 mA (Photo 8).

The second screen shows the same information, but this time with a resistance of approximately 5.9 k Ω , which causes the maximum current to be limited to approximately 800 μ A (Photo 9).

The third screen shows the current in the reverse direction and the test voltage. The resolution is 100 nA (Photo 10).

SMDs

You've probably been wondering how to use the SMD portion of the tester. The leads of the component to be tested must make contact with the corresponding copper areas on the circuit board. SMD diodes must be connected between area 1 and area 3. A small plastic rod can be used to properly press an SMD device against the circuit board so that its leads make good contact.

Conclusion

The SC Analyser 2005 is a handy, easily constructed instrument with a large number of features, which can be a quite valuable aid to electronics hobbyists and professionals. A brief list of its possible applications includes searching for an equivalent type, testing device operation, sorting devices, measuring unmarked transistors, and simply determining the lead arrangement without thumbing through a data book. Time to warm up your soldering iron!

(030451-1)

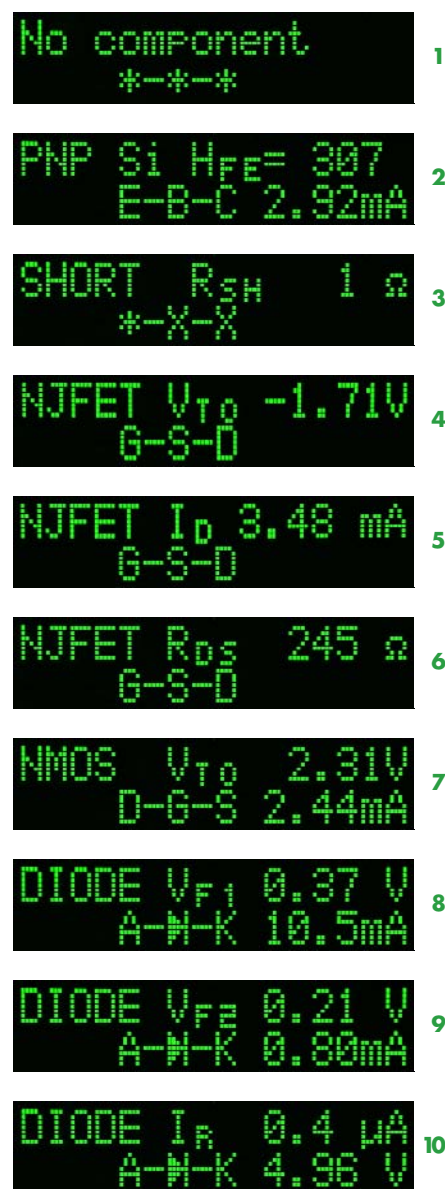
Internet links

Author's website:
www.mwinstruments.com

FAQ site:
www.mwinstruments.com/SCA2005/sca2005.html

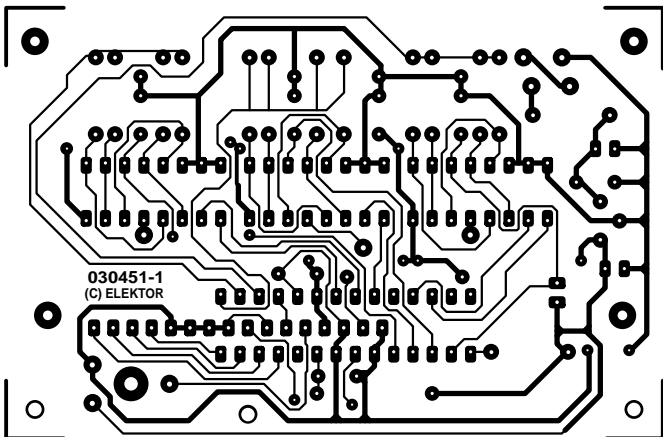
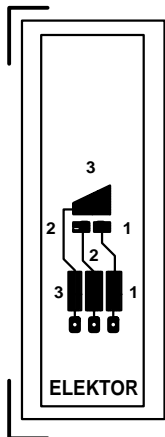
Author's e-mail address:
m.waleczek@mwinstruments.com

Microchip:
www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf

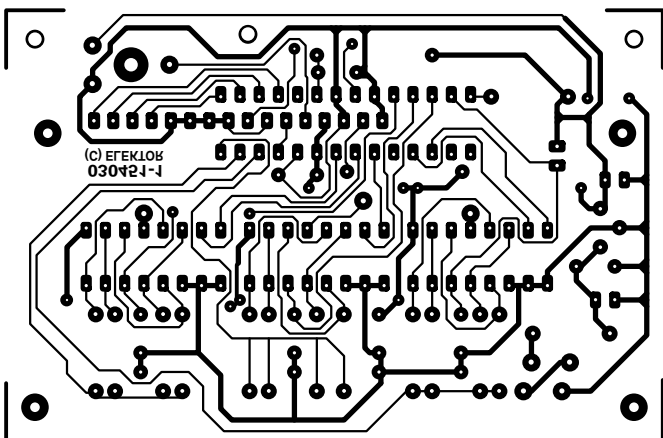
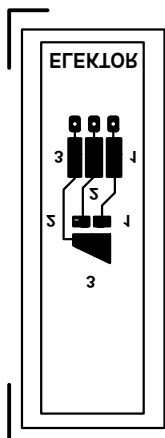


About the author

The author was born in 1965 and studied at the École Nationale Supérieure d'Électronique et de Radioélectricité de Grenoble (ENSERG) in France. He worked as an engineer for seven years in the research department of a large French manufacturer of weighing equipment. Following this he changed to the French branch of Hameg, where he holds the position of Director of Development. Between his travels to the other side of the globe, he manages to find a bit of time to design measurement instruments for electronic hobbyists.



non reflected



reflected

Working with 32-bit micros

Keil ARM7 Development Kit

Paul Goossens

Most microcontroller-based projects in Elektor Electronics employ 8-bit devices, simply because 8-bit micros usually have sufficient capacity to handle the job. Usually..., not always! In a few cases, we need to look out for more powerful alternatives. This review covers a development kit based on a 16/32-bit micro.



Lots of electronics equipment relies on an 8-bit microcontroller for its intelligence. Despite predictions to the contrary, 16-bit microcontrollers never knocked out their 8-bit counterparts. Today's generation of 32-bit controllers, on the other hand, is increasingly seen in the latest electronic apparatus. These powerful number crunchers come into their own in complex equipment like DVD players, camcorders and so on. In most cases, this type of equipment needs to employ some kind of operating system, which adds to the processor load. Although there are operating systems for 8-bit micros, they tend to be rather limited in terms of memory control and memory space.

Today's 32-bit microcontrollers usually do have resources available required by modern operating systems. The reason is, of course, found in the general trend of consumer electronics

growing towards the PC as well as being able to perform several different functions.

A good start

People with a practical interest in electronics (which includes all readers of this magazine) have come to value easy ways of entering new technology. In the case of microcontrollers, one of the best ways to get started is to obtain a development kit, especially if you get your hands on one comprising both hardware and supporting software.

Keil have an excellent reputation for comprehensive development systems for a wide variety of microcontroller families.

Keil's MCB2100 kit discussed in this article is based around the LPC2129 microcontroller from Philips. In addition to the board proper Keil also supply a JTAG interface with a USB connection. The software that comes with the kit is

also complete in every way: compiler, linker, simulator and debugger utilities are supplied in 'evaluation' versions. The limitations are not too serious: the debugger is limited to 16 kBytes of code and the software generated by the compiler may not be used commercially.

Installation

The best thing to do is first install the software in the kit

and then connect up the JTAG interface. This enables Windows XP to automatically recognise the interface and perform a smooth installation. Next, we installed DK-ARM and the GNU compiler. After a restart of the PC, the software was ready for use.

First example

Encouraged by the smooth installation we cheerfully

LPC2129 main features

- 16/32 bit core
 - 256 kB on-chip Flash memory
 - 16 kB on-chip RAM memory
 - In-System Programmable (ISP)
 - Built-in debugger options
 - 2 CAN interfaces
 - 2 serial ports
 - I2C interface
 - 2 SPI interfaces
 - 2 32-bit timers
 - 6 PWM outputs
 - 4 10-bit A/D converters
 - 46 I/O pins
 - PLL controlled clock
- On-board interfaces**
- 2 serial interfaces
 - 2 CAN interfaces
 - 2 pushbuttons
 - 8 LEDs
 - JTAG interface
 - Analogue potentiometer
 - Various signals available for user applications

started to compile the first example and then went on to test it straight away on the board. Unfortunately there were a few hiccups, the software reporting that a certain program could not be found. Eventually, it turned out that the software was making use of the Keil compiler which, apparently, was incomplete. By contrast, the example supplied for the GNU compiler seemed to work like a dream. Although the compile operation finished okay, the program could not be downloaded to the board. To allow this to be done, you first need to configure the Flash utility, which goes by way of the menus *Flash/Configure Flash Tools/Settings*. In the new window that pops up (Figure 2) you need to add the right algorithm.

Having done this, we were able to download the program, but nothing happened. Apparently the processor was still in the reset state. Although it can be started via the debugger, we found that a cumbersome solution. Having clicked around some more menus, we found that the option 'Reset and run' was disabled in the above window. This appears to be the default setting. Having ticked the option, the program was found to start instantly after downloading. All of our further experiments were entirely successful, although the Flash tools had to be configured again with each new project.

The judges' verdict

The development kit contains everything to get started with 32-bit processors.

Not a quick start in our case, but after the first few hurdles nothing should keep you from writing your own applications.

The problems with the Keil compiler that came with the kit remain unsolved and we were unable to compile examples using this tool. Fortunately this is not a serious problem because the GNU compiler works like a dream and Keil's development platform including the debugger also work just fine. The GNU compiler is fully embraced by the platform, so it is not necessary to exit the development shell just to compile a program.

The hardware is pretty Spartan. In view of the raw processing power of the LPC2129 we had expected to see some more I/O. Some processor pins have been brought out to pins for connection of your own extension circuits. Unfortunately, the PCB does not contain the necessary connectors — if you need any you have to solder them yourself.

Overall, the kit is satisfactory if you are after a development platform for 32-bit processors. The JTAG interface in particular is very useful, although we found its practical realisation a bit frugal considering the price of £249 for the kit.

(050004-1)

Kit kindly supplied by Hitex (www.hitex.co.uk)

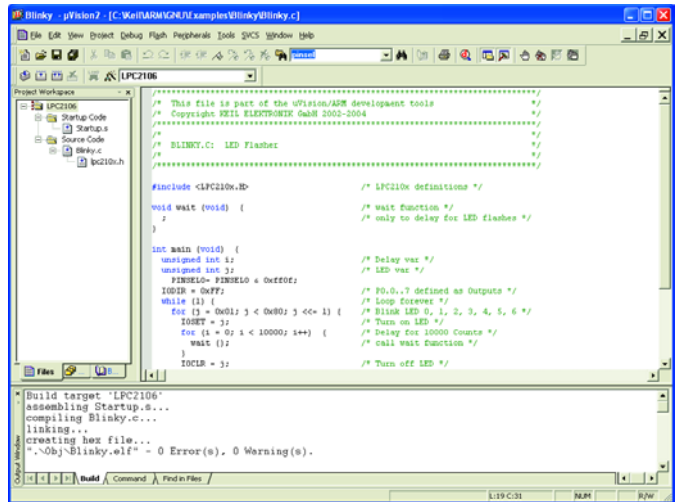


Figure 1. The development platform.

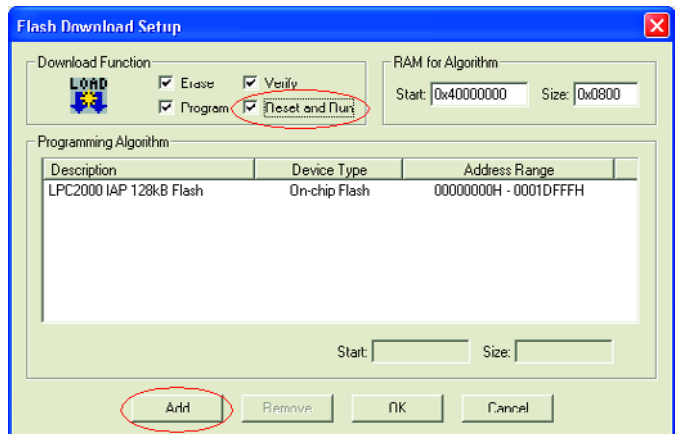


Figure 2. Do remember to properly configure the Flash tool!

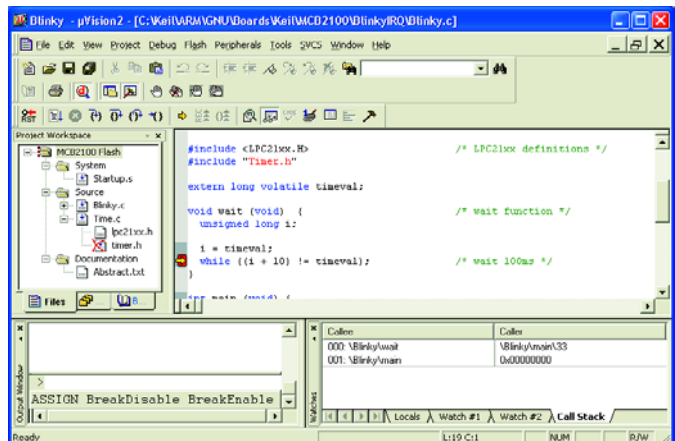


Figure 3. Example showing the debugger in action.

DELPHI FOR ELECTRONIC ENGINEERS

Part 4: creative use of the sound card

The screenshot shows a Delphi IDE window titled 'uAudioControl.pas'. The code defines a Tformat record for a WAV file header and includes procedures for volume control and header preparation. An 'IMPLEMENTATION' section shows the start of a procedure for volume control. Overlaid on the code is a dialog box titled 'AWG Arbitrary Waveform Generator'. The dialog has a title bar with 'AWG' and standard window controls. It features three waveform icons (sine, square, and pulse) and a 'STOP' button. Below the icons are two input fields: 'Amplitude' and 'Frequency', each with a slider and a text box.

```

uAudioControl
uAWG1 uAudioControl
Tformat =
  PACKED RECORD
  ChunkID: ARRAY[0..3] OF Char;    //'RIFF'
  ChunkSize: longword;             //file size - 8 [bytes]
  Format: ARRAY[0..3] OF Char;    //'WAVE'
  SubChunk1ID: ARRAY[0..3] OF Char; //'fmt ' mind the space!
  SubChunk1Size: longword;        // hex10
  AudioFormat: word;              // hex 01
  NumOfChannels: word;            //1 mono, 2 stereo
  SampleRate: longword;           // number of samples/sec
  ByteRate: longword;             //samplerate* num of channels*
                                  //bytes per (mono) sample
  BytesperSample: word;           //size of (mono) sample [bytes]
  BitsPerSample: word;            // BytesperSample *8
  SubChunk2ID: ARRAY[0..3] OF Char; //'data'
  SubChunk2Size: longword;        //number of data bytes
  END;

PROCEDURE Left_volume(value: Dword);
PROCEDURE Right_volume(value: Dword);
PROCEDURE PrepareHeader(VAL: Tformat);

IMPLEMENTATION

PROCEDURE Left_volume(value: Dword);
VAR Rvol, temp: Dword;
BEGIN
  waveoutgetvolume(WAVE_MAP1, temp);
  Rvol := hiword(temp);
  ASM
    shl Rvol, 16
  
```

Herman Bulle

Almost all modern computers have a sound card, either on the motherboard or on an expansion card. This card is used to convert digital data into analogue sound signals. You may wonder if such a card could be used as a function generator. This is certainly possible, although the limitations of sound cards have to be kept in mind. There are various programs available that do this, but that isn't as much fun and educational as writing your own Delphi program that generates user-defined signals.

In this article we'll build a software function generator that can output a sine wave, sawtooth or square wave via the sound card. As a bonus we'll add the facility for the generator to read a file from the hard disc and output it as a waveform.

The bandwidth of a typical soundcard is about 20 kHz. When you use such a card to generate waveforms, you can create sine waves with frequencies up to 20 kHz. For other waveforms you need at least the third harmonic, which limits the maximum frequency to about 7 kHz. You therefore shouldn't expect to use this method to make a function generator with a 1 MHz output.

As with most projects that have some complexity, this project can be divided into a number of logically connected blocks (flowchart).

In the first block one complete cycle of the waveform is created. This data then has to be converted into a suitable format for the soundcard. And finally we have to program the soundcard to repeatedly output this signal until the stop button is pressed.

Creating the waveform

For the simpler waveforms we have declared a single **TDataArray** array, which can be accessed throughout the program (global). The number of samples stored in the array is defined by the global constant **NumOfSamples**. For a sine wave we store one period of the signal in the array. In the function for the sine wave we've declared the array **SineAr**, which is of type **TDataArray**. This is then filled with the values for a single period of the sine wave. Since the data sent to the sound card are 2-byte integers, the values have to be rounded first.

To help with calculating the sine wave we've made use of a special function that exists in Delphi. The **sincos(phi, sin, cos)** function is much faster than the normal calculation for sines and cosines because it was written completely in assembler by Borland and we gladly take advantage of it.

```
FOR index := 0 TO NumOfSamples DO
  BEGIN
    Sincos(2 * pi * index / NumOfSamples, sin, cos);
    SineAr[index] := round(amplitude * sin);
  END;
```

We use a similar method to create the sawtooth and square wave signals:

```
FOR Index := 0 TO NumOfSamples - 1 DO
  SawToothAr[Index] := round(2 * Index *
  Amplitude/NumOfSamples) - amplitude

FOR Index := 0 TO NumOfSamples DIV 2 - 1 DO
  BEGIN
    SquareAr[Index] := amplitude;
    SquareAr[NumOfSamples - 1 - index] := -amplitude;
  END;
```

We have to take a slightly different approach when we want to use an externally created waveform. It is fairly simple to create a waveform using Excel. The Excel example *Signals.xls* (part of the project software for this article, found on the Elektor Electronics website under 040240-4-11) shows how to make a waveform consisting of 100 cycles at maximum amplitude, followed by a 100 cycles of 10% amplitude. This type of signal is commonly used to test the transient response of amplifiers. The waveform itself is exported as a CSV (Comma Separated Variables) file.

