

FREE DVD

elektor electronics

leading the way

November 2005

£3.70

www.elektor-electronics.co.uk

DESIGN IT ON THE PC

- schematics, simulation and circuit boards
- from concept to reality
- programming in pictures

(almost) ALL E-CAD PROGRAMS ON **FREE DVD**



ELECTROSMOG again!
UK's HPA report says:
EMF CAN MAKE YOU SICK

E-blocks
handy modules,
great strides





Volume 31, Number 348, November 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.

Publishers: Elektor Electronics (Publishing), P.O. Box 190, Tunbridge Wells TN15 7WY, England. Tel.: (+44) (0)1580 200657, fax: (+44) (0)1580 200616. www.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, Ernst Krempelsauer, Jens Nickel, Guy Raedersdorf.

Design staff: David Daamen (head of design), Ton Giesberts, Paul Goossens, Luc Lemmens, Karel Walraven

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

Graphic design / DTP: Ton Gulikers, Giel Dols

Managing Director / Publisher: Paul Snakkers

Marketing: Margriet Debeij (m.debeij@segment.nl)

Subscriptions: Worldwide Subscription Service Ltd., Unit 4, Gibbs Reed Farm, Pashley Road, Titchhurst TN15 7HE, England. Telephone: (+44) (0)1580 200657, Fax: (+44) (0)1580 200616. Email: admin@worldwidesubscriptions.com. 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: r.elgar@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

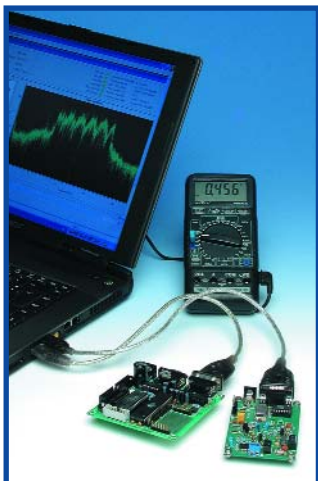
The circuits described in this magazine are for domestic use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, software carriers and article texts published in our books and magazines (other than third-party advertisements) are copyright Segment. 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.

Disclaimer

Prices and descriptions of publication-related items subject to change. Errors and omissions excluded.

© Segment b.v. 2005

Printed in the Netherlands



USB to Centronics

Dear Editor — the article ‘USB to RS232 Hurdle Race’ caught my eye as I was just starting out with a simple kit based on an Atmel microcontroller. I was hoping the article would provide an alternative to the rather clumsy method, used with the kit, of using an LPT port for uploading programs. Unfortunately, I have a printer attached to that port and the relevant connector is difficult to reach on the back of the PC. I also object to the use of the LPT drive signals (and RS232 signals) for the programming interface. At the end of the day, although your article did not present a solution to the problem it did succeed in explaining the real problems you run into when using a USB to RS232 converter in order to program a microcontroller via control signals. After reading your article, I quickly abandoned the idea of using a USB to Centronics/LPT converter. Recently, I came across an alternative solution, ‘project AVR910’, on www.klaus-leidinger.de/mp/index.html. The circuit employs a microcontroller and is based on an older Atmel application note. Data is sent to the microcontroller in block mode — normally via the data lines, not the control lines. As a matter of course, the PC has to be able to handle this different method of data transport.

Ronald Marr (UK)

Thanks for that Ronald, the word is being passed to other readers.

Sallen & Key vs. MFB

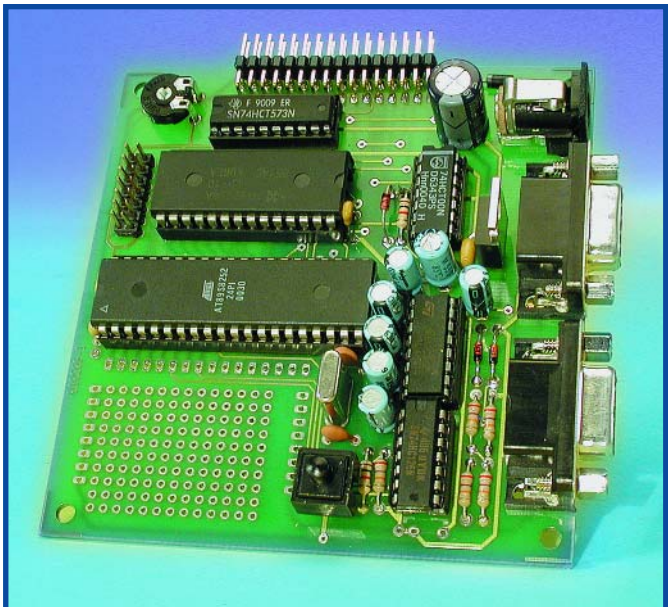
I really enjoyed reading the article ‘THD: Sallen-Key versus MFB’ in the July/August 2005 issue which I obtained through downloading — thanks!). The article would have been even better if you had included a graph showing the high-frequency response. In that rejection range one will see the response of a Sallen & key filter rising again for high f values, indicating that the opamp is over exerted. Its output impedance will rise considerably, causing the output to lose its ideal voltage source characteristic.

Referring to your schematic, C1 and C2 then enable a virtual path between input and output. In MFB configurations, this effect is far less pronounced, preventing higher frequencies to ‘leak’ from input to output. Put it to the test! My comment is intended as a positive contribution rather than criticism!

H. Huynen (Netherlands)

Live recordings using portables

Dear Jan — I am using a Mini-Disc recorder to record live



Programming with the Elektor 89S8252 Flash Micro Board

Microflash should not be used anymore, especially under Windows XP, as it often produces erratic results.

We now advise downloading and using ATMELISP from www.modul-bus.de/mbnews/mbnews02/isp.htm. (atmelisp.zip, Version 21.5.2005, 709 k).

After pressing “DK7JD” in the lower left corner the program is configured correctly for the Elektor 89S8252 Flash board. Also, select the right COM port in the upper left corner.

About the cable: use a 1:1 RS232 cable with at one side a male connector and at the other side a female connector. Connect pin 1 with pin 1 and so on.

Note: The pin numbers on male and female connectors are mirrored!

music performances. A frequent problem I run into is that the sound level is so high that the microphone starts to distort the signal (‘brick walling’). Actually, the source of the trouble is not so much the microphone itself but the supply voltage it gets from the Minidisc recorder. Sure, commercial units (with or without internal preamplifier) can be found to remedy the problem but DIY plans are hard to find. May I suggest Elektor designs a circuit including a preamp for Minidisks or iPods, enabling us to make high quality live recordings?

Mike Duffin (UK)

One of our editors is also a keen sound recordist and built the ‘Phantom Supply from Batteries’ from the July/August 2005 issue. He extended the circuit with a standard power supply, allowing the circuit to operate with a wall cube. Although this circuit still does not include a

microphone preamplifier, a good solution would be the INA217-based ‘Balanced Microphone Amplifier’ published in the July/August 2003 issue. This is a reasonably simple yet high quality design that can be built on a compact little board. The INA217 may be replaced by an INA163, which will produce even less noise. Designing a similar circuit for the iPod is problematic because of the iPod’s peculiar construction.