This file is read into a variable of type **TStringlist**. This is a block of memory that doesn't have its size defined at the start. This type of variable is called a dynamic variable. We'll see more of these variables at a later stage. Since the declaration only defines the type of the variable we must also establish the start address, which will be used to access the variable. This is done with the **Create** method. In this case we've declared the variable **DataIn** to be of type **TStringlist**:

```
Var DataIn: TStringlist;
```

In the body of the procedure the following line is added:

```
DataIn := TStringlist.Create;
```

This variable has now been initialised for use. The name already suggests that this type is meant to store strings. Since the data from Excel are stored in a .CSV file (readable text), we can put these details into a variable for further processing. Those clever boys at Borland have created a method (procedure) to easily read this type of file from the hard disc:

```
DataIn.LoadFromFile('signals.csv')
```

To make the program more flexible we've added a file dialogue from Windows (File, Open), which is used to choose the file. This component can be found on the Dialogs tab and is called *OpenDialog*. It is used here as follows:

```
IF OpenDialog1.Execute THEN
  DataIn.LoadFromFile(OpenDialog1.FileName)
ELSE
  BEGIN
    //generate an error message
    MessageDlg('File is not correct', mtError,
    [mbOK], 0);
    //exit the procedure
    exit;
  END;
```

At this stage we have created or loaded the signal data. We now have to convert this data into a format that is compatible with the sound card.

Preparing data for the sound card

Next, we are going to fool the sound card into thinking that it's playing a sound fragment.

One file-type that is suitable for playback on a sound card has the file extension 'WAV'. A WAV file consists of a header of 44 bytes, which contains information about the file, followed by the actual data. This data is made up of the values of the signal, sampled at regular intervals. If you take enough samples at small intervals you can reproduce the signal fairly well. The sampling process is based on an extensive theory, which we won't cover here. A simple example of sampling is shown in **Figure 1**.

A WAV file is part of a broader range of file types, which also includes video files (such as AVI). We only concern ourselves with the simplest type here. This means that the header can be kept much simpler than the documentation suggests. A result of this is that some items appear twice in the header.

All information within the header is combined in a record. This is a block of data that contains a number of items of data (fields), which may be different types, such as strings, arrays, integers etc. When a record is speci-

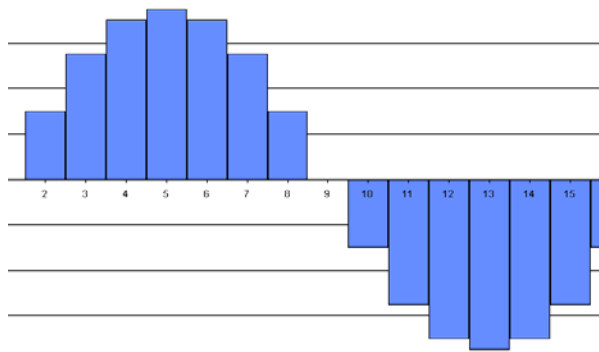


Figure 1. Division of a sampled signal into 16 steps.

fied we fix the offsets from the start of the record for all variables, as well as their size. If we make a mistake the compiler will warn of a type conflict, which avoids errors creeping into the program.

To improve the access times in complex data structures (such as records) the fields are often rounded upwards to the nearest 2, 4, or 8 bytes. The **ChunkID** variable, for example, is 3 bytes long and would be rounded up to 4 bytes. All following variables are then moved up by one byte, with the result that the sound card won't be able to make sense of the record. We can avoid this by declaring the record as a **PACKED** type.

The names of the fields have been kept as close as possible to those used in the specification for WAV files.

Type

```
TFormat = PACKED RECORD
  ChunkID: ARRAY[0..3] OF Char;
  ChunkSize: longword;
  Format: ARRAY[0..3] OF Char
  SubChunk1ID: ARRAY[0..3] OF Char;
  SubChunk1Size: longword;
  AudioFormat: word;
  NumOfChannels: word;
  SampleRate: longword;
  ByteRate: longword;
  BytesperSample: word;
  BitsPerSample: word;
  SubChunk2ID: ARRAY[0..3] OF Char;
  SubChunk2Size: longword;
END;
```

A lot of information within the header never changes. The procedure **PrepareHeader** is called to set these values in the headers. The values shown below are then assigned to the fields in the header:

```
WITH header DO
BEGIN
  ChunkID := 'RIFF';
  Format := 'WAVE';
  SubChunk1ID := 'fmt ';
  SubChunk1Size := $10;
  AudioFormat := $01;
  SubChunk2ID := 'data';
  NumOfChannels := 1;
  BytesperSample := 2;
  BitsPerSample := 16;
END;
```

As we're limiting ourselves to a mono signal for the time being, the value of **NumOfChannels** is 1 (stereo=2).

The sample size is also fixed at 16 bits (smallinteger). The **SampleRate** is the frequency at which the samples are output. For 'normal' signals a single cycle is output. The sample rate is therefore equal to the frequency of the signal multiplied by the number of samples in one cycle. The **ByteRate** is the number of bytes that is read per second and is therefore equal to $\text{SampleRate} \times \text{NumOfChannels} \times \text{size of one (mono) sample}$. In our case this is $2 \times \text{SampleRate}$.

The remaining fields hold information about the size of the data block:

Chunksize := data size (in bytes) +36

SubChunk2Size := size of the data block

We're now at the stage where we have all the information required to create the WAV file. We just have to combine the header and data and send them to the sound card.

Delphi offers a very flexible way to reserve an area of memory and fill it with data. This uses parts from the **TStream** 'family'. This family has several members of which we'll use the memorystream one (**TMemoryStream**). TMemoryStream has inherited all properties from Tstream and has added a few of its own.

Since streams are dynamic variables, only its type will be defined when the program starts and the actual variable still has to be created. As we can only send one WAV file to the sound card we only need a single variable of this type. The variable is created at the start of the program in the OnCreate method:

```
Wavestream:=TMemoryStream.Create
```

Just as with the TStringlist, the size of the variable is not defined here. It is this property that makes a stream very flexible.

Before the header is copied to the Wavestream it is first cleared:

```
WITH WaveStream DO
BEGIN
  Clear;
  WriteBuffer(Waveheader, sizeof(Waveheader));
  WriteBuffer(ar, sizeof(ar));
END;
```

Next, the **WriteBuffer** method is used to copy the header:

```
WriteBuffer(Waveheader, sizeof(Waveheader));
```

The first parameter is the variable that is written to the stream and the second parameter tells how many bytes have to be written. The data has also been prepared and is waiting in an array. This is written to the stream in the same way. The stream itself keeps track with a pointer where the end of the stream is and it starts writing from that position. In this way the data follows on neatly after the header.

```
WriteBuffer(ar, sizeof(ar));
```

We can read or change the value of the above-mentioned pointer using the **Position** property.

All the information is now ready to be sent to the sound card.

Because the creation of the WAV file made use of three pre-defined functions, we have combined PrepareHeader

and WriteBuffer in one procedure:

```
Data2Wav(Sender: TObject; VAR ar: TDataArray)
```

It wouldn't be very useful if we only displayed the sample frequency. We'd rather set and show the signal frequency. Because we already know how many data samples there are, the relationship between the signal frequency and the sample rate is straightforward:

```
WaveHeader.SampleRate = Freq * NumOfSamples;
```

With the help of a trackbar we can set the signal frequency and use that to calculate the SampleRate and show it on the display.

After a wave function is started the position of the **FreqTrackBar** is read and the SampleRate is set with the calculated value.

Once this is done we can use the FreqTrackBar to adjust the signal frequency. The position of the Trackbar is read and the SampleRate is calculated. We then turn the sound off for a moment (more on this later). The pointer in the stream is set to 0, at the start of the header and WriteBuffer is then used to copy the modified header over the old one. The sound is then turned on again. After every change in frequency the display is of course updated.

These actions have to be performed every time that the slider of the FreqTrackBar is moved. It therefore makes sense to start these actions via the **OnChange** event. Select the trackbar and double-click on the OnChange event under the tab for Events in the Object Inspector. We then add the following commands to the OnChange procedure:

```
PROCEDURE TForm1.FreqTrackBarChange(Sender: TObject);
BEGIN
  WITH FreqTrackBar DO
  BEGIN
    Waveheader.SampleRate := FreqTrackBar.Position *
    NumOfSamples;
    //turn sound off
    PlaySound(NIL, 0, 0);
    WaveStream.Position := 0;
    WaveStream.WriteBuffer(Waveheader, sizeof(Waveheader));
    //with '%5d' numbers are displayed with
    a field-width of 5 characters
    FreqLabel.Caption := Format('%5d', [FreqTrackBar.Position]) + ' Hz';
    //turn sound on again
    PlaySound(MStream.Memory, 0, SND_MEMORY OR
    SND_LOOP OR SND_ASYNC);
  END;
END;
```

The imported waveform still has to be converted into a WAV file. This was already read into the **DataIn** StringList, which is a stream optimised for strings. The StringList has a **Count** property, which returns the number of strings read. This therefore tells us the number of samples. Because the waveform was created externally we've assumed that the sample rate is 50 kHz is. The sample rate has to fit in 2 bytes and therefore has a maximum value of 64 kHz. 50 kHz is close to the maximum and is easier to use in calculations. We have already made sure that the maximum amplitude is within limits when we created the data file in Excel, so that is also take care of. We can use this information to construct the wave header.

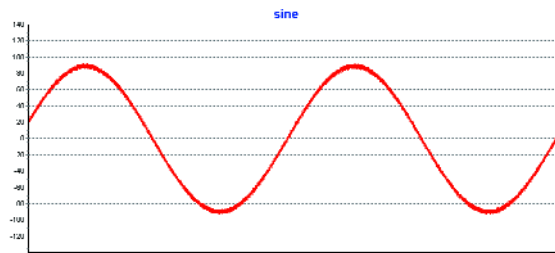


Figure 2. A sinewave output from the sound card (2 kHz).

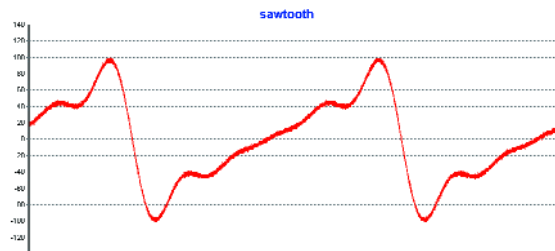


Figure 3. This is what the sawtooth looks like (also 2 kHz).

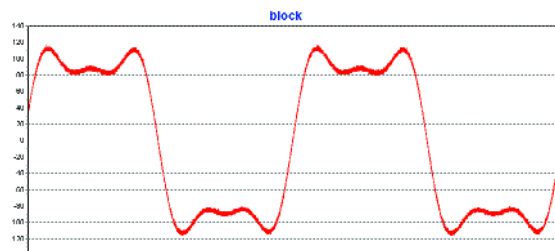


Figure 4. And finally a squarewave output (2 kHz as well).

der. We then read the StringList sample by sample, convert it to a *smallinteger* and add it to the WaveStream:

```
FOR index := 0 TO SampleCount - 1 DO
  BEGIN
    sample := StrToInt(DataIn.Strings[index]);
    WaveStream.WriteBuffer(sample, sizeof(sample));
  END;
```

Once this is completed, the sound card can be loaded with the WaveStream and the playback can start.

Driving the sound card

The unit *MMSystem* has several procedures that we need to control the sound card. The most important procedure is the one to play back a file:

```
PlaySound(<param1>, <param2>, <param3>)
```

In the simplest case param1 is the name of a WAV file, for example: *c:\winnt\media\chimes.wav*.

It is also possible to pass a memory address (pointer) to a WAV file stored in memory and that is what we use here. A *MemoryStream* has a **memory** property, which is of type **pointer**. This was the main reason for

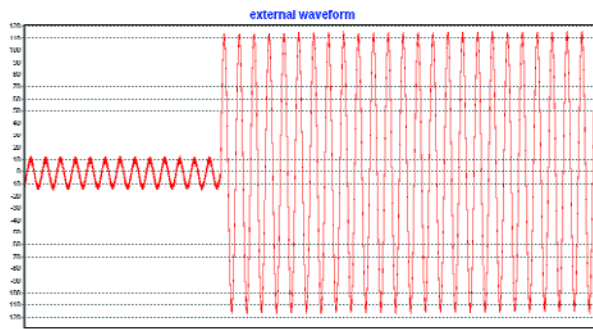


Figure 5. Part of the waveform created in Excel. The complete cycle consists of a 1-Hz signal, 500 ms at 100% amplitude, followed by 500 ms at 10% amplitude. The sampling frequency is 50 kHz.

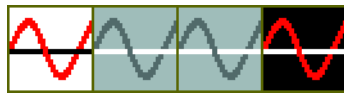


Figure 6. The construction of a bitmap for a SpeedButton.

choosing this type of stream. The first parameter then looks as follows:

```
PlaySound(WaveStream.memory, <param2>, <param3>)
```

Param2 is 0, unless sound is played back from a resource, but we won't go into this here. By choosing a suitable value for param3 we tell the sound card that param1 isn't a file, but a pointer to a memory address. Param3 is made up of a combination of 3 constants that are defined in the unit *MMSystem*. *SND_MEMORY* indicates that param1 is a memory address, *SND_LOOP* makes the file play indefinitely and *SND_ASYNC* tells that the playback has to happen autonomously. The last value is required in combination with *SND_LOOP*. Whenever we want to send the data stream to the sound card we use the procedure:

```
PlaySound(WaveStream.Memory, 0, SND_MEMORY OR  
SND_LOOP OR SND_ASYNC);
```

To turn the sound off we also use *PlaySound*, but with different parameters:

```
PlaySound(NIL, 0, 0)
```

Since the first parameter is a pointer, we can't just put a zero in it. Instead, we have to use a special type of empty pointer, called *NIL*.

The only thing left to do is to set the amplitude of the signal, which we do via a Windows function from the *Wave Amplitude Control*. This requires a 4-byte variable, which holds the signal level. The two most significant bytes determine the signal level of the left channel and the other two bytes are for the right channel.

Two functions have been defined to adjust both channels independently. We first use **waveoutgetvolume** to read the current volume level. We then replace the highest or lowest two bytes and write back the variable using **waveoutsetvolume**.

The volume setting usually follows a logarithmic rule. This hasn't been taken into account when we show the ampli-

tude as a percentage of the maximum signal level.

Layout

We have taken particular care with the layout. For the first time we've used buttons that contain images. We have chosen to use so-called speedbuttons because they have several useful properties. Depending on the state of a button it can show a different image. It is also possible to unselect all other buttons when a single button is clicked. If you want to unselect a group of buttons they have to know about each other's existence. They know this via the property **groupindex**. When a button is clicked, all other buttons with the same groupindex are unselected (groupindex > 0). This is an ideal solution when we want to select a single button, for example when we choose a waveform.

Images can be added via the *Glyph* property. A glyph (a 'raised' character or symbol) is a bitmap that contains a maximum of four images, which are all of the same size and are next to each other in a horizontal row. The *SpeedButton* shows one of these images, depending on the state of the button (see **Figure 5**).

- Image 1 Button is unselected
- Image 2 Button disabled
- Image 3 Button is clicked. If groupindex=0 image 1 appears again afterwards.
- Image 4 Button stays down.

We also have to tell the speedbutton that there are 4 images in the glyph via the property **NumGlyphs=4**.

Conclusion and results

Even though the externally generated WAV file has a limited frequency range (max. sample frequency of 64 kHz), this is a very useful function generator. Because we can use externally generated waveform files, the number of possible waveforms is infinite, which makes this fairly unique amongst signal generators.

We could expand the generator by making use of both (stereo) channels. The left and right channels could then be programmed separately. As an example, when working with complex signals one channel could be programmed with the signal and the other channel could be used for a trigger signal with a programmable delay.

When we used a spectrum analyser (Agilent 4395A) to analyse a sine wave output, we obtained surprisingly good results:

- S/N ration: > 95 dB
 - 2nd harmonic: < -65 dB
 - 3rd harmonic: < -80 dB
- Measured with a 1 kHz signal at 0 dBm.

These values were measured on an arbitrary computer that had its sound card on the motherboard. Other cards may give very different results.

A number of sound cards were found to produce significant distortion at maximum output levels. It is important to keep an eye out for this if distortion is of concern. If necessary, the maximum amplitude should be set to a lower value.

The waveforms shown in this article have been captured by a digital oscilloscope (leCroy WaveRunner LT344L). All program files for this article can be found at www.elektor-electronics.co.uk (April 2005, file no. **040240-4-11**).

(040240-4)

Flash PICs with USB 2.0 Connectivity



Microchip recently announced the first Flash PIC® microcontrollers with Full-Speed USB 2.0 connectivity and 48 MHz operation, for 12 megabits-per-second (Mbps) data-transfer rates. Combined with a wide variety of peripherals and nanoWatt Technology power management, these features provide a complete solution for designers working with USB in industrial, medical and many other embedded applications.

Most USB-capable microcontrollers are optimised for personal computing (PC) and consumer applications, rather than embedded designs. Microchip's new USB PIC microcontroller family offers the benefits of Full-Speed USB to a broader range of applications that may operate in harsh environments and connect to personal computers less often. Key features of the new devices include:

2 kBytes of RAM, 1 kByte of which can be a dedicated USB buffer;

256 bytes of EEPROM data memory;

EUSART module supporting RS232, RS485 and LIN serial interfaces;

Master Synchronous Serial Port (MSSP) supporting I²C™ and SPI™ communication;

10-bit Analog-to-Digital Converter with up to 12 channels, Two

Analog Comparators
Capture/Compare/PWM (CCP)
module with 16-bit capture
and resolution

Enhanced
Capture/Compare/PWM
(ECCP) with dead-time control
and fault-protection inputs
Four Timers (3 x 16-bit, 1 x 8-bit)
Programmable Brownout Reset
and Low Voltage Detect circuits
Enhanced In-Circuit Debugging
capabilities with up to three
hardware breakpoints.

Microchip offers an extensive set of libraries for the most common application classes, including the Human Interface Device (HID), Communication Device Class (CDC) and custom drivers. Microchip is also publishing an application note illustrating how to replace a serial port with a Full-Speed USB link in a few easy steps.