Metal Detector Coils

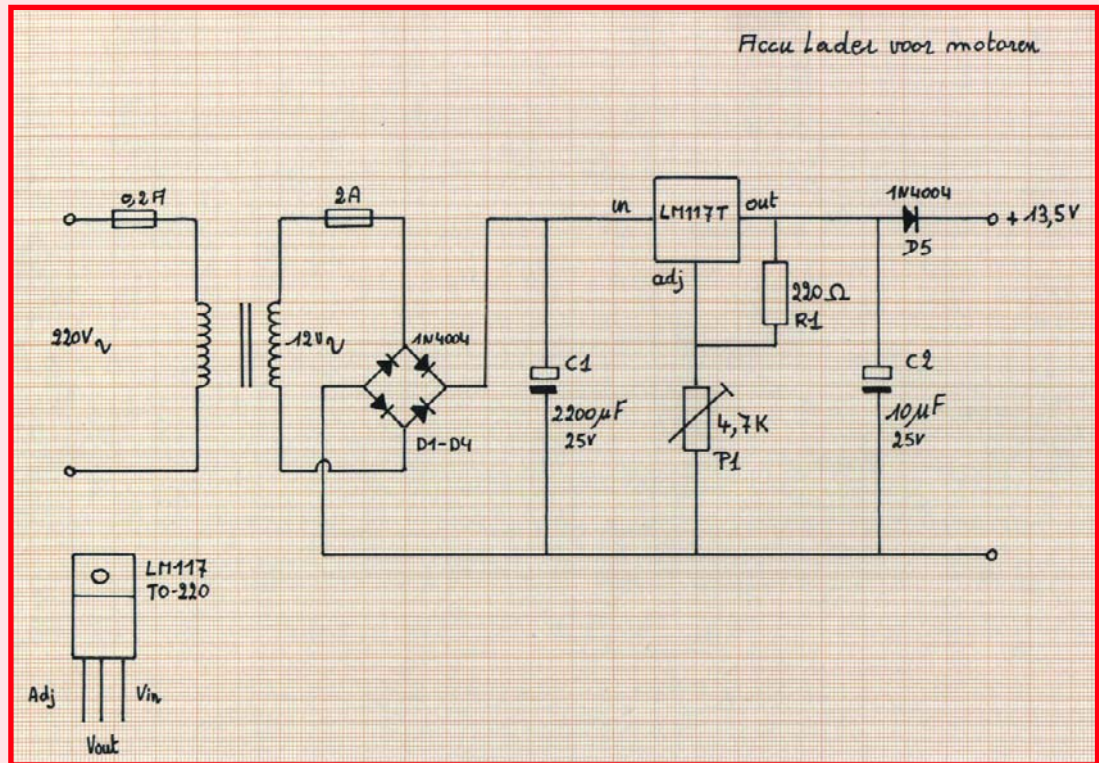
Dear Jan — despite a lot of effort both in reading your brilliant articles and building things on an experimental basis, I have been unable to get to grips with the construction of the search head in your latest metal detector (‘CCO Metal Detector’, July/August 2005, Ed.). The electronics is working just fine but I keep struggling with the wires in the

Rejektor

Under this heading we will occasionally publish circuits, ideas and suggestions that did not make it to full publication in this magazine for various reasons (like lack of space).

Battery Charger

Johny Verhoeven sketched this circuit diagram of a simple battery charger. Similar circuits have been published many times before. Although an LM117 is used in this circuit, any similar adjustable voltage regulator should do the job equally well. Users should note that this is a standby charger — if the battery is deeply discharged, the voltage drop across the regulator and hence the LM117's dissipation will rise beyond safe limits and calling adequate cooling. If the charger is to be up to flat batteries as well, then a switching regulator like National Semiconductor's LT1074 is better suited to the job.]

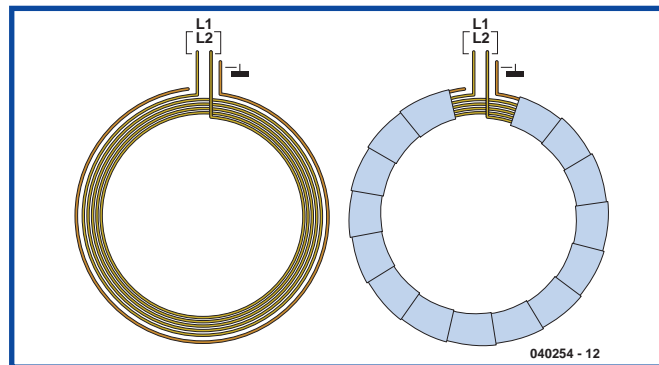


coils and how to connect them up. Please help!

Tony Birrell (UK)

From correspondence with other readers, we can conclude that usually it's the coil screening that's not properly understood. Here is some supplementary information. The coil is covered with a conductive foil. If you use a material that can be soldered, for example, copper or brass foil, the foil is directly connected to ground. Aluminium (kitchen) foil may also be used. Unfortunately, this cannot be soldered, so to ensure a proper electrical contact a length of bare wire (connected to ground) has to be added to the coil windings. When the foil is applied around the coil, it will touch the bare wire and provide the ground contact.

It is essential that neither the screening foil nor the bare wire forms closed loop as that equals a short-circuited winding greatly reducing the detector's sensitivity. Although we've usually shown a



round coil in our drawings, square, rectangular or D-shaped alternatives are also perfectly adequate. The construction also applies to the 'One-Component Metal Detector' from the October 2004 issue of *Elektor Electronics*.

Corrections & Updates

Balancing LiPo Cells
July/August 2005, p. 114,
054026-1

In the circuit diagram, IC1 should be identified as a MAX6382XR41D2-T. Pin 1 should be connected to ground, and pin 3, to the positive supply voltage. Pin 2 is the RESET output.

Precision Barometer/Altimeter
September 2005, p. 54,
040313-1

Problems with the real-time clock may be solved by downloading updated software from our website. A 56-pF capacitor between pin 12 of IC1 and ground will improve the clock accuracy.

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.

Micromouse robotics championship



The race is on to build robotic 'mice' for the 2006 international robotics championship, Micromouse, when it returns to Birmingham's Millennium Point, on Saturday 10th June next year. The event will be hosted by UCE Birmingham's Technology Innovation Centre (TIC), for the third successive year. Organisers have launched Micromouse 2006 well in advance, to accommodate the rapidly growing interest from new competitors, both from the UK and overseas.

With this early announcement, competitors from schools, colleges and universities, will now have time to develop their entries from the start of the new academic year.

Micromouse 2006 will see mini-robots, racing to reach the centre of a specially constructed maze, in the fastest possible time. The competition has three classes and is open to anyone with a suitable 'mouse'.

In the top 'maze-solver' class, mice find the optimum route to the centre of the maze, unaided, over a series of 'runs'. This class of the competition has seen highly sophisticated robots competing in previous years. They feature four-wheel drive, steering and anti-lock brakes.

The competition's other 'wall-follower' categories, provide a route for new competitors to develop their skills. 'Non-contact' and 'contact' wall-following mice, solve the maze by following the left-hand wall, using either sensors or physical contact with the walls of the maze. Instigated in 1980, by the Institution of Electrical Engineers (IEE), Micromouse continues to stimulate wide interest in electronics and software technology. The coveted Micromouse 'cheese trophy' for 2005, was won by Derek Hall. His maze-solver mouse, Motormouse IV, achieved the fastest run-time, after a tense battle with long-time rivals, Ken Warwick and West Midlander, David Woodfield.

The 2005 competition saw Coventry University student, Andrew Lloyd, and a team from the Sharif University in Iran, awarded special newcomer prizes, for exceptional first-time entries. Teenagers Ryan Chidley and Shannon Hall, successfully defended their respective titles in the non-contact and contact wall follower categories.

For further details visit www.tic.ac.uk/micromouse or contact Dr Tony Wilcox on 0121 331 5400.

(057174-1)

New power management ICs

Microchip introduces the TC1303/TC1304 and TC1313 series of power management ICs that combine a synchronous buck switching regulator, low dropout (LDO) regulator and power-good function into single-chip solutions.

All five device options include the 500 milliamp (mA) Buck Regulator, and the 300mA LDO. The differences between the devices relate to the power-good function.

- The TC1303A power-good operates on the buck regulator output
- The TC1303B power-good operates on the LDO output
- The TC1303C power-good operates on both the buck regulator and LDO outputs
- The TC1304 power-good is used to provide a power sequencing function
- The TC1313 does not feature a power-good function

These devices provide a low-power, low-cost, minimal board space solution for a variety of battery-operated and dual output voltage systems.