The four new USB PIC microcontrollers are planned for general sampling and volume production in December, in several package options. For additional information visit the Microchip website at www.microchip.com/usb

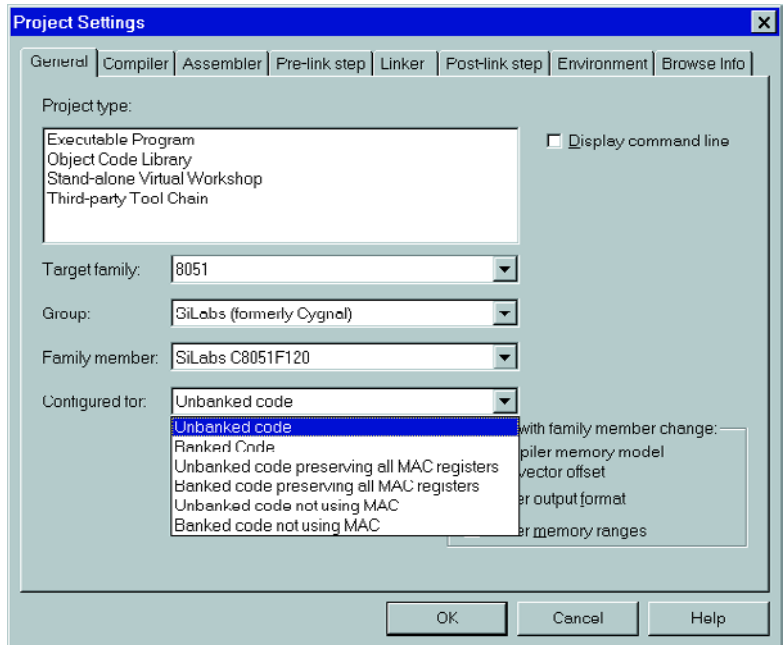
**Microchip Ltd., Microchip House, 505
Eksdale Road, Winnersh Triangle,
Wokingham RG41 5TU. Tel. (+44)
(0)118 921 5869. Fax (+44) (0)118
921 5820. www.microchip.com.**

(057017-1)

Support for Silicon Labs C8051F12X and C8051F13x MCUs

Crossware has added full support to its 8051 Development Suite for the Silicon Laboratories (www.silabs.com) C8051F12X and C8051F13X mixed signal microcontroller units (MCUs). The Silicon Laboratories' chips feature advanced pipelining and caching and a phase locked loop which together allow the core to run at up to 100 MIPS. A multiply and accumulate (MAC) engine can also perform a 16-bit signed multiplication in two cycles. The Crossware enhancements allow developers to rapidly exploit the advanced features of these chips by providing a thorough combination of wizards, simulation, debugging, compiler extensions and pre-configuration. Code Creation Wizards are provided for all on-chip peripherals. They allow the phase locked loop to be rapidly configured to exploit the 100 MIPS maximum performance and with the compiler and libraries using multiplication code that exploits the MAC engine, developers can instantly take full advantage of the speed offered by these chips. The Crossware environment is

fully pre-configured to take advantage of the MCU's 128 kBytes of in-system programmable banked flash and when simulating and debugging the banking process is virtually transparent to the developer. In addition, the flash memory Code Creation Wizard will generate all the code necessary to erase and rewrite flash memory during program execution. Crossware's source level debugger drives the Silicon Laboratories' JTAG emulation cartridge directly. This allows it to make use of the full range of on-chip debug facilities including the hardware data breakpoints and hardware stack overflow and underflow trap. Crossware's 8051 development suite includes a full featured ANSI C compiler, a relocatable cross-assembler, an advanced overlay linker, a state of the art



source level simulator that can be extended to simulate a complete target system and a debug monitor which supports source level debugging on the target system itself. All of these integrate into Crossware's Embedded Development Studio development environment to bring together the only single vendor

tool set that can be used to develop and fully debug 8051 programmes both with and without hardware. **Crossware Products, Old Post House, Silver Street, Litlington, Royston SG8 0QE. Tel. (+44) (0)1763 853330. Website: www.crossware.co.uk**

(047199-9)

Sweeping Changes in your Application

Lascar's new SP5 1710-BL has a large, clear LCD that shows a 17-position 'analogue' style display. Essentially a voltmeter with a full scale deflection of 1 V, this module can be used in many places where a quick indication of level, rather than an exact reading may be required. The needle on the SP5 1710-BL can be centre or left justified to suit a variety of applications and the LCD features a permanent colour scale and markers to help gauge the status of the measurement. The ultra low profile, splash proof housing also contains a

vivid green LED backlight that allows the module to be seen in all environments.

The P5 1710-BL has been designed for ease of use in small and high volumes. It is available directly from Lascar at a price of £ 21.95 plus VAT.

Lascar Electronics Limited, Module Hose, Whiteparish, Salisbury, Wiltshire SP5 2SJ. Tel. (+44) (0)1794 884567, Fax (+44) (0)1794 884616. Website: www.lascarelectronics.com

(047199-7)



BRACHION D

Does alcohol cause violent behaviour?

Peter van Herpen

Researchers at Warrick University recently discovered new particles and dubbed them 'brachions' (from Latin brachius which means arm). The term also has a strong association with Brachiosaurus, one of the most ferocious creatures that ever trod the earth, so clearly the newly discovered particle is related to violence in some way.

The experts were able to prove that a high concentration of brachions in blood increases the risk of a person turning violent. One of the most intriguing questions is of course how brachions end up in blood. After a good deal of research, mostly in pubs, it turned out that brachions are mostly (but not exclusively) present in alcoholic drinks

To everyone's surprise, the relation between brachion concentration and the alcohol percentage is far from linear. Also, the effect of brachions is dependent on personal factors like age, body mass index (BMI) and sex. The correlation between these factors is illustrated in **graph 1** (for females with BMI between 20 and 30). Dr Poof and her pub-crawling co-workers at the University were able to design an interesting circuit for use in

clinics, football arenas, boxing halls and prisons that measures the brachion concentration in any liquid, showing the (possible) effects on a display.

The brachion concentration is measured in a non-invasive manner, using two sensors. One is an infrared sensor measuring the absorption of infrared light, the other, a conduction sensor. Based on a complex formula, these two factors establish the brachion particle count in blood. The formula is implemented in the system software.

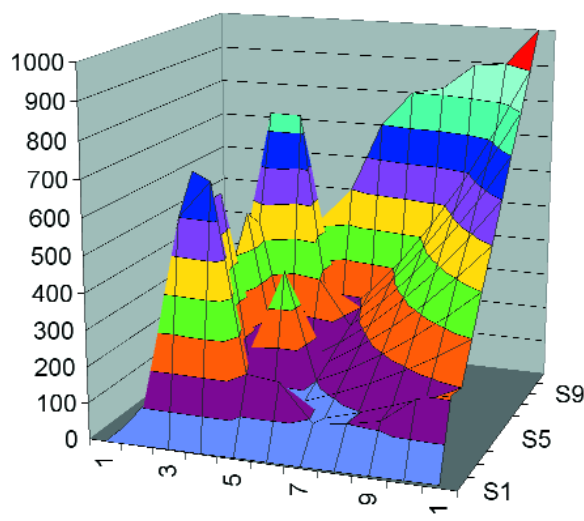
Circuit

As a spin-off of the scientific project, a circuit was designed specially for Elektor readers to enable them to experiment. The heart of the circuit is a Siemens 80C535 processor ticking at 12 MHz. The sensors — here, a common reflective light port and a simple conduction sensor — are connected direct to it. For the rest, the schematic does not amount to much. The display is connected directly to port 0 via pull-up resistors (because of the open-drain outputs) and the rotary/pushbutton is tied to some inputs on port P1. The software may be blown into an EPROM. Because of the experimental nature of the circuit, pre-programmed EPROMs are not available.

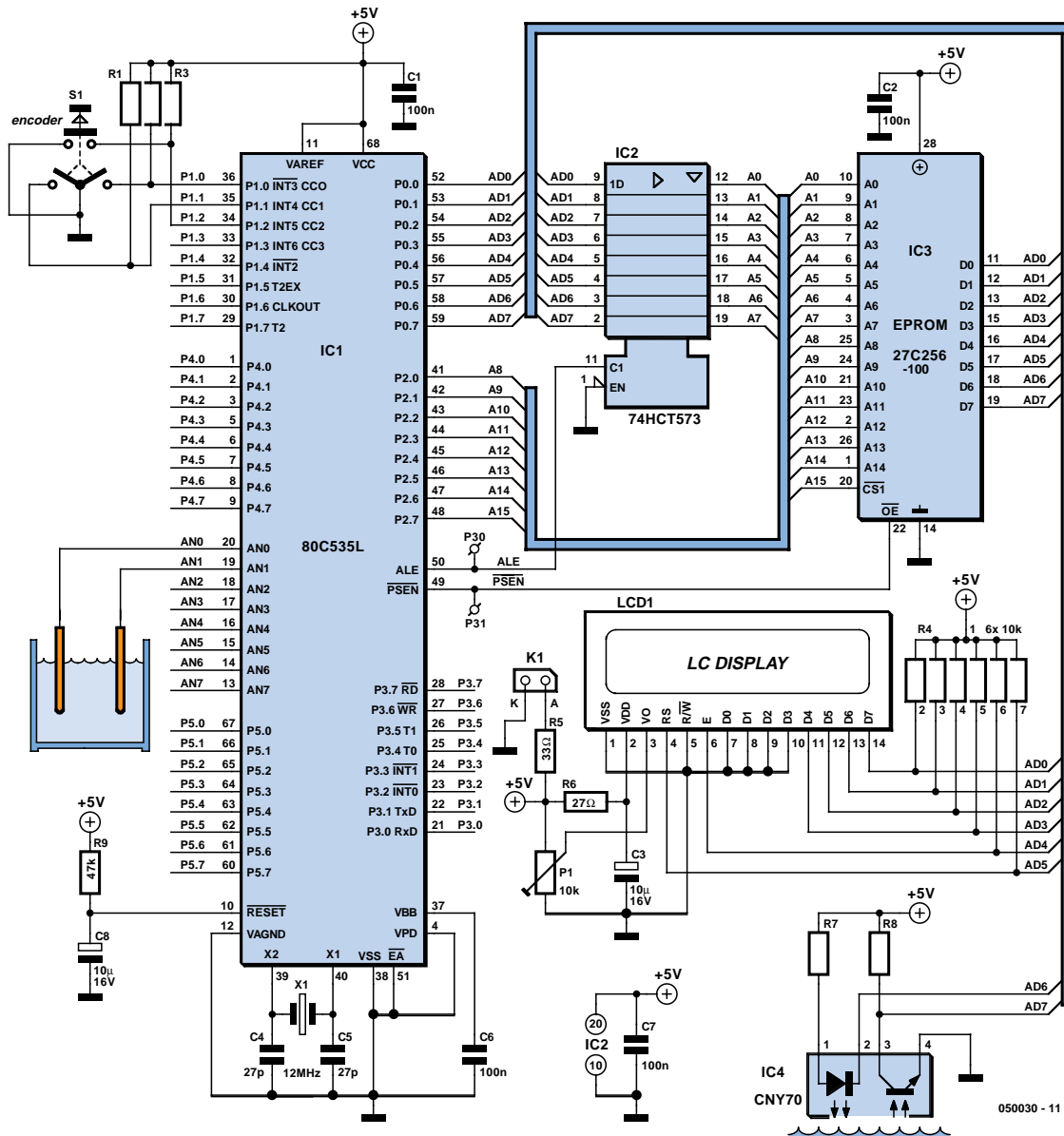
User instructions

Operation

First use the rotary/pushbutton to select the sex (M/F), then the person's length, weight and age. The software will use these data to compute the BMI and select the right parameters for the graph. It is important not to tell your test person(s) that the variable 'age' is actually limited to 35 when 'Female' is selected because apparently women can not be older than that, nor that 'weight' is actually increased by 15% because women, as is com-



ELECTOR



monly known, will always lie about this. Next, the probes are submersed in the liquid to be examined and the actual measurement may begin. It will take about 10 seconds before the calculations begin — the process is easily followed on the display. The result is expressed in brachion count per litre. A typical value for (French) red wines (not older than 1995) is 4.37×10^{-12} brachions / litre. Next, if you press the button again, you will see the effect on the test person if he/she drinks 0.1 litre of the liquid. By repeated pressing of the button, the amount is increased in steps of 0.1 litre (five key presses approximate one pint). For women, the order of the steps is from 'none' to 'feel like' and 'feel nothing'; and for men, from 'none' to 'looks bigger than it is', 'well and truly spent', 'violent', right up to 'com-

pletely plastered'. Many more intermediate levels are available in the software. We probably do not need to tell you that field-trialling the software was a huge undertaking mostly done outside our office hours and we wish to express our gratitude to Jameson and Guinness for their kind support. Readers interested in obtaining the software may send an email to editor@elektor-electronics.co.uk, subject: 'brachion detector April 2005'.

(050030-1)

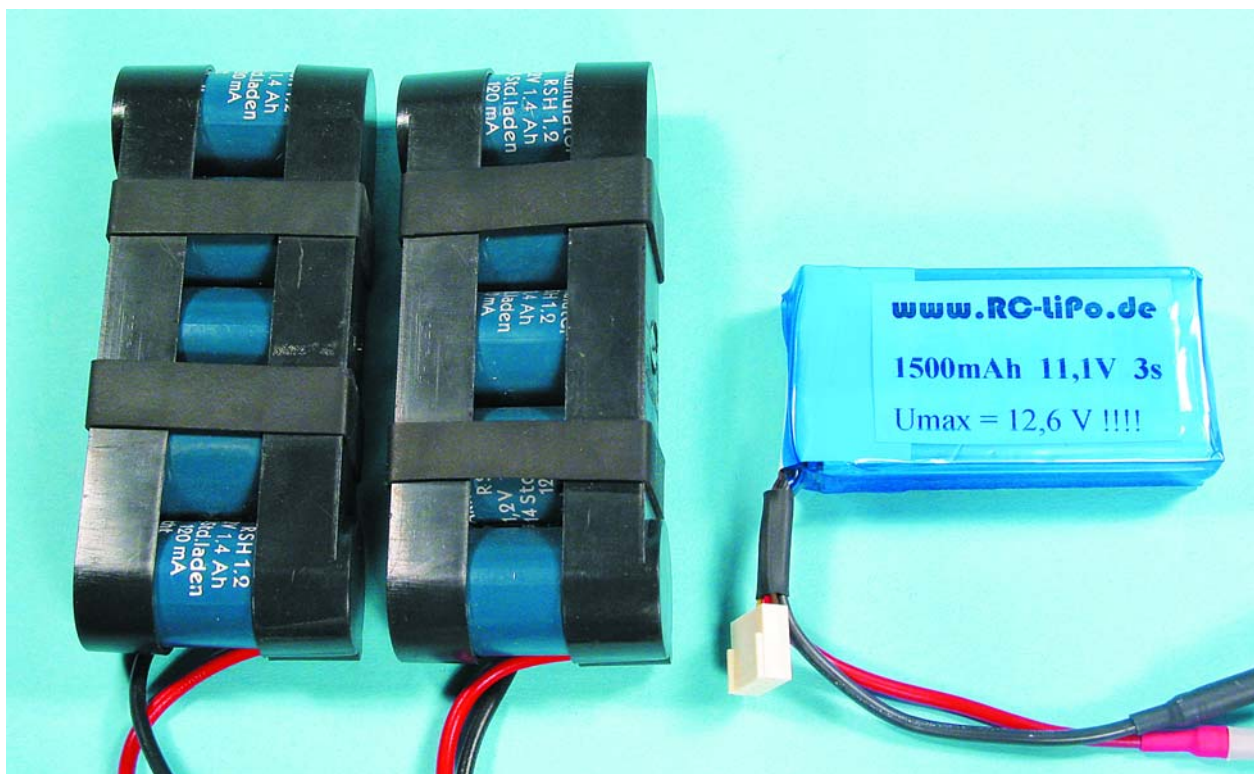
Literature:

Brachions, Elementary Particles and their Effects, by Dr April S. Poof, Warrick University publication, 2004.

SUPER LITHIUM

Everything you've ever wanted to know

Ulrich Passern



Modern mobile electronic gadgets demand ever more portable and powerful energy storage devices. Research by battery manufacturers all over the world moves on apace; new developments supplant old technologies, and are themselves almost instantly superseded by even newer technologies. The field is in a state of constant flux with new surprises every day.

There are three battery technologies that currently dominate the market. The longest established and most technically outmoded is the lead-acid battery. However, its almost legendary robustness and low price guarantees its place powering the starter motor in vehicles and for

energy storage for emergency supplies. The nickel family consists of two well-known products: the nickel-cadmium (NiCd) battery and the nickel-metal hydride (NiMH) battery. This will soon change, as the cadmium in NiCd batteries is harmful to the environment

BATTERIES

about lithium batteries...

and, under an EU regulation, the batteries will disappear from the market in 2008. The industry is prepared for this and has driven the development of NiMH cells to the point where they are a match for NiCd cells in almost every respect. NiCd and NiMH cells have a built-in chemical 'charge reserve' and hence are relatively insensitive to overcharging or to deep discharge.

For some time it has not been possible to meet the energy requirements of modern mobile phones, laptops and PDAs using nickel-based batteries and, unnoticed by many, a third battery family has gained considerable importance in recent years. These are lithium-based batteries: lithium-ion (Li-ion) and lithium polymer (Li-polymer). Lithium technology has answered many researchers' prayers, since, with a standard electrode potential of 3.04 V, it sits at the top of the electrochemical series. These new energy storage devices are ideally suited for modern portable electronic equipment, since they are light, small, and have an extremely high energy density. It would be wrong to believe that we have reached the end of the road: advanced battery development, especially for laptops, is leading towards even more powerful energy sources, and in a few years lithium battery technology will be overtaken. The next generation in mobile power sources is already appearing on the horizon: the fuel cell, on which more later.

A sensitive soul

For a long time it was thought that engineers would not be able to control the complex processes in a lithium cell. However, enormous amounts of research effort have finally paid off and now lithium batteries have replaced nickel batteries in almost all key applications. However, as observant *Elektor Electronics* readers will know, it has until recently been practically impossible to buy individual Li-ion or Li-polymer cells. The reason is that it has not been possible to equip lithium cells with the 'charge reserve' we mentioned earlier, and so they are extremely prone to damage. If a Li-ion or Li-polymer cell is even slightly overcharged, it starts to produce gases and can rupture (**Figure 1**). The gases produced are also extremely poisonous. If a lithium cell is heavily overcharged, it can catch fire or, in extreme cases, explode. Since lithium is highly reactive, a lithium fire is very hard to put out. Water behaves like petrol, aggravating the fire, and must on no account be used on a lithium fire. It is better to use a fire blanket, a suitable extinguisher, or even sand.

Electronic protection

The industry has replaced the 'charge reserve' missing from lithium cells with intelligent protection circuitry.



Figure 1. When a Li-polymer battery has ruptured, it is useless. In extreme cases, the battery can explode.



Figure 2. The integrated monitoring circuits in industrial lithium batteries are always microprocessor-controlled.

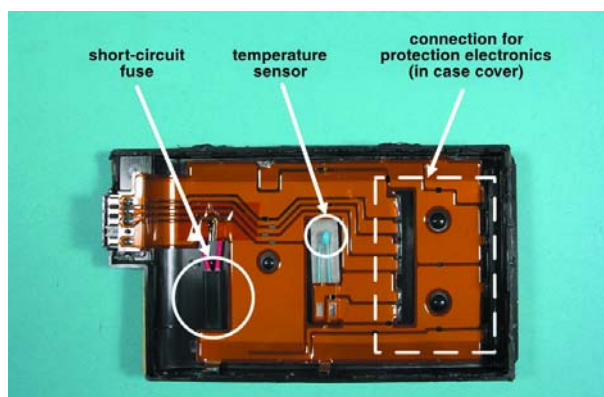


Figure 3. Monitoring using multiple sensors protects the battery from overcharging.

These are built into the battery pack (see **Figure 2**) and use a number of sensors to protect the cells against overcharging or short-circuit (**Figure 3**). In tests by consumer organisations these protected battery packs have been found to be safe. It would be irresponsible to dispense with the safety electronics: you may remember stories in

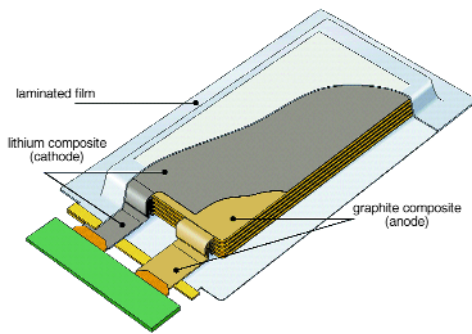


Figure 4. Schematic construction of a Li-polymer battery (source: Varta).



Figure 5. Li-ion battery for laptops in 3S2P configuration (10.8 V terminal voltage)

5S4P 6000mAh

↳ cluster of 4 parallel cells

↳ 5 clusters in series

Number before "S" to be set on charger (here: 5)

Figure 6. Labelling scheme for Li-ion and Li-polymer batteries.

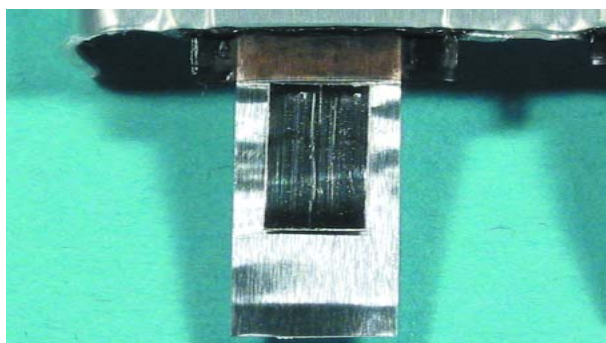


Figure 7. A piece of tin is usually attached to the aluminium terminal of a Li-polymer battery.



Figure 8. A wide piece of tape serves as a strain relief for the connection wires.

the press about exploding Nokia mobile phones. Here cheap third-party battery packs were to blame, which lacked short-circuit protection.

Lithium-polymer on top

Within the lithium family two members have been competing for superiority over the last two years. The market share of Li-ion cells has been falling steadily as Li-polymer cells gain ground. The background to this is simple: Li-ion cells contain a liquid electrolyte and must therefore be protected by a rather heavy steel jacket. Li-polymer cells, on the other hand, use a polymer film to fix the electrolyte and can be packaged in lightweight aluminium foil (Figure 4). Li-polymer cells can therefore offer more energy in a smaller volume and are considerably lighter than Li-ion batteries. Nevertheless, the thin aluminium casing is easily damaged, and so Li-polymer cells must be handled and stored carefully.

The flat construction of the Li-polymer cell, which was described in the May 2000 issue of *Elektronik*, makes it easy to use in many applications. A wide range of Li-polymer cells with capacities from 250 mAh to 6000 mAh is available. Prices, however, are fairly high, ranging from £ 4 for a 350 mAh cell to £ 30 for a 2100 mAh cell, depending on the quality, capacity and performance under load.

High demand, especially from modellers, has meant that over the last year many manufacturers and suppliers have started offering individual Li-polymer cells or ready-assembled packs (generally without protection electronics). In this connection a **warning** is in order: individual Li-polymer cells must always be stored out of the reach of children, since they look just like outsize strips of chewing gum asking to be unwrapped or bitten into.

Pack assembly

Li-ion and Li-polymer cells can be connected in series or in parallel. Figure 5 shows an industrial Li-ion pack for a laptop, where cells are connected in parallel in pairs, three such pairs being wired in series. Figure 6 shows the standard marking scheme that has been developed for these battery packs.

If you wish to build your own Li-polymer battery pack, there are a couple of problems to overcome. A particular inconvenience is that the positive terminal is made of aluminium and cannot be soldered using normal solder. The manufacturer therefore attaches a small solderable metal strip to the terminal (Figure 7), which, in accordance with Murphy's law, is usually on the wrong side when it comes to assembling the pack. A better alternative is to use a special aluminium solder, which is unfortunately not cheap. The thin terminal connections tend to break easily and a strain relief must be fitted (Figure 8). To simplify the job of assembling packs, many manufacturers offer special solderable circuit boards as an accessory, or provide the cells with the connections already brought out to a small circuit board.

Pure power

Since Li-polymer cells are available in a range of different sizes and can be wired together into packs in various ways, some extremely high-power units have become available — at a price. The two Thunder Power Li-poly-

mer packs (see photograph at the beginning of this article) each have a capacity of 6 Ah, a nominal voltage of 18.5 V, and can handle a continuous 8 C (48 A) load! The two batteries together can deliver over 1600 W, despite weighing only 1.2 kg. The pair of batteries costs about five hundred pounds.

The best price-performance ratio is currently offered by the 850 mAh HDHE cells with a weight of 18 g produced by Kokam. These can be discharged at up to 10 C (i.e., up to 8.5 A) with peak loads up to 15 C. A hand-soldered battery pack constructed from these cells is shown in **Figure 9**. Here six cells are wired in parallel to form a block, and five such blocks are wired in series. The pack (5.1 Ah, 18.5 V, 30 cells) can deliver 51 A continuously with peaks up to 76 A, has a weight of 850 g and costs about £ 250 in total.

Short-circuit protection electronics

Lithium batteries must be monitored both during charging and during discharge to ensure they are not overloaded. Measurements show that the cells in a lithium battery behave differently from one another both when charging (**Figure 10**) and when discharging (**Figure 11**). Individual monitoring of the cells is therefore recommended. The whole battery can be monitored using a circuit along the lines of that shown in **Figure 12**: the six-cell battery shown requires just two extra wires to allow the cells to be monitored individually. At present there is no standard for these extra connections. Some manufacturers use multiple connectors (**Figure 13**), while others use just a single extra connector (**Figure 14**).

During charging a so-called 'Li-polymer balancer' circuit (**Figure 15**) is connected to the extra wires for individual cell monitoring. The balancers route current past the fully-charged cells and thereby protect the cells (or blocks of cells connected in parallel) from overcharging. This allows the battery as a whole to be charged further, by diverting the current to the other cells or blocks of cells. Li-polymer balancers do not by themselves prevent overcharging, but they do keep the cells in step with one another and thus increase the total energy available. Circuits to monitor individual cells in lithium battery packs are still under development. The circuits that are generally used interrupt the discharging of the battery as a whole when a voltage of 3 V per cell is reached. Li-polymer cells suffer permanent damage if discharged below 2.5 V, and so this threshold value allows for a certain amount of variation between the cells in the battery pack.

Charging techniques

Lithium batteries, like lead-acid batteries, are charged at constant voltage. The charger must simply provide a fixed voltage of 4.1 V per cell for Li-ion batteries with a lithium cobalt oxide cathode or 4.2 V per cell for Li-polymer, with a tolerance of 0.05 V per cell. The current is given by the formula

$$I = (U_{\text{charger}} - U_{\text{batt}}) / R_{\text{ibatt}}$$

and falls to zero when the battery is fully charged. Overcharging is in principle impossible, and the battery can be left connected to the charger for as long as desired, although for safety reasons lithium batteries should never be charged unattended. In order that the current is not



Figure 9. Hand-made Li-polymer battery using thirty 850 mAh cells in a 5S6P configuration. The 5.1 Ah, 18.5 V pack can deliver currents up to 10 C (51 A).

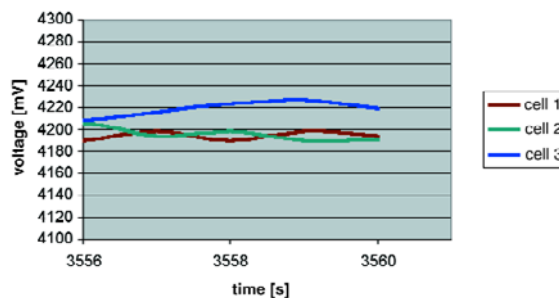


Figure 10. The individual cells in a Li-polymer battery can reach their upper threshold voltages.

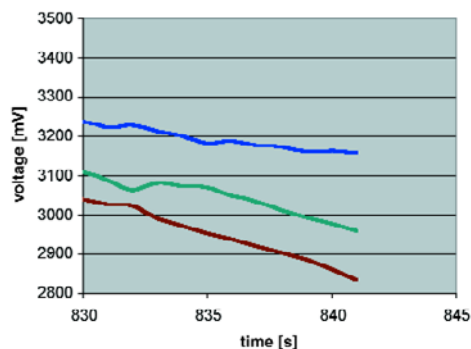


Figure 11. The cells also usually behave differently when discharging, and therefore must be monitored individually.

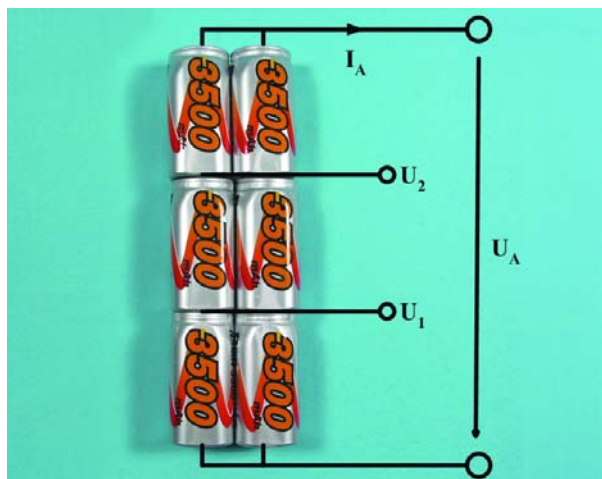


Figure 12. Lithium cells (Li-ion and Li-polymer) can be connected in series or in parallel. The figure shows the principle with NiMH cells.



Figure 13. Extra connections are brought out to allow the individual cells to be monitored.

About the author

Ulrich Passern was born 52 years to the day after the first powered flight by the Wright brothers. Inspired by his father (an airman) at an early age, he developed a passion for constructing and flying model aircraft. In professional life he found his second passion: electronics. After studying electrical engineering he started his career at a large computer company. After founding his own engineering consultancy for measurement technology, model building and profession came together in the development of a battery testing device, which was subsequently developed into a computer-controlled charger which was very successful both in industry and in the world of model building. Meanwhile, Ulrich is once more working in the field of computers and regularly carries out comparative testing of batteries and chargers for the German modellers' magazine FMT.



too high when charging begins (it should not exceed 1 C), cheap chargers are simply arranged to have a suitable internal resistance. More advanced chargers provide proper current limiting. It is thus very easy to build a lithium battery charger. **Figure 16** shows the LiPo402 charger from Kokam, which easily fits in the pocket. This charger is microprocessor controlled and can be adjusted to suit various Li-polymer battery types. The

price is around fifty pounds. You can of course construct your own lithium battery charger, as long as you are careful to observe the exact voltage and current threshold values.

Ageing

Lithium batteries generally exhibit a very low self-discharge and can be stored for a long time without requiring attention. The inevitable ageing is minimised if it is discharged to about 20 % of its normal charge and stored in a cool dry room (at about 10 degrees Celsius). Cycle life of modern lithium batteries is similar to that of other batteries: about 300 to 500 charge/discharge cycles should be achievable without difficulty if the battery is not abused. The 'memory' or 'lazy battery' effect found in nickel-based batteries does not occur in lithium batteries, and so they need not be discharged before charging.

The outlook

The power requirements of modern mobile devices are forever increasing, and all over the world research continues at a feverish pace on a completely new type of energy source: the miniature fuel cell.

In 2003 Japan identified the fuel cell as one of the six technologies most important to the future of the Japanese economy. Almost as soon as this was announced, three Japanese companies simultaneously presented their plans: NEC, Toshiba and Hitachi already have functioning fuel cells in their laboratories and are working on a schedule for mass production. Hitachi's PDA fuel cell should come onto the market in 2005. Mobile phone manufacturer Nokia is already testing Bluetooth headsets powered by a fuel cell. To prove their suitability for everyday use the devices are being used in-house by a hundred Nokia employees.

The long-term aim of the Japanese manufacturers is to replace all currently-used batteries with suitably-sized fuel-cell equivalents. It is more probable, however, that the various types will coexist, as over one hundred years of history of the lead-acid battery shows. However things turn out, mobile power supply devices have an exciting future.

(040168-1)

Figure 14. Li-polymer batteries from Graupner are fitted with a special connector to allow the individual cells to be monitored.

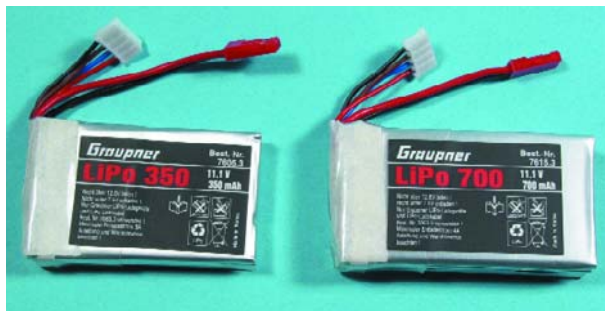


Figure 15. Li-polymer balancer circuits protect the battery from overcharging and keep the cell blocks in step with one another.

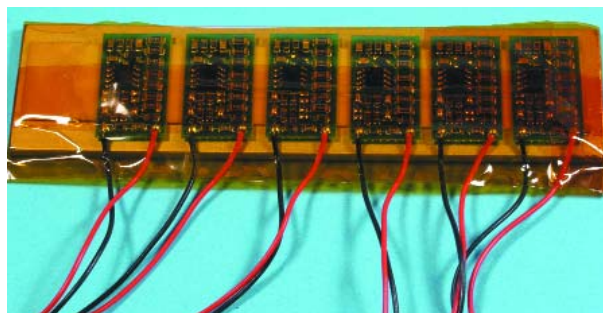


Figure 16. The LiPo402 by Kokam is a modern compact charger for Li-polymer batteries.



Comparison of energy capacities

Normally the capacity of a battery is specified in Amp-hours (Ah). Using this unit allows us to determine almost immediately how long a battery can continue to provide power under a given load. For example, to calculate roughly how long a standard 21 W bulb can be run from a fully-charged 44 Ah car battery:

$$I = P / U = 21 \text{ W} / 12 \text{ V} = 1.75 \text{ A}$$

and

$$44 \text{ Ah} / 1.75 \text{ A} = 25.1 \text{ h}$$

In order to compare batteries with different terminal voltages it is better to work with the energy capacity of a cell in Watt-seconds or Watt-hours (Wh), and the specific energy in Wh/kg. The often (mis)used term 'energy density' usually refers to the energy per unit volume, rather than per unit weight.

Web pointers

The following web sites have further information on lithium batteries, protection circuits and chargers.

<http://www.PowerBox-Systems.com>

<http://www.graupner.de>

<http://www.ikarus-modellbau.de>

Lithium-sulphur batteries

Another development is lithium-sulphur (Li-S) technology, which gives a cell voltage of 2.1 V. This should not be confused with the lithium-sulphur dioxide (LiSO₂) technology. The company Sion (www.sionpower.com) has developed a prototype specified at 350 Wh/kg. This is a major advance in comparison with current commercially-available technologies. The NiMH cells in the Toyota Prius hybrid car, for example, have a specific energy of just 46 Wh/kg. The relatively high specific energy is not the only benefit of this technology: Li-S batteries can deliver very high currents, perform well at extremes of temperature and in principle are immune to overcharging.

Of course, there are also disadvantages when compared to other systems: after about 350 to 400 charge cycles the capacity of even the most modern Li-S batteries falls to about 80% of its original value. For most applications a minimum of 350 charge cycles is specified, and 500 cycles is seen as a desirable target.

One of the reasons for the rapid loss in capacity is an effect whereby the composition of the liquid cathode slowly changes. A film which forms on the fixed electrode during discharge affects the characteristics of the battery. It is difficult to keep the cell in a stable, effective condition.

NiMH cells with a pressure switch

Nickel-based rechargeable cells are ideal for powering modern electronic equipment such as MP3 players, mobile communications devices or digital cameras, which require the cells to be able to deliver high currents for short periods of time.

The other side of the coin is the lengthy charging time, or, alternatively, the high cost of a charger that can charge the cells quickly without (even more quickly) destroying them. Such 'intelligent' chargers invariably employ a microcontroller and temperature sensors. Of course, microcontrollers with integrated sensors make the unit somewhat smaller and cheaper, but the next step, integrating the intelligence into the cell, has foundered for reasons of cost.