The synchronous buck regulator is a very efficient switching DC-DC converter. Under heavier loads, the converter operates in PWM mode at a fixed 2.0 MHz switching frequency. For lighter

loads, such as in a standby mode, the converter switches to a power-saving pulse frequency modulation (PFM) mode. The regulator has adjustable and fixed output voltage options.

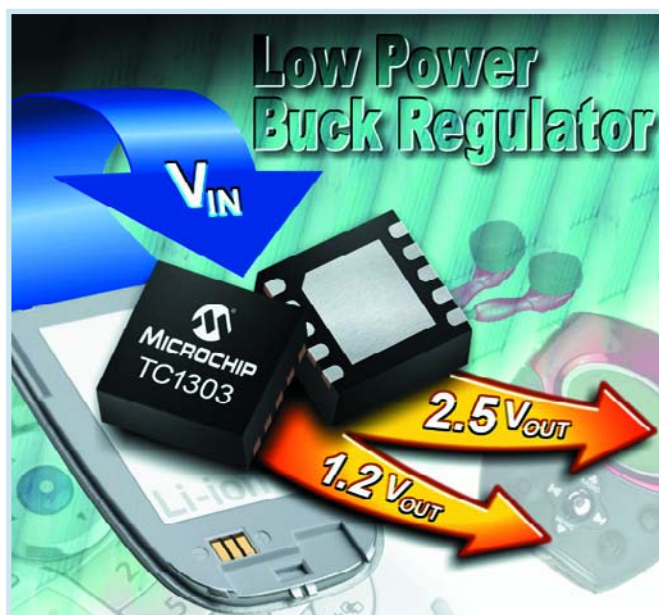
The low-noise linear LDO regulator supports up to a 300 mA load with a very low dropout voltage (137 mV at 200 mA typical), and is stable with a single, 1.0 uF ceramic capacitor on the output.

All the devices offer a number of protection features, including under-voltage lockout (UVLO), over-temperature protection and output short circuit protection.

The TC1303/TC1313 devices offer individual shutdown controls for the two regulators, while the TC1304 uses the power-good function together with the SHDN pin to provide a power-sequencing function (where, during power-up, the LDO must be operating within specification before the buck regulator is activated, and, during power-down, the buck regulator output is disabled prior to the LDO output). To help demonstrate the effectiveness of these power management solutions, Microchip is offering the TC1303BDM-DDBK1 Demonstration Board.

www.microchip.com

(057174-3)



Electrical fields can make you sick

Jan Buiting & Martin Cooke

'The writing is on the wall' applies to two major articles in *Elektor Electronics* June 2005 issue, both covering the subject of Electromog, one in terms of know-how ("Electromog: can't be that bad!") and the other, for a bit of real-life soldering ("Electromog Detector").

Here at *Elektor* we are satisfied to see not only that the June 2005 issue was our best selling single-month issue so far this year, but more importantly, that our discus-

that people can suffer sickness, muscle pains and headaches when having been exposed to electromagnetic fields! The condition known as 'electrosensitivity' is a heightened sensitivity to electromagnetic fields and the condition will be officially recognised as a physical impairment. The contents of the report have been 'leaked' prior to its official publication due in October. The HPA report will state that increasing numbers of British people are suffering from the electrosensitivity syndrome.

Stewart, chairman of the HPA and the government's adviser on mobile phone radiation matters, warned that a small proportion of the population could be harmed by exposure to electromagnetic fields, and called for careful examination of the problem.

The HPA has now reviewed all scientific literature on electrosensitivity and concluded that it is a real syndrome (the condition had previously been dismissed as psychological).

The HPA's findings, it is hoped, should lead to better treatment

sensitivity have moved to remote areas where they think electromagnetic pollution is lower.

In January the Stewart report also called for a national review of planning rules for 3G and other mobile phone masts. The review was launched by the government in April. Significantly, the Ofcom online Sitefinder (reported about in the first of the two *Elektor* articles) has seen far more frequent updates since April 2005.

The HPA report is eagerly awaited by anti-3G campaigners. Alasdair Philips, director of the campaign group

Powerwatch,

It's official (in the UK) and Elektor had the articles to match



sion of a highly controversial issue like the dangers of exposure to electrical fields was spot-on as well as properly balanced in respect of presenting the pro and con arguments of the scientists, engineers, telcos, anti-3G mast lobbyists, captains of industry and last but not least the so-called 'independent' institutes. Why?

A report to be published in October 2005 by the UK's Health Protection Agency (HPA) is to confirm that **several disorders can result from exposure to electrical fields generated by, among others, mobile phones and mobile phone base stations.**

The agency reviewed all the published reports to arrive at their conclusions and the report acknowledges (for the first time)

While the total figure is not known, thousands are believed to be affected to some extent.

The report, by the agency's radiation protection division, is expected to say that GPs have no guidelines on how to treat sufferers and that additional research is needed to find treatment and cures. It will give a full list of the symptoms, which include dizziness, irregular heartbeat and loss of memory. Although most European countries do not recognise the condition, Britain will follow Sweden where electrosensitivity was recognised as a physical impairment in 2000. About 300,000 Swedish men and women are officially recognised sufferers. The acknowledgement may fuel legal action by sufferers in the UK who claim mobile phone masts have made them ill. In January 2005, Sir William

for sufferers. In Sweden people who are affected by over exposure to electrical energy receive government support and/or subsidies to help reduce exposure in their homes and workplaces. For example, special cables are installed in sufferers' homes while electric cookers are often replaced with gas stoves. Walls, roofs, floors and windows can be covered with a thin aluminium foil to reflect the electromagnetic field back to the source.

British campaigners believe electrical devices in the home and the workplace, as well as mobile phones or phone masts emitting microwave (SHF) radiation, have created an environmental trigger for the syndrome. There is particular concern about exposure to emissions from mobile phone masts or base stations, often located near schools or hospitals. Some sufferers of electrosen-

said: "This will help the increasing number of people who tell us their GPs do not know how to treat them."

Rod Read, chairman of Electrosensitivity UK, added: "This will be the beginning of an awareness of a new form of pollution from electrical energy."

The two articles on Electromog published in *Elektor Electronics* June 2005 may be downloaded from

www.elektor-electronics.co.uk, click on 'Magazine', then June 2005.

The Elektor Electromog Detector is available as a ready-built instrument for just £50, see www.elektor-electronics.co.uk/Default.aspx?tabid=30&ProductID=2572

(057177-1)

Kaleid

Software for schematic capture and PCB design

Programs for schematic capture, board designs (PCBs) and simulation are now indispensable tools for anyone serious about electronics design. This issue of *Elektor Electronics* comes with a free DVD packed with demo, evaluation, trial and complete versions of all popular E-CAD programs. The DVD, we hope, allows you see what program is best suited to your requirements and applications. This article provides brief discussions of all programs on the DVD.

Abacom: Splan 6.0 – SprintLayout 4.0	14
AMS: Circuit Creator	14
Baas: Layo PCB 10.0	14
Bartels AutoEngineer: BAE 4.3.0.347	15
Cadence Design Systems: OrCAD V 10.0	15
CADInt PCB: CADInt PCB 2.11.15.0	15
CADSoft: Eagle 4.15	15
Capilano: DesignWorks 4.3.1	16
Edainnovation: EPD 7.2.31	16
Electronics Workbench: Design Suite 8.1	16
Free PCB: Free PCB 1.2	16
Friedrich (Bureau): Target 3001! V11 & V12	17
Holophase: CirCAD '98	17
KiCAD: KiCAD	17
Labcenter Electronics/Multipower: Proteus 6.8	17
McCAD: McCAD Schematics – MAC PCB-ST	17
Merco electronics: PCB Elegance 3.0	18
Merlin: MerliPCB	18
Micrelec: WinSchém 6.1 – WinTYPON 6.1	18
Niche Software: PCB Designer 1.3.3	18
Number One: Easy PC 9.0	18
Pad2Pad: Pad2Pad 1.6.0	18
PCB123: PCB123 V2.1	18
PCB Editor: PCB Developer's Assistant V3.0	19
Seetrax: RangerXL – Ranger2XL	19
Tsien: Boardmaker 3	19
Visionics: EdWinXP 14.0	19
Vutrax: Vutrax 12.5	19
Zuken: CADStarexpress 7.0	19