Rayovac (Varta) has recently developed the so-called I-C3 cell ('in-cell charge control'), which can be recharged without using an intelligent charger in 15 minutes, and, thanks to its optimised construction, also offers longer discharge times. There is no electronics in the cell; instead there is a small mechanical pressure switch. This is not only economical, but also technically elegant: the pressure inside the cell is a better measure of its state than its temperature or its voltage. When the pressure reaches a set value (in the range 1000 kPa to 3500 kPa), a sprung disc breaks the connection: the cell is fully charged.

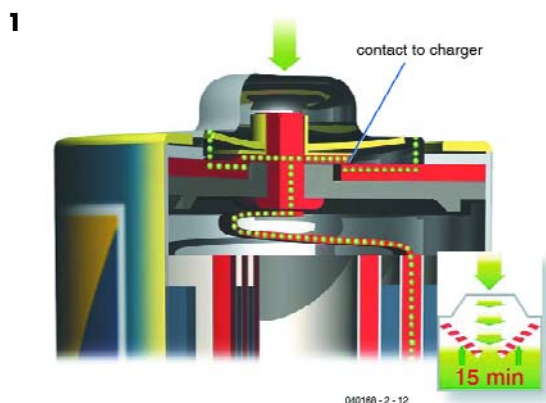
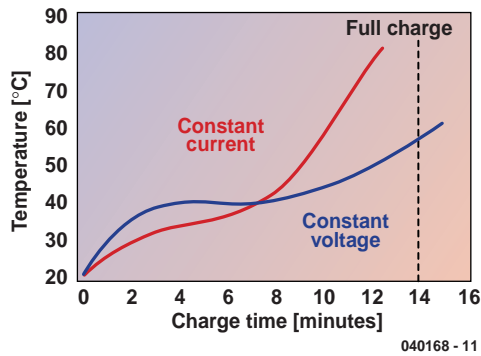


Figure 1 shows this arrangement in the closed state, when the cell is being charged. When the cell becomes fully charged, the sprung disc moves and pushes the contacts apart, interrupting the current flow.

Using this patented pressure switch it is possible to use relatively simple and therefore cheap fast chargers to charge a cell in just 15 minutes using a current of up to 7.5 A, with-

2 Rapid charging Consumer AA cell 1,700 mAh



out detriment to the life of the cell. Recharging partially-charged cells is not a problem either, since the current is reliably interrupted when the cell reaches full charge. The cost of the charging circuit is between 30 % and 50 % less than that of conventional fast chargers. The batteries also have a higher capacity (AA size 2.0 Ah, AAA size 0.8 Ah), and are not significantly more expensive than 'normal' NiMH cells. Only about half the normal amount of anode material is required to absorb the gases produced during charging, which means that the cathode can be made larger, increasing the capacity of the cell.

The I-C3 cells are not charged in a normal charger at constant current, but rather using a constant voltage: this makes the temperature rise less during charging (**Figure 2**). For safety reasons the I-C3 charger also includes a timer which stops charging after 15 minutes, and a simple over-tempera-

ture switch. The final safety measure is an external vent, as used in normal NiMH cells, which opens under extreme pressure, to prevent the cells from rupturing under any conditions.



Zinc-polymer

The American company Zinc Matrix Power (www.zmp.com) offers a new rechargeable battery with a cell voltage of 1.5 V, based on zinc, a material whose use in batteries is well established. Using a specially-developed polymer, capacities of up to 240 Wh/kg should be possible. The polymer is designed to prevent the zinc oxide which is produced during discharge from migrating around the cell. Without the polymer, the zinc would redeposit haphazardly on the anode during charging, reducing its effective surface area, and thereby reducing the current that the cell can deliver.

There is a further effect that complicates the use of zinc in batteries: the charging voltage causes the water in the electrolyte to be broken down into oxygen and hydrogen, so unless precautions are taken the cell will slowly dry out. In practice this means that environmentally-harmful heavy metals must be added to prevent the hydrogen being released. Currently more environmentally-friendly additives are used, but these are only suitable for 'normal' cells, not for rechargeable ones.

Electric aeroplanes

Electrically-powered model aeroplanes are no longer a rare sight. Less well-known, however, is that recently electric aeroplanes capable of manned flight have entered mass production. The first manned electric aeroplane flight was in 1973 by an Austrian HB-3 'Crow' motor glider, powered by a 10 kW Bosch DC motor and Varta NiCd batteries with a total weight of 125 kg. Constructor Heino Brditschka piloted the nine minute flight himself, reaching an altitude of 300 metres.

The mass-produced aeroplanes are also motor gliders. Despite considerable improvements in battery technology, aeroplanes powered by electric motors are still limited to fly-

ing short circuits and so have some way to go before finding many customers. Matters are different with modern motor gliders, which have a mechanism which allows the motor and propeller to be retracted completely inside the fuselage behind the cockpit when gliding. Motor power is only needed for take-off and to gain altitude ready for gliding, or for assistance in still air and at the end of a flight in order to avoid landing far from base should the thermals die down. The conventional solution is a relatively low-power two-stroke engine and a small reserve of fuel. This has several disadvantages not shared by the electric system, including noise, unreliability of operation (especially when



starting) and a relatively complex technology, including an electric starter, a propeller brake, and a device to stop the propeller in the correct position, all in a very small space. The electric motor provides a more elegant solution. The battery, unfortunately, is not as small or light as a fuel tank and stores less energy. However, if an electric system offers enough power for the particular requirements of the motor glider, then success in the market is just a matter of price.

The first electric motor glider with motor retraction mechanism was the AE-1 Silent by Air Energy of Aachen in Germany (www.airenergy.de), which took off on its first flight in 1997 and received German type certification as an ultralight aircraft (12 m wingspan, 200 kg net weight) in 1998. The electric motor is rated at 13 kW and weighs 8.5 kg, while the NiMH battery weighs 40 kg and stores 1.4 kWh. Air Energy is currently working on a derivative using considerably more powerful Li-ion batteries.

The move from NiMH batteries to Li-ion batteries has already been made by Lange Flugzeugbau of Zweibrücken in Germany (www.lange-flugzeugbau.com) in their Antares motor glider, which makes optimal use of the available technologies and which sets a new standard among such craft. The propulsion system uses a brushless permanent magnet DC motor with an external rotor with a rated power of 42 kW, a torque of 250 Nm and an efficiency of 98.5 %(!), weighing only 28 kg. The motor was developed at HTA Biel in Switzerland. Seventy-two Saft (www.saft.fr) VL 41 M Li-ion cells are used for energy storage. The individual cells have a capacity of 39 Ah at 3.6 V, for a total of 10.1 kWh at 260 V with a weight of about 76 kg. The Antares can climb to 1000 m in under four minutes, and to the maximum 3000 m in 13 minutes. Without thermals this altitude is sufficient for 1.5 hours of gliding and gives a range of over 150 km. The electronics handles propulsion system control and battery monitoring and includes a built-in charger (8 h charging time at 230 V). The Saft VL cells used here should achieve a life of 1000 cycles over 11 years. Information on the cells and a data sheet can be found at www.saft.fr.

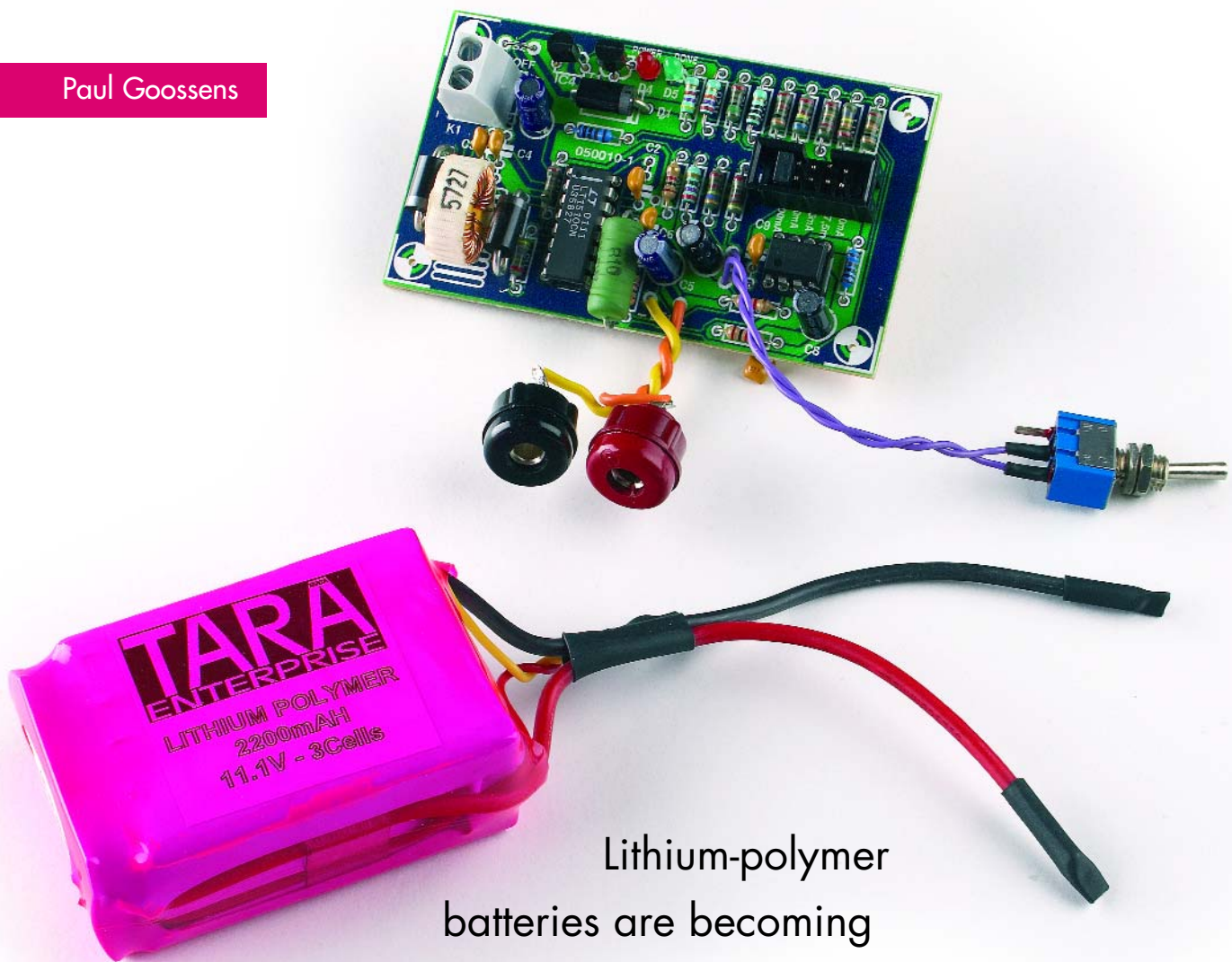
Of course, people's minds have already turned to the possibility of electric aeroplanes powered by fuel cells. An American team supported by Boeing originally announced that they would make a first flight on 17 December 2003 (the hundredth anniversary of powered flight), but at the time of writing there are no indications that a successful flight has yet taken place.



SIMPLE LIPO-CHARGER

for 2 or 3 cells

Paul Goossens



Lithium-polymer batteries are becoming increasingly common in new electronics equipment. They are superior in terms of both size and weight compared to NiCd and NiMH batteries. However, the charging method for LiPo cells is completely different from other battery types. In this article we present a simple but effective charger for this type of battery.

Lithium-polymer batteries are frequently used these days in small electronic equipment such as cellphones, PDAs and the like. In these applications it is important to the manufacturer that the battery being used is as small and light as possible. It is also important, of course, that the battery has enough capacity so that the user does not need to charge it every day. Taking all these considerations into account, you will quickly arrive at LiPos (Lithium-Polymer batteries). There are, however, a few disadvantages when using LiPos. Not all LiPos are able to deliver large currents. They can usually deliver a maximum current of 10C (see inset). There are notable exceptions, however, some LiPos being able to deliver 20C or 30C of current! Another important consideration that needs to be taken into account is the

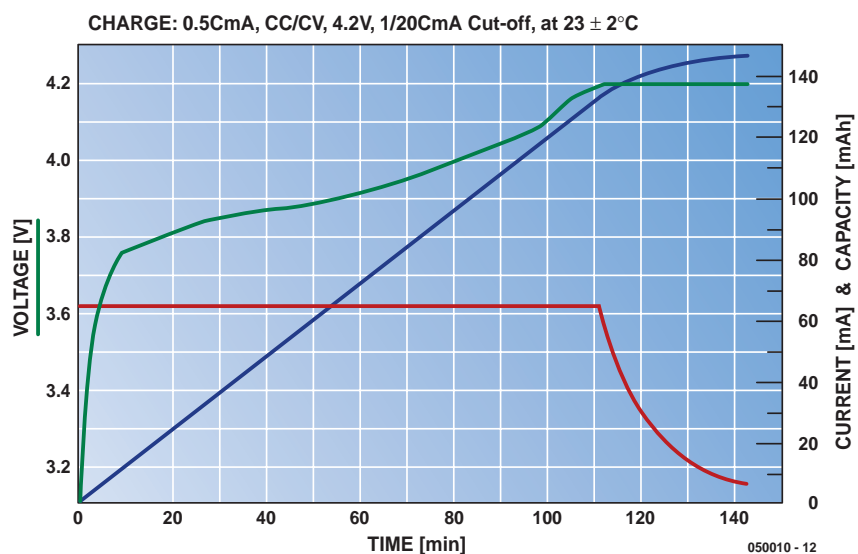


Figure 1. Charging characteristic of a LiPo cell. At a constant charging current the cell voltage rises slowly to 4.2 V.

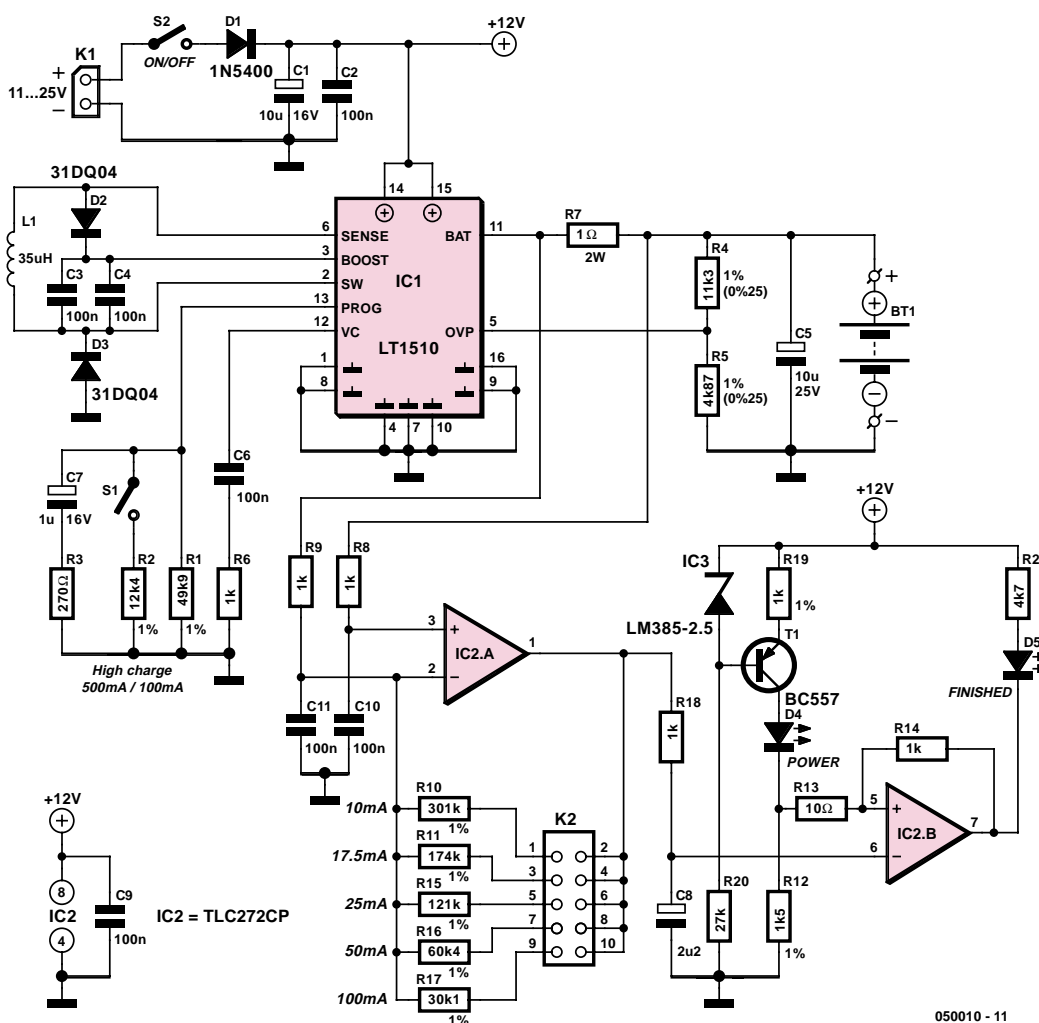


Figure 2. The schematic consists of the charger section around the special charger IC type LT1510 and an indicator section around IC2a/b.

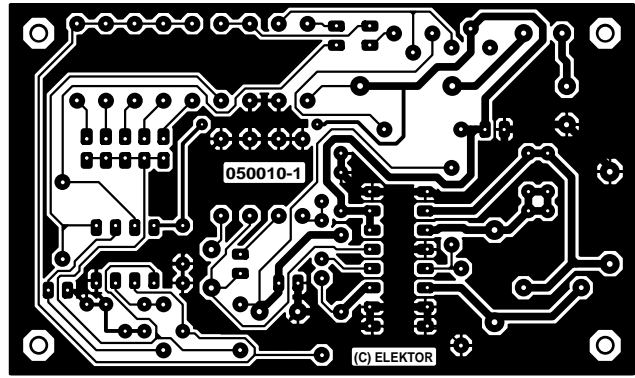
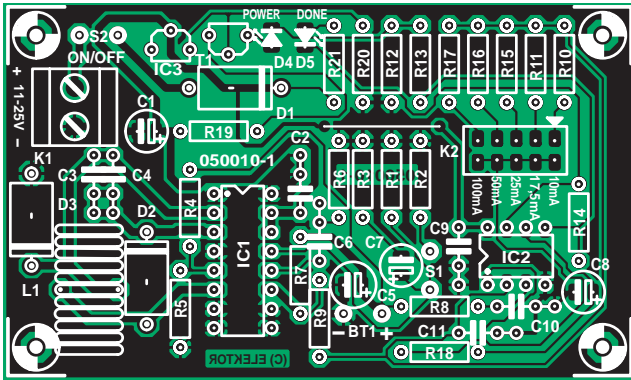


Figure 3. The PCB for the charger makes construction very straightforward. Remember to fit the wire link between K2 and C2.