Extras: Simulators, Autorouters, Front Side Design, Video Presentations (in French), Adobe Reader, etc.

Guy Raedersdorf

oscope

U-743

**Free DVD
'Kaleidoscope'
supplied with
this magazine**

0055FEMELLE

When we first got the idea of making this overview we were convinced it would be an almost Herculean task to convince about thirty companies of the advantages of having their software product(s) appear on this month's free DVD, even if the competition in this field is, well, cutthroat. But what better way to highlight the strengths of a company than just by showing you're very much alive and kicking?

Originally we had intended to employ a small circuit (Figure 1), the 1-MHz Frequency Meter from the 2005 Double Summer Issue, using each program, followed by, if possible, a design of the corresponding board, possibly even using auto-placement and auto-routing. This hope turned out to be just a dream due to the time that would have been required to undertake each program and create the missing components one by one, finding that that they did not exist. That sure would have given the advantage to those programs for PCB design only, like Layo1PCB, because all we wanted to do was design a board in manual mode. So, we were forced to change our plan and propose instead, screendumps of certain examples that come with the various software packages. How many times have you asked yourself, where can I find a small simple program that can make a printed circuit quickly and have it etched within an hour? Thanks to the contents of this DVD, you now have plenty of choices.

Still, we did not want to completely abandon our original idea and invite you, our readers to participate in this venture:

Get going with the programs you find interesting (including the ones you may already know), using the schematic in Figure 1 as a starting point. A dedicated page is available on the Elektor Electronics website where you can report your findings with the program(s) of your choice. We also welcome feedback from professional users of the various PCB design programs.

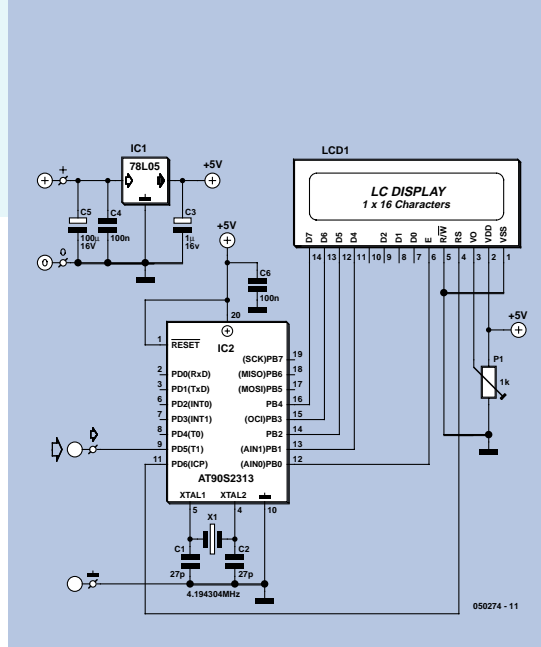


Figure 1. Here is the small schematic that we would have liked to submit to each of the schematic capture and PCB design programs on this DVD.

We do not intend to discuss each program on the DVD in great detail. Rather, we focus on problems and peculiarities encountered during installation and practical use. We also give you special tips; allowing you to form your own opinion by trying them out and help you decide for yourself which one is the most appropriate for your objectives.

Let's look at the programs on the DVD in alphabetical order, using the first letter of the company name or, if there are several programs, the name of the program itself (a number of companies having taken the name of their program or vice-versa).