COMPONENTS LIST

Resistors:

R1 = 49kΩ 9 1%
 R2 = 12kΩ 4 1%
 R3 = 270Ω
 R4 = 11kΩ 3 1%
 R5 = 4kΩ 87 1%
 R6, R8, R9, R14, R18 = 1kΩ
 R7 = 1Ω 2W
 R10 = 301kΩ 1%
 R11 = 174kΩ 1%
 R12 = 1kΩ 5 1%
 R13 = 10Ω
 R15 = 121kΩ 1%
 R16 = 60kΩ 4 1%
 R17 = 30kΩ 1 1%

R19 = 1kΩ 1%
 R20 = 27kΩ
 R21 = 4kΩ 7

Capacitors:

C1, C5 = 10μF 25V radial
 C2, C3, C4, C6, C9 = 100nF
 C7 = 1μF 25V radial
 C8 = 2μF 2 25V radial
 C10, C11 = 680nF

Inductors:

L1 = suppressor coil 35μH >2A (e.g. Digikey # M5727-ND)

Semiconductors:

D1 = 1N5400 (e.g., Digikey # 1N5400RLOSCT-ND)
 D2, D3 = 31DQ04 (e.g., Digikey # 31DQ04-ND)
 D4 = LED, red, low-current

D5 = LED, green, low-current
 IC1 = LT1510CN (Linear Technology, www.linear.com)
 IC2 = TLC272CP (e.g., Digikey # 296-1824-5-ND)
 IC3 = LM385BLP-2.5 (e.g., Digikey # 296-9559-5-ND)
 T1 = BC557

Miscellaneous:

S1, S2 = switch, on/off
 K1 = 2-way PCB terminal block, lead pitch 5mm
 K2 = 10-way (2x5) pinheader with 1 jumper
 Bt1 = 2 PCB solder pins (battery connection)
 1 wire link
 Enclosure, e.g., Hammond type 1591B
 PCB, ref. 050010-1 available from ThePCBShop

different charging method. In addition, LiPo batteries are very sensitive to incorrect use. It is not entirely impossible that this type of battery can catch fire or even explode when used incorrectly! So, when designing a charger for LiPo batteries, a few rules have to be obeyed.

Charging Method

When charging a LiPo battery, two things have to be monitored, namely the current and the voltage. Manufacturers always supply a recommended charging current in terms of C (nominal capacity in ampere-hours). This value is usually in the range from $1C$ to $2C$. In addition, the maximum voltage may never be exceeded. This is (depending on the manufacturer) 4.2 or 4.25 V per cell. Our charger is actually a power supply that is both current and voltage lim-

ited. A typical charging curve for a LiPo battery is shown in **Figure 1**. The current is constant at the beginning of the charging cycle. While charging, the voltage increases slowly to about 4.2 V. From that moment on, the voltage remains constant and the current will decrease slowly.

The charging process is just about completed when the current has dropped to about $0.05C$. At that point the battery is nearly full and any additional charging is happening only very slowly.

The circuit

Figure 2 shows the complete schematic for the charger. At its centre is an IC from Linear Technology, the LT1510. This IC contains a step-down converter together with a current and voltage limiter, so that for a

complete charger, only a handful of external parts have to be added. Because of the step-down converter, the charger is quite efficient and no additional heatsinking is necessary.

The converter requires only an external inductor, diodes and a capacitor for the boost circuit. The regulator operates at a reasonably high frequency (about 200 kHz), so the inductor (L1) can be quite small.

Adjustments

The charger, as drawn here, is intended for two LiPo cells in series. The maximum voltage of 4.2 V per cell **must never be exceeded**. That is why we have selected a maximum voltage of 8.2 V (4.1 V per cell). This adjustment is accomplished with resistors R4 and R5. The IC ensures that the voltage at pin 5 (OVP) does not exceed 2.365 V. The voltage at OVP is equal to:

$$V_{OVP} = V_{batt} \times R5 / (R4 + R5)$$

When the voltage across the battery is equal to 8.18 V, the voltage at pin 5 will be 2.47 V. In this way the IC ensures that the voltage cannot rise any further and the voltage across our battery cannot become too high.

If the charger needs to charge three cells in series, we can easily adjust the value of R4 so that the IC will limit the voltage at 4.1×3 V. At a value of 19k6 for R4, the voltage will be limited at 12.39 V. This is equivalent to 4.13 V per cell.

Since the cells are very sensitive to the applied voltage, as already mentioned, it is very important that R4 and R5 are resistors with 1% tolerance. That way, the worst-case voltage will ultimately be no more than 1% greater than intended. Also, the IC itself is not 100% accurate. That is why we designed for a cell voltage of 4.1 V instead of 4.2 V. We can now be sure that the maximum voltage will not be exceeded!

Since we are using a step-down regulator, the circuit needs to be supplied with a voltage that is a few volts higher than the desired charging voltage for the battery. Consequently, a battery with two cells in series can easily be charged with a power supply voltage of 12 V. When there are three cells in series, the power supply needs to increase to a minimum of about 15 V.

Current

The adjustment of the current is done simply with a resistor between ground and pin 13 (PROG) of IC1. C7 and R3 are necessary to stabilise the regulation loop; never omit these components! When switch S1 is open, R1 determines the maximum current that the IC will deliver. The formula is:

$$I_{const} = 2.465 \times (2000 / R1) \text{ or} \\ R1 = 2.465 \times (2000 / I_{const}).$$

In our circuit R1 has a value of 49k9. This sets the current to about 100 mA. When switch S1 is closed, the IC 'sees' a resistance value equal to the parallel value of R1 and R2. In this case, we have 49k9 in parallel with 12k4, resulting in a resistance of about 9k93. Using the above formula, a charging current of about 500 mA is obtained.

If you would like to increase the charging current, be aware that the peak currents are much higher than the programmed average current. This current

Current, C and batteries

Whenever we are discussing batteries, the term 'C' appears over and over again. For many of you, this may cause confusion. However, the concept is quite easy.

The term 'C' is used to indicate the amount of charging and discharging current of a battery. With this, 1C is the current that a full battery can deliver for a continuous stretch of one hour. With a battery of 1000 mAh this is therefore a current of 1000 mA (1 A). If we charge the cell with 0.5C, in theory it can deliver this current for two hours.

An important property of a battery is its capacity (expressed in mAh) and its current rating (the maximum current that a battery can supply, usually expressed in C). Common values are 2C, 10C and 12C.

For instance, assume you want to use a LiPo battery for your own circuit that requires a peak current of about 1 A. The battery has to be able to supply this current. The manufacturer offers several type of LiPo, each rated for a maximum load current of 2C. Our battery therefore, needs to have a minimum capacity of 500 mAh ($2C = 2 \times 500 = 1000$ mA). Only then you can be reasonably confident that the LiPo battery can actually supply this current!

flows, among others, through D1, D2 and D3, the inductor and also the IC. In addition, you have to make sure that the recommended charging current of the battery is not exceeded!

Charging Indicator

A disadvantage of the LT1510 is that there is no output available indicating how much current the IC is actually delivering. This would be a useful indication as to whether the charger is still doing something. To still be able to measure the current, resistor R7 is included in the current loop (the voltage drop across R7 is, after all, a measure of the output current of IC1). The voltage across R7 is amplified by opamp IC2a. IC2b compares this signal with a fixed voltage of 2.85 V. When the voltage at the inverting input of IC2b is lower than 2.85 V, the opamp will turn on the LED, indicating that the current has dropped below a certain value. The measuring method used here is not very accurate but sufficient to provide an indication of the state of the charger. If desired, resistor R7 can be replaced with an accurate ammeter.

The voltage gain of IC2a is adjustable with a jumper on K2. This jumper selects the resistor value for the feedback resistor for this opamp and thus the voltage gain. Next to K2 is shown to what value the current has to be reduced to (approximately) in order for LED D5 to come on.

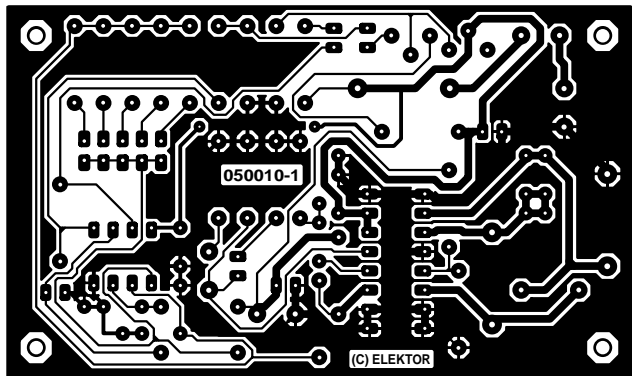
Construction of the circuit

Figure 3 shows the PCB layout designed for the LiPo charger. The design is quite spacious so the mounting of the parts should be easy. Note the polarity when fitting the diodes, electrolytic capacitors and ICs and don't forget the wire link between K2 and C2. IC2 can be fitted in a socket, but is better if IC1 is soldered directly in the board (or use a good quality socket with turned pins!).

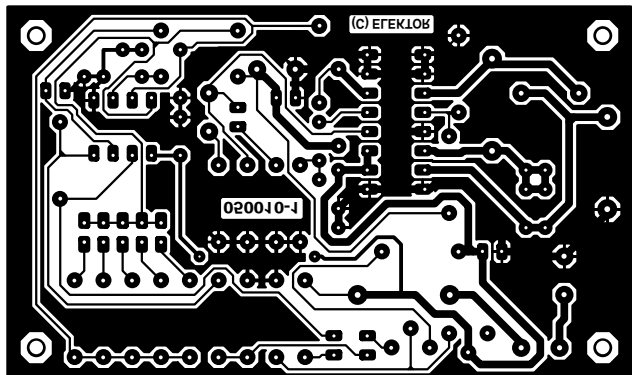
Fit a jumper to K2 for the desired current for the LED indicator. It is best to choose a value close to the C/10 rating of the battery. The circuit is now ready to charge a battery. The battery is connected to pins BT1. Now connect a mains adapter (which has to be rated for the desired output current) to K1 and plug it into a mains outlet. During the charging process, you can check the proper operation by measuring the voltage across the battery (8.2 V maximum with two cells) and the voltage drop across R7 (a measure of the charging current, 0.1 V at 100 mA).

If this is all correct, you can build the circuit into an enclosure with a few suitable connectors for the power supply and the battery. This makes it much more convenient when moving it around.

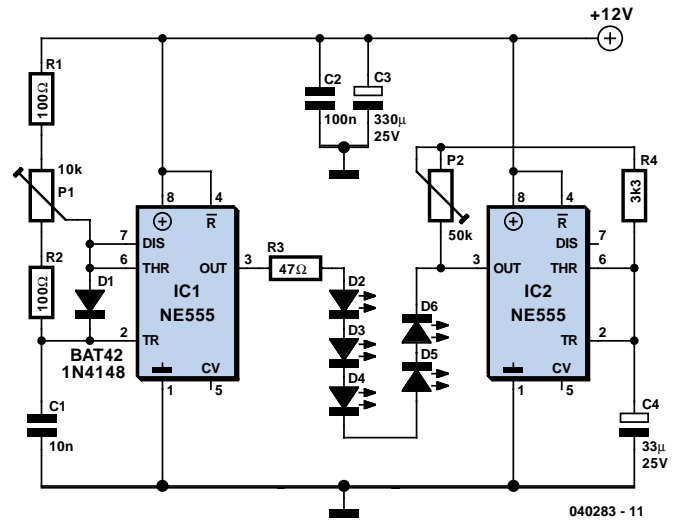
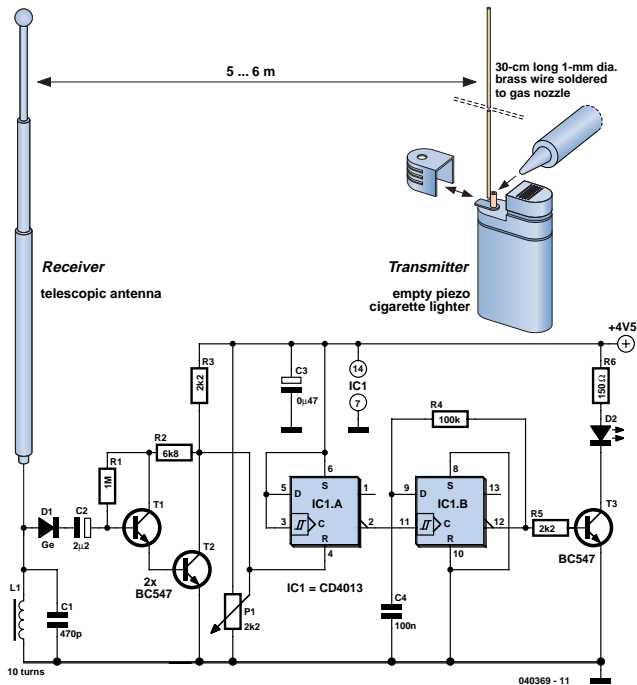
(050010-1)



non reflected



reflected



Spark Transmitter

Markus Bindhammer

In 1896, Marconi successfully covered a distance of about 3 km using electromagnetic waves. A little later, he established radio contact across water between Lavernock Point, South Wales and Flat Holm Island. The transmitter consisted of a spark inductor coupled to a dipole antenna. At the receiver side, he used another dipole and a glass tube filled with silver and nickel filings, the so-called coherer. The filings enabled the coherer to act as a 'defined' bad contact with the RF pulse energy 'rattling' the contact and thus driving the audio amplifiers connected to the output. The circuit shown here operates on the same principle. Admittedly it does not cover the same distance as Marconi's extensive radio equipment, but then you do not need to start filing away on the family's silver cutlery! The transmitter consists of an empty (!) disposable cigarette lighter with a piezo spark mechanism. On the gas nozzle of the **empty** lighter you solder a 30-cm long, 1-mm diameter (20 SWG) solid

brass wire which acts as an antenna. When the spark button is pressed, the antenna briefly radiates electromagnetic energy in the VHF FM frequency band. The effect is easy to verify by holding the transmitter close to an FM radio and 'firing' it! Compared with the primitive transmitter, our receiver is a more up to date design realised in 'all solid-state' electronics. A telescopic antenna picks up the transmitter's RF energy and applies it to a resonant circuit consisting of an inductor and a capacitor. The electrical signal is rectified by a Germanium diode (AAxxx or OAx) and amplified by a darlington stage. The collector of the second transistor causes voltage changes at pin 4 of the bistable (flip-flop) IC type 4013. The receiver sensitivity can be adjusted using preset P1. Each time the transmitter sparks away, the flip-flop toggles and switches the LED on or off. As a circuit variation, you may want to use a relay instead of a LED to allow lamps, a TV set, motors or actuators to be switched on and off using your 'primitive' spark transmitter. Your friends will be impressed.

(040369-1)

Flashing Light with Dimmer

Myo Min

Most cheap emergency flashing lights at fixed locations employ glass-based bulbs to generate light. The typical filament bulb is fragile, has a short lifetime and requires a (relatively) high voltage. To overcome these disadvantages, the author designed the above circuit, using a solid-state light source, a dimmer function and readily available ICs, all aiming at maximum compactness.

In a flashing light unit, ultra-bright LEDs have distinct advantages over normal LEDs and of course glass-based bulbs. IC1 with R1, R2, P1 and C1 supplies the pulsewidth modulated (PWM) signal that will drive the ultrabrights at a constant frequency. P1 allows you to adjust the brightness. Additional drivers are not necessary as the NE555 can source currents up to 200mA. If you need more light output, simply add some more LED strings (R3, D2~D6); up to 8 strings may be connected in parallel with the one shown.

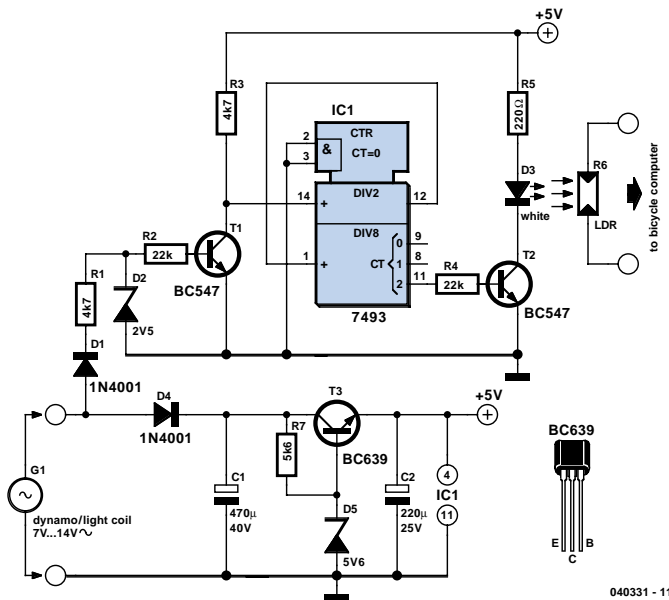
IC2 with P2, R4 and C4 forms another astable multivibrator that sinks the LED string current according to flash rate. Preset P2 controls the flash rate. C2 and C3 are decoupling capacitors to ensure stable operation at high and low frequencies. The circuit is best powered by a 12 VDC adapter with regulated output, or a step-up converter circuit around, say, a MAX761 and some AA batteries. For the latest blue and white LEDs, it is necessary to consider current limiting resistor, R3, and the number of LEDs in the string. For IC1, the PWM frequency is calculated from

$$f = (1.44 / (P1C1))$$

The frequency must be greater than 5 kHz to eliminate flicker effects.

For IC2, the design equation is the same (but using the values of P2 and C2). The flash rate should be adjusted between 0.5 Hz and 5 Hz.

(040283-1)



040331 - 11

Rev Counter for Mopeds

Peter van der Maarel

Older mopeds are not usually fitted with a rev counter, which is a bit of a shortcoming. The making or finding of a suitable indicator instrument or display is often the greatest obstacle for the hobbyist. The author of this circuit has devised a practical solution to this problem in the shape of a (cheap) bicycle computer. Such bicycle computer is easily attached to the handlebars and it usually has a large and very readable display.

The moped engine's generator is used to detect the rev speed. The generator is connected directly to the engine drive shaft and generates an AC voltage for the on-board electrical system. The frequency of this voltage corresponds with the rev speed of the engine. This frequency, however, is too high to be used directly by the bicycle computer. The solution for this is to divide the frequency of the signal by 16, using a binary counter of the type 7493, before connecting it to the cycle computer.

The generator signal is first rectified by D1, R1 and D2 and then limited to 2.5 V. Transistor T1 turns it into a usable logic signal. Counter IC1 contains four flip-flops, one after the other, which divides the signal by 16. This signal drives, via T2, the white LED D3. LDR R6 reacts to the blinking LED and is connected to the cycle computer in place of the supplied wheel sensor.

The generator signal also supplies the power for the circuit. D4/C1 provide rectification and filtering, after which the voltage is regulated to 5 V by T3 and D4.

For a correct read-out (calibrated rev counter), the bicycle computer needs to be adjusted for a wheel circumference of 889 mm or 89 cm (wheel diameter 28 inch).

Make sure that when building the circuit it is suitably protected against vibration and moisture. Mount the LED and LDR directly opposite each other and keep in mind that they need to be well shielded from ambient light.

(040331-1)

E-ONLINE

Pick your micro

CHOOSING ON THE INTERNET

Thijs Beckers

What things can you do with a microcontroller? Which type do you need for a particular application? Where do you obtain one? These questions are not easy to answer, that is why a number of manufacturers provide a helping hand.

The screenshot shows a web browser window displaying the National Semiconductor website. The page is titled "Parametric Product Selection Table" and lists various microcontroller products. The table includes columns for Part Number, Title, Price, Package, and Qty. The products listed include families of 10-bit CAN enabled CompactRISC Microcontrollers, 16-bit 120K-ws-eval 13-amp-400K-16-bit Microcontrollers, and 8-bit CMOS Flash Microcontroller with 8k Memory Dual Op Amps Virtual EEPROM Temperature Sensor 10 Bit A/D and 1.17 to 1.5V Brownout Reset.

Part No.	Title	Price	Package	Qty
C18494A(3)	Family of 10-bit CAN enabled CompactRISC Microcontrollers	\$5.9	QFP	80
C18494B(3)	Family of 10-bit CAN enabled CompactRISC Microcontrollers	\$11.9	QFP	80
C18494C(3)	Family of 16-bit 120K-ws-eval 13-amp-400K-16-bit Microcontrollers	\$10.9 \$11.9	QFP	181
C18494D(3)	Family of CompactRISC 10-bit Microcontrollers	\$10.5	PLCC	14
C18494E(3)	Family of CompactRISC 10-bit Microcontrollers	\$11.5	PLCC	14
C18494F(2)	Family of CompactRISC 10-bit Microcontrollers	\$9.5	QFP	80
C18494G(3)	Family of CompactRISC 10-bit Microcontrollers	\$11.5	PLCC	14
C18494H(3)	Family of 13-amp-400K-16-bit Microcontrollers	\$10.1 \$11.5	QFP	44
C18494I(3)	Family of CompactRISC 10-bit Microcontrollers	\$10.4	PLCC	14
C18494J(4)	8-bit CMOS Flash Microcontroller with 8k Memory Dual Op Amps Virtual EEPROM Temperature Sensor 10 Bit A/D and 1.17 to 1.5V Brownout Reset	\$4.11	SOIC-WR	241
C18494K(3)	8-bit CMOS Flash Microcontroller with 8k Memory Virtual EEPROM 10 Bit A/D and 2.7V to 2.0V Brownout Reset	\$4.4 \$4.45	LLP PLCC TSSOP	44 41
C18494L(5)	1408 120K 1.5k Microcontroller with 32k Memory Virtual 11-PRIM 1040 A/D and 4.3-3.0V to 2.4V Internal	\$4 - \$4.5	11-P MFC 1500HP	14 18 20 20 20
C18494M(6)	8-bit CMOS Flash Microcontroller with 8k Memory Virtual EEPROM 10 Bit A/D and 1.17V to 1.5V Brownout Reset	\$7.1 \$6.6	LLP PLCC TSSOP	44 41
C18494N(4)	1408 120K 1.5k Microcontroller with 32k Memory Virtual 11-PRIM 1040 A/D and 4.3-3.0V to 2.4V Internal	\$4.4 - \$4.11	11-P MFC 1500HP	14 18 20 20

"All sorts of things"; "Well, pick something, it all depends on what you want"; "Order on-line or from your friendly retailer around the corner..."; These are the kind of answers you could expect from someone who deals with these things on a regular basis. Well, that

was a great deal of help! And what now? Elsewhere in this issue there is an explanation of the things you need to keep in mind when selecting a microcontroller. There we show a handy list of web addresses that you can use when looking for the 'best' microcon-

troller for your application. There is unfortunately no (free) website to be found where you can enter all the parameters for a particular application and which subsequently suggest a chip type, manufacturer and supplier. At **EEM ICmaster** [1] under the link 'microcontroller' it is possible to do a parametric search per manufacturer. However, it is necessary to register first, but this is free and we have not yet had a problem with spam.

National Semiconductor [2] offers the possibility, on the Internet, to select a microcontroller from a product table. You subsequently can make a choice regarding the application, which leads you to another page where you can choose, based on the parameters of the desired chip.

On the **Keil Software** website [3] you can perform a parametric search of the chips supported by the software from this company. Here you can, for example, easily search for an ARM micro with certain specifications.

The **Philips** website [4] has a catalogue overview. Here you can look up the specifications for each chip. This applies only to products from Philips, of course. Unfortunately there is no parametric search facility available.

Renesas [5], established by the well known Hitachi and Mitsubishi Electric companies, does offer a parametric search option. At the top right of the page there is a link to 'parametric search'. This link leads to a page where you have to choose the product category. In this case we select the MPU/MCU product category. This results in a conveniently arranged table where you can enter all the desired parameters.

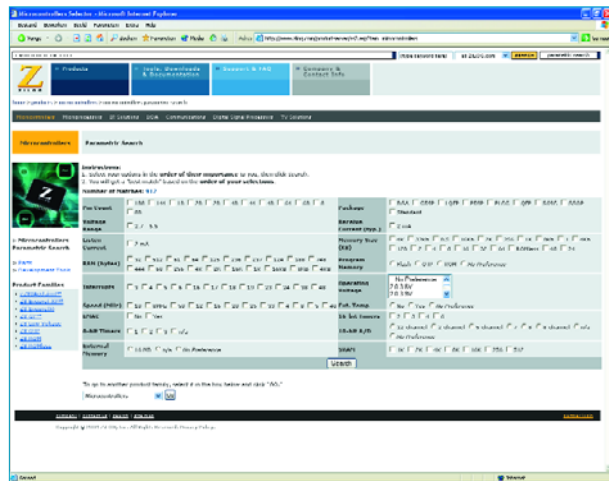
Zilog [6] too, offers the possibility to do a parametric search of their product database, just as **Freescale Semiconductor** [7]. The latter is part of the more perhaps more familiar Motorola. At Freescale we also find the parametric search option in the top right corner.

When we follow this link we get to choose from several different types of products. In this case we again select the (8-, 16- or 32-bit) microcontrollers, of course. We then get a small list where we can tick the desired search options. Ticking everything obviously results in the greatest amount of choice. On the next page (finally) we have the possibility of entering the desired features and retrieve the part number of the controller that meets the criteria.

Infineon [8], which is well known to microcontroller enthusiasts, also has microcontrollers in its product line-up. After making an initial selection between 8-, 16- or 32-bit controllers you get a page where you can choose from quite a few parameters.

Texas Instruments [9] has a wide selection of microcontrollers. Via 'Microcontrollers' (in the blue 'Products' field) we can choose from a number of controller families. On the page that now opens there is more information about different families and there is the possibility to search for a particular controller. A visit to the 'Technical Documents' section is certainly worth the effort (link on the left).

Finally we provide the link to the website from **Wichit Sirichote** [10], where a number of projects are described. Here is a selection of nice and possibly also educational projects that can easily be the trigger for a new project idea.



Internet addresses

- [1] EEM ICmaster: www.icmaster.com/Expertsearch.aspx
- [2] National Semiconductor: www.national.com/catalog
- [3] Keil Software: www.keil.com/dd/parm_search.asp
- [4] Philips: www.semiconductors.philips.com/markets/mms/products/microcontrollers/index.html
- [5] Renesas: www.renesas.com/homepage.jsp
- [6] Zilog: www.zilog.com/productserver/ss2.asp?fam=microcontrollers
- [7] Freescale: www.freescale.com
- [8] Infineon: www.infineon.com/cgi/ecrm.dll/ecrm/scripts/search/parametric_search_overview.jsp
- [9] Texas Instruments: www.ti.com/
- [10] Wichit Sirichote: www.kmitl.ac.th/~kswichit/

(045118-1)

NEW ROAD T

Automated toll collection



Ekkehard Scholz

If you are planning to drive a lorry over to Continental Europe you won't get far on any motorway before running into a toll system. Germany's sophisticated new network using GPS, GSM and infrared technology has only been up and running since the beginning of the year but the Department of Transport and UK Customs and Excise are watching closely and have announced their intention to introduce a similar system in the UK.

TOLL SYSTEMS

After a long and protracted birth the new German motorway toll collection system for Heavy Goods Vehicles (HGVs) is finally up and running. Among its innovative features are the use of in-cab GPS positioning systems and automated toll collection. Traffic flow will no longer be impeded by tollbooths and road construction/maintenance costs will be kept to a minimum without the traditional toll collection infrastructure. The specification for this new toll system was very demanding and included these main requirements:

- The toll levy takes into account the amount of pollution produced by the HGV and its axle count.
- The system must allow for a variable toll levy (according to the time of travel and road section used).
- Toll collection must be performed automatically without the need for the lorry to stop at a tollbooth (free flow system). An On Board Unit (OBU) fitted in the HGV vehicle takes care of this aspect of toll collection.
- Drivers of lorries without OBU's must be able to pre-book a journey manually using special terminals installed in petrol stations, service areas or be able to do so on-line.

In addition to fulfilling these requirements the system components should be flexible enough to support other European toll systems. This should encourage deployment of the equipment into other European countries.

Toll payment with the windows wound up

The new toll system requires an On Board Unit (OBU) to be fitted to the dashboard of every HGV (**Figure 1**). This unit is about the size of a standard car radio and performs automatic toll registration when the truck drives onto a stretch of toll road. The driver is not required to book the journey or register any paperwork. Each OBU has the individual HGV specification programmed into it when it is first installed in the vehicle.

When the lorry is started the OBU switches on and reads all preloaded information. At the beginning of any journey the lorry will not necessarily be in the same configuration as the last so the driver can enter details for each trip indicating:

- The gross weight of the lorry (a lorry without its trailer may weigh less than 12 tonnes and thus travel toll free).



Figure 1. The dash-mounted On-Board Unit (OBU) automatically logs motorway toll sections. The driver can input vehicle configuration variables (e.g. the lorry may be travelling without a trailer). (Picture: Grundig)

- The number of axles in use (the toll levy will depend on the type of trailer fitted).
- Whether the HGV is starting a new journey or if it is just a continuation of the previous journey.
- The cost centre that will be used. (This optional input helps the transportation company to identify the particular truck and journey on its monthly toll statement).

Once the data has been entered the OBU will be active. It will now use information from its GPS receiver

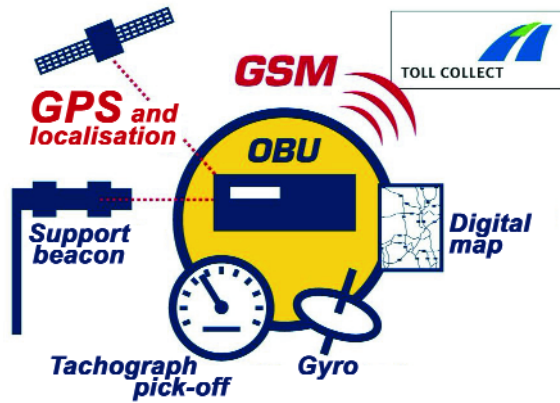


Figure 2. Using GPS (Global Positioning System) together with Tachograph and gyroscope data the OBU pin points the truck position on a stored digitised map to a resolution of 10 m. Any toll payable is relayed to the central toll office using a mobile phone (GSM) network. (Picture: Toll Collect)

along with other positional data (Figure 2) to compare its coordinates with a digitally stored map of the motorway system and determine if a toll is payable for the stretch of road that it is on. The positioning works on the same principle as an in-car navigation system. Additional motion sensors are used to improve the system accuracy. A gyroscope fitted with piezo strain sensors provides inertial information while data from the vehicles tachograph supplies speed and distance information. This dead-reckoning system functions independently of the GPS system. The vehicle positional data from both systems are continually compared. The dead reckoning system ensures the OBU can maintain its positional accuracy for short periods when there is no GPS signal available.

The OBU calculates the toll based on the current tariff and road section that the lorry is travelling on. The toll is

stored and then sent to the toll control centre using a mobile (GSM) phone link (see Figure 3). The data is sent periodically either when the toll tally reaches a pre-defined level or after a set time interval.

300 control gantries

A vital part of the toll system responsible for vehicle detection and communication is mounted on the 300 overhead gantries (gates) scattered over the German motorway network. As a vehicle approaches the gantry, a detection and tracking unit (developed by Daimler-Chrysler) uses DSRC (Dedicated Short Range Communications) to sense the vehicle position and determine when it will pass the gantry. At this point a 3D outline is captured and the vehicles weight category assessed. Any truck with a gross weight of less than 12 tonnes is not liable for duty so all information collected on this vehicle is erased.

When it is determined that duty is payable, equipment mounted on the gantry interrogates the vehicle using DSRC infrared signals to determine if an OBU is fitted. When no reply is received a toll violation will be indicated if the driver has not previously booked the journey. A camera now records the registration number and passes it to the central toll office where it can be automatically compared to all the registration numbers for manually pre-booked journeys. If the journey has been booked all vehicle data is again erased. Where no match is found a toll violation incident is recorded.

Policing the system

Situated behind around half of the control gantries are areas where the motorway officials can pull over vehicles for investigation. The gantries are equipped with two-way radio links allowing motorway officials situated in a parking area or by an exit to communicate with the toll centre. Information relating to toll-evading vehicles will be available in a matter of seconds after it has passed the control gantry.

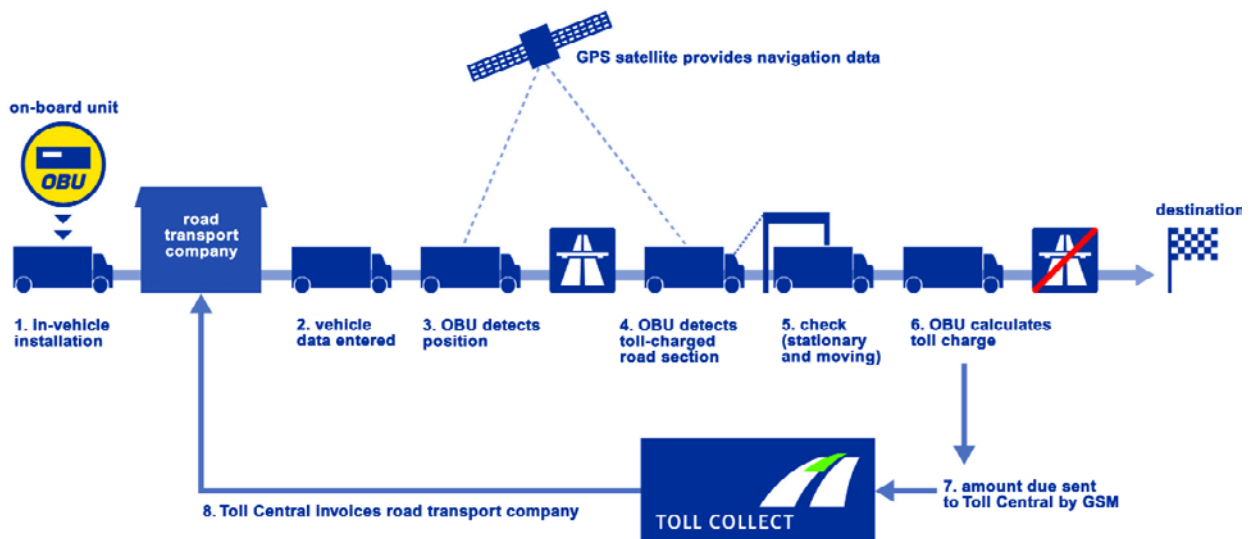


Figure 3. Automatic toll collection for a truck with an OBU fitted. (Picture: Toll Collect)

There is also a team of around 300 mobile patrol units spread over the entire network carrying out checks on the moving traffic 24 hours a day ensuring that the tolls are enforced. These units operate in the flow of traffic allowing the operators to check vehicle details to determine if the toll is payable. The mobile unit can also determine if the HGV is registered with the toll system and if the axle count is correct. Any HGV not registered will be checked with all the pre-booked vehicles stored in the central data bank and if there is no match the lorry will be pulled over and investigated. A fine will be registered if the toll has not been paid.

Both the fixed automatic vehicle recognition and mobile units collect vehicle details without interrupting the flow of traffic. The system checks if the correct toll has been paid but does not perform fine collection.

The vehicle position can be pinpointed on any of the 5,200 sections of road that make up the 12,000 kms of motorway in Germany. The present system is relatively new and so far only the basic features have been implemented to prove this new technology. In the coming year we can expect to see the concept of 'road pricing' come one step closer when trucks are kitted out with the next generation (dynamic) OBU version 2.0 to allow the inclusion of major roads in the scheme. This should prevent HGV drivers taking alternative routes to avoid tolls.

Here in the UK, best estimates indicate that an equivalent system for heavy goods vehicles will not be functioning before 2008 and any universal road-pricing scheme will not be introduced before 2014. The Department of Trans-

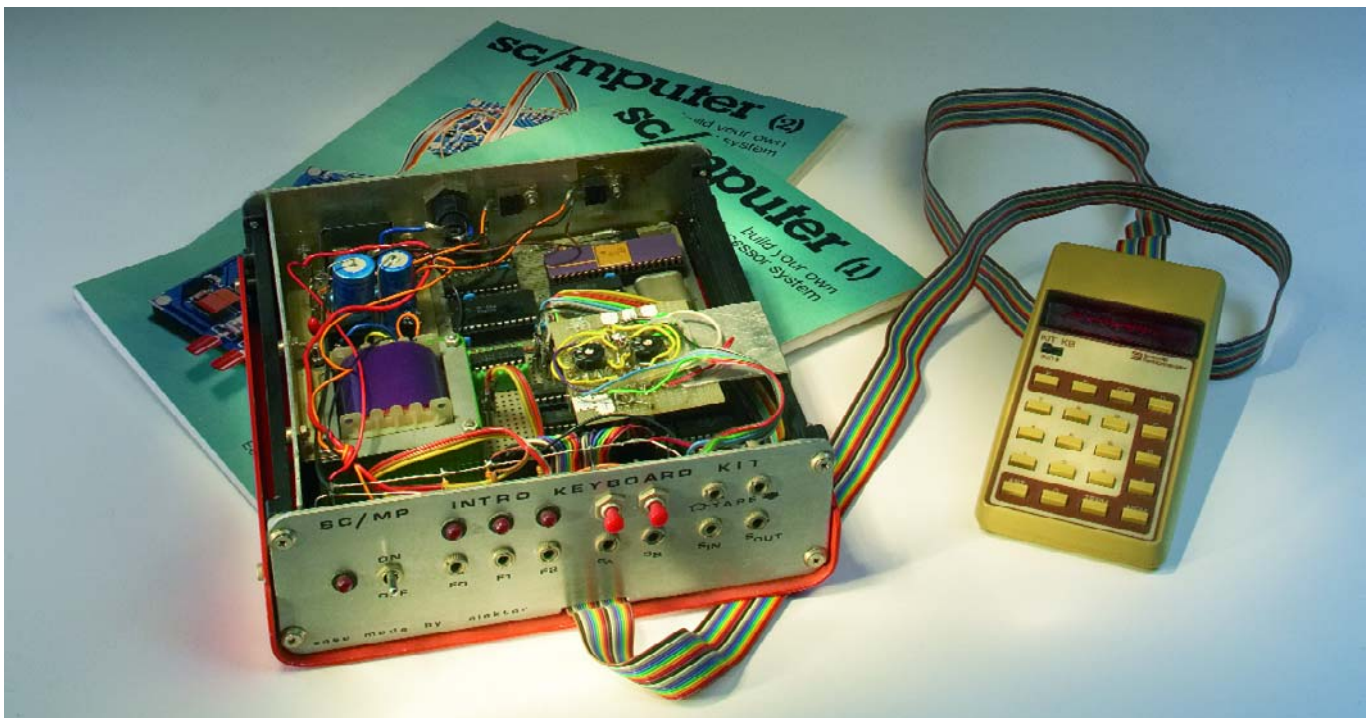
port have indicated that this system could be applicable to the UK motorway network if the OBU can resolve vehicle position to within 60 m (poorer resolution could result in vehicles incurring tolls when travelling on free roads running parallel to toll roads). The German company who developed the system (Toll Direct) claim the system has a positional resolution of 10 m.

(045121)



The existing motorway gantries fitted with vehicle recognition sensors for toll collection. (Picture: Toll Collect)

Advertisement



Elektor SC/MP Computer (1978)

Jan Buiting

It is safe to say that anyone who wishes to maintain a serious interest in electronics [...] must get to grips with microprocessor technology. The implications of the microcontroller revolution are so far reaching that anyone who fails to do so will become hopelessly out of date. [...]

Preface to a new magazine on the latest 512-bit microcontrollers running at 300 GHz? Strongworded but unimpressive commercial advice from Microchip or Atmel?

No, you just read the preface to a series of articles started in November 1977 describing Elektor's SC/MP Microcontroller System.

The impact of the series was enormous, very few competitors having had the audacity to bite the ?P bullet. And controversial it was, too, staunch supporters of good old analogue and digital design declaring that the micro-

processor in the hands of the home worker meant the end of hobby electronics.

SC/MP stands for *Simple Cost-effective Microprocessor*. The product was officially called ISP-8A/500D and came from National Semiconductor. A later version of the chip employing N-MOS technology was dubbed ISP-8A/600D.

The SC/MP, housed in a 40-pin DIP case belongs in the category of very early microprocessors, sporting a traditional structure with registers, an accumulator and the ability to address external memory. As such, it was National Semiconductor's attempt at securing market share. Not a very successful attempt, as it turned out later, Intel, Motorola, Signetics and even RCA competing fiercely in a then totally new arena.

Reportedly the then almighty Elektor Editor-in-Chief in the end decided to swing towards the SC/MP in favour the Motorola 6800 because the NS product was... cheaper!

From the start, the Elektor SC/MP was a cluttered system, despite attempts at making wiring diagrams look as orderly

as possible. If the Elektor prototype shown on a photograph in the November 1977 issue is just 'two boards strung together with lots of wires' the average reader was sure to make an even greater mess of it.

But it all worked — readers diligently etched and drilled their own PCBs, faithfully typed in long hex listings from the magazine pages, and started sending their programs and home-made extension circuits to Elektor. Within months the SC/MP was famous and occasionally we still meet older (not necessarily wiser...) readers proudly telling us the SC/MP was the first computer they built to impress their friends and family members with.

The publication of the basic SC/MP system was followed by several 'bits-and-bobs' articles discussing subjects like BASIC, the Singing SC/MP, a disassembler, an RF modulator, a 300-baud tape/record interface, a 4-K RAM board, games like Luna and Keyplay, and the famous Elbug monitor. Going from strength to strength, an afterburner effect occurred when were published and the SC/MP

was again mentioned in combination with the 'TV Typewriter' (ElekTerminal & ACII Keyboard), a blockbuster project published in December 1978. I now wonder how many readers used the TV Typewriter to program microcontroller systems not using the SC/MP but, for example, the pricier but eventually far more popular 6800?

The photograph shows what I believe to be a system built around an original SC/MP development board supplied to Elektor by National Semiconductor. The system looks specially designed to read and write cassette tapes and is possibly the Master Tape generator for Elektor's Software Service (ESS) who at the time supplied SC/MP software on 45 rpm vinyl records. Anyone got copies of ESS 001 and 002?

(045113-1)

Retronics is a monthly column covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcomed; please send an email to editor@elektor-electronics.co.uk, subject: Retronics EE.

QUIZZ'AWAY



Martin Ohsmann is a Professor of Electrical Engineering and Information Technology at FH Aachen and a long-time contributor to *Elektor Electronics*. Through *Quizz'away* he aims at stimulating thought, speculation, construction and simulation as well as raise interesting questions.

This month we look at an everyday item: the humble coin, be it a 2-pound, or a 1- or 2-euro piece. Money, including coins, seems to have a strong attracting effect on people. But how do the above coins behave within a magnetic field? Will they be attracted or repelled? To find out, we suspend a coin freely in front of an electromagnet (**photograph**). Next, we pump a strong current of 8 to 10 A_{rms} through the coil. First, we apply an alternating current of 50 Hz and then we repeat the experiment using a frequency of 20 kHz.

This month's question is:

Is the coin attracted or repelled?

Hints

The two-pound coin and the 1- and 2-euro pieces contain nickel/brass and copper/nickel alloys (www.predecimal.com/dictionary.htm). You can use either of these three coins. The coil consists of an Epcos coil former type P36x22 with 50 turns of 90x0.1 CuL RF litz wire. The coil former sits on a ferrite core half type ETD 44/22/15. The inductance of the coil is about 240 mH. In order to obtain an effective current of 8 A through the coil you'll need about 240 volts — an extremely dangerous undertaking as the reactive power amounts to about 2 kilowatts.

Editor's note: all relevant electrical safety precautions should be observed and the nature of the above experiment makes it definitely unsuitable for any place other than a professionally equipped laboratory.

The question may be answered by simple reasoning — complex maths should not be necessary. For a lab experiment, find someone with experience in the design of heavy



A one-euro coin freely suspended in front of an electromagnet.

power supplies. The 50-Hz current may be obtained from a toroidal transformer on which a suitable secondary is wound. Other coil arrangements are also possible, as long as the quality factor is high enough at 20 kHz to ensure the large current flow. At 20 kHz, a high current can be obtained from a full bridge equipped with good MOSFETs ($R_{DS(on)}$ 0.0x ohm) powered by a 12-20 VDC supply and controlled (via the gates) by a 20-kHz square wave. The coil is connected to the bridge via a series capacitor dimensioned for resonance at about 20 kHz together with the coil. Be sure to use a premium-quality MKP capacitor rated for at least 600 V and capable of passing 20 A at about 200 V (these parts are found in SMPSUs). The square wave frequency is adjusted for resonance to occur and all being well the coil will pass the required 8 amps or so.

Quizz'away and win!

Send in the best answer to this month's Quizz'away question and win an

Elektor Flash Micro Board Starter Kit

(010208-91)

worth £69. The kit contains the ready-assembled and tested board with connecting cable and software on CD-ROM (see www.elektor-electronics.co.uk)

All answers are processed by Martin Ohsmann in co-operation with Elektor editorial staff. Results are not open to discussion or correspondence and a lucky winner is drawn in case of several correct answers.



Please send your answer to this month's Quizz'away problem, by email, fax or letter to:

**Quizz'away, Elektor Electronics,
PO Box 190,
Tunbridge Wells TN5 7WY, England.
Fax (+44) (0)1580 200616.
Email: editor@elektor-electronics.co.uk,
subject: 'quizzaway 4-05'.**

**The closing date is
29 April 2005**

(solution published in May 2005 issue).
The outcome of the quiz is final.
The quiz is not open to employees of Segment b.v., its business partners and/or associated publishing houses.

Solution to the February 2005 problem

(p. 78; Lamp Intensity)

The equivalent circuit is shown again in **Figure 1**. Normally, you would expect the current through R1 to rise when S1 is closed. After all, a parallel network (R2//C) has a smaller resistance than the individual parts, and the voltage source then has to supply more power. However, with the component values shown (R1 = 100 Ω, R2 = 200 Ω, Z_c = -j200 Ω) the amount of current through R1 is not affected by S1 being open or closed, hence **the lamp brightness is unchanged**. This only applies if the capacitor value equals

$$C = \frac{1}{2\pi f \sqrt{2R_1 R_2}}$$

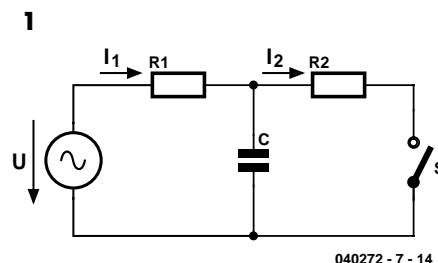
where *f* is the operating frequency. The function of the capacitor may be taken over by a coil as long as it has the same (absolute) impedance at *f*. In addition, the power dissipated in R1 with the switch closed can be made to equal that in R2 simply by selecting R2 = 2R1 and that is the case in our circuit. The operation of the circuit may also be explained with the aid of vectors and AC theory (**Figure 2**).

When switch S1 is open, the total impedance Z₀ of series network R1 and C is easily calculated as

$$Z_0 = R_1 + Z_c = (1-2j) \cdot 100 \Omega$$

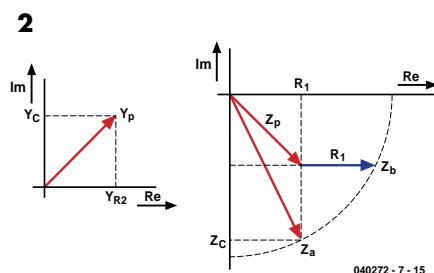
In the right-hand part of Figure 2 (impedance area), this calculation can be verified. A slightly more complex situation arises when the switch is closed, because we are then looking at a parallel and a series circuit. For the parallel value it is easiest to determine the conduction (Y_p; left-hand part of Figure 2) because it is the sum of the conduction in the two branches. Summing is done with vectors, but this time in the conduction area (left-hand part of Figure 2). The conduction of R2 and C have the same absolute value but are at 90 degrees in the complex area. Figure 2 also shows that adding these two vectors results in the conduction increasing by a factor of √2 at a phase angle of -45 degrees. That conduction equals an impedance of 200 Ω / √2 at a phase angle of -45 degrees. Entering it into the right-hand part of Figure 2 and by vector-adding the value of R1, we not only find the total impedance Z_b but also that Z_a and Z_b are at the same distance from the origin, namely on a circle with a radius Z = 100 Ω / √5. In other words, the circuit has an impedance that's not affected by operating switch S1!

The explanation of this 'AC paradox' is found in the fact that the value of current I₁ remains unchanged when the switch is operated, although the phase angles does change. With the switch closed, more power is taken from the source and dissipated in R2. By contrast, R1 dissipates the



same amount of power irrespective of the switch position. In the impedance area we can also see that R2 carries a current I₂ that's √2 times smaller than the total current I₁ flowing through the parallel circuit. Since R2 is two times the value of R1, the formula P = I² R dictates that the power dissipation in the two resistors is equal.

Moral of the story:
"Watch that phase angle!"



Advertisement

Prototypes and small series up to 1000 pcs

Value for Money

Up to 6 layer boards

Always Available

Attractive price

On-line Printed Circuit Boards

Double Sided, 1.6 mm,
Plated Through,
Green Solder mask,
Hot Air Level,
1 White Legend.

Price including
tooling + plotting + VAT

100 mm x 160 mm
1 pc = 45.74 Euro

100 mm x 80 mm
5 pcs = 68.97 Euro

www.ThePCBSHOP.com



27(C)512 Emulator

Despite the inroads of Flash technology, lots of microcontroller applications still employ an EPROM for program storage. The reasons are probably best summarized in two words: 'cost' and 'tradition'. The downside of an EPROM is the need for an emulator when it comes to debugging. Our emulator is based on a combination of an FPGA and an AT90S8515 controller. For connectivity it offers SPI, JTAG and RS232 interfaces.

Theme Plan for 2005	
January	Power Supplies
February	Wireless
March	Sound
April	Microcontrollers
May	Sensors
June	Environment
July/August	Summer Circuits
September	Test & Measurement
October	Security
November	CAD Software
December	Optoelectronics

USB-GPS

This article describes an interface between navigation applications and a miniature GPS module coupled to an equally small active antenna. The modular structure of the project allows you to use selected parts for your own applications.



Serial Bus Systems

Two in-depth articles will deal with the variety of systems available to route signals in and between equipment while using as few 'wires' as possible. Protocols and specifications are described as well as guidance for designers.

Also...

The Digital Future; PetCheck; Lap Counter; Delphi for Electronic Engineers (5); Magnetic Flux Meter; Theremin; NMRA Model Train Programmer; eZ80 Acclaim!

RESERVE YOUR COPY NOW!

The May 2005 issue goes on sale on Saturday 23 April 2005 (UK distribution only).

UK subscribers will receive the magazine a few days before this date.

Article titles and magazine contents subject to change.

NEWSAGENTS ORDER FORM

SHOP SAVE / HOME DELIVERY

Please save / deliver one copy of *Elektor Electronics* magazine for me each month

Name:

Address:

Post code:

Telephone:

Date:

Signature:



Please cut out or photocopy this form, complete details and hand to your newsagent. *Elektor Electronics* is published on the third Friday of each month, except in July. Distribution S.O.R. by Seymour (NS).

INDEX OF ADVERTISERS

Allgood Technology, Net Links	www.allgoodtechnology.com80	Net Links80, 81
Audioexpress, Net Links	www.audioexpress.com80	New Wave Concepts, Net Links	www.new-wave-concepts.com81
Beta Layout, Net Links	www.pcb-pool.com	19, 80	Number One Systems	www.numberone.com66
Burn Technology LTD, Net Links	www.burntec.com80	Nurve Networks	www.xgamestation.com49
CMS	www.cms.uk.com69	O2M8	www.o2m8.com19
Compucut, Net Links	www.compucutters.com80	PCB World, Net Links	www.pcbworld.org.uk81
Conford Electronics, Net Links	www.confordelec.co.uk80	Peak Electronic, Net Links	www.peakelec.co.uk81
Cricklewood	49	PHYZX, Net Links	www.phyzx.co.uk81
Danbury, Net Links	www.livinginthepast.demon.co.uk80	Picdos, Net Links	www.picdos.com81
Display Electronics	www.distel.co.uk	32	Pico	www.drdaq.com4, 65
Easysync, Net Links	www.easysync.co.uk75, 80	Quasar Electronics, Net Links	www.quasarelectronics.com64, 81
Elexol, Net Links	www.elexol.com80	RD Research	www.spice-software.com33
Elneec, Net Links	www.elneec.com80	Robot Electronics, Net Links	www.robot-electronics.co.uk81
Ervan International, Net Links	www.ervan-int.com80	Telnet, Net Links	www.telnet.uk.com81
Eurocircuits	www.thepcbshop.com79	Total Telematics	www.total-telematics.com75
ExpressPCB	www.expresspcb.com19	Ultraleds, Net Links	www.ultraleds.co.uk81
Forest, Net Links	www.fored.co.uk80	University of Derby	www.vertigo.derby.ac.uk19
Future Technology Devices, Net Links	www.ftdichip.com80	USB Instruments, Net Links	www.usb-instruments.com81
Futurlec, Net Links	www.futurlec.com80	Viewcom, Net Links	www.viewcom.f9.co.uk81
ILP Electronics Limited, Net Links	www.ilpelectronics.com80	Virtins Technology, Net Links	www.virtins.com81
Ipeva Limited, Net Links	www.ipeva.com80			
Jaycar Electronics	www.jaycarelectronics.co.uk3			
JLB Electronics, Net Links	www.jlbelectronics.com80			
Komcard, Net Links	www.komcard.com80			
Labcenter	www.labcenter.co.uk88			
London Electronics College, Net Links	www.lec.org.uk80			
Matrix Multimedia Ltd	www.matrixmultimedia.co.uk66			
Microchip	www.microchip.com/seminars2			
Milford Instruments	www.milinst.demon.co.uk31			
MQP Electronics, Net Links	www.mapelectronics.co.uk80			

**Advertising space for the issue of 17 May 2005
may be reserved not later than 19 April 2005**

with Huson International Media – Cambridge House – Gogmore Lane – Chertsey, Surrey KT16 9AP – England – Telephone 01932 564 999 – Fax 01932 564998 – e-mail: gerryb@husonmedia.com to whom all correspondence, copy instructions and artwork should be addressed.