SPlan 6.0 Sprint Layout 4.0

Abacom kindly sent us a CD-ROM with a demo of all versions of all of their programs, and even created a program launch interface.

Just enter the Abacom directory and click on the Start icon to launch the demo.

SPlan, version 6.0, is used to produce schematics.

The Zoom page function is used to make the entire schematic reappear on the page.

The creation of a component list only takes a few seconds and a components search is very easy and fast.

Modifying the components is simple, possibly because the possibilities are limited.

The Sprint Layout board design program will not impress those who are used to 'large' programs, such as Pads Layout or OrCAD Layout, but it is a very easy way to make one's own trace design.

Placing various pads (several pad models are available) and traces is easy, as is editing.

Since we wanted to limit ourselves to programs related to schematic capture

and board design, we transferred all of the other demos into a separate subdirectory called 'Extras'. If you would like to try something other than sPlan or Sprint Layout, all you need to do is to move the relevant file(s) from the 'Extras' directory to the corresponding main directory located above it (Abacom\Demo\).

Circuit Creator

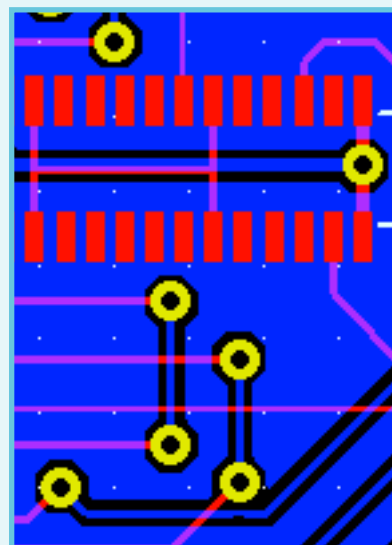
Circuit Creator is a set of programs including Board Creator, Route Creator, Symbol Creator, Logic Creator, and View Creator, plus a set of documentation, including a .pdf manual of no less than 364 pages. If the documentation is faithful, the entire design process is covered: schematic capture, creating interactive symbols, circuit simulation and automatic router design. In the schematic capture phase, the zoom is not progressive and uses several steps. There is also another, better version. Platinum, featuring a simulation option. The commands are not very sophisticated, but none the less efficient. The simulation program, SpiceCREATOR, also from AMS, is presented to you in the Extras/Simulators folder.

Layo1PCB 10.0

The program installation is the same no matter which language is selected, English, Dutch, or French. The installation is done in the wink of an eye, due to the small size of this program, hardly 2.5 MB.

Despite its simplicity of use, this program is really worthwhile. The netlist generated by a schematic capture program may be used to establish the required links. The 0.0001-mm internal resolution is useful when designing high-resolution boards.

An DRC (Design RuleChecker) is available online.





This small program is easy to test and use to successfully create one's own PCB designs.

BAE 4.3.0.347

BAE (Bartels AutoEngineer®) is a set of programs for schematic capture, PCB design and to ASIC/integrated circuit design.

There is a version for Windows and one for Linux.

Installation does not present any problems, taking just a few seconds, due to the size of the program. Once installation is finished, you will be asked if you want to go to demo mode or to schematic capture only. It is easy to pass from demo mode to schematic mode by replacing the *router.cfg* configuration file with the *schema.cfg* file (but that implies license ownership, which beyond the scope of this article).

BAE is clearly a professional program that has already evolved over several years. The structuring of a project is well done. It goes from schematics to part placement and then attempts auto-routing.

Each time, a new panel is opened within the triple-panel (screen) layout. Manipulating the components is self-evident.

Board b1 of the *sysdemo.ddb* project is very representative. The auto-router demo is impressive (260 interconnections in less than 30 s on a 1.8-GHz system —

the world has really changed since fast CPUs and large memories came about). That is a comment often heard when one sees the actual power of the auto-routers, even if we let auto-placement do its thing, the auto-router has no problem inserting the interconnections in the little space left between the components.

OrCAD 10.0 Demo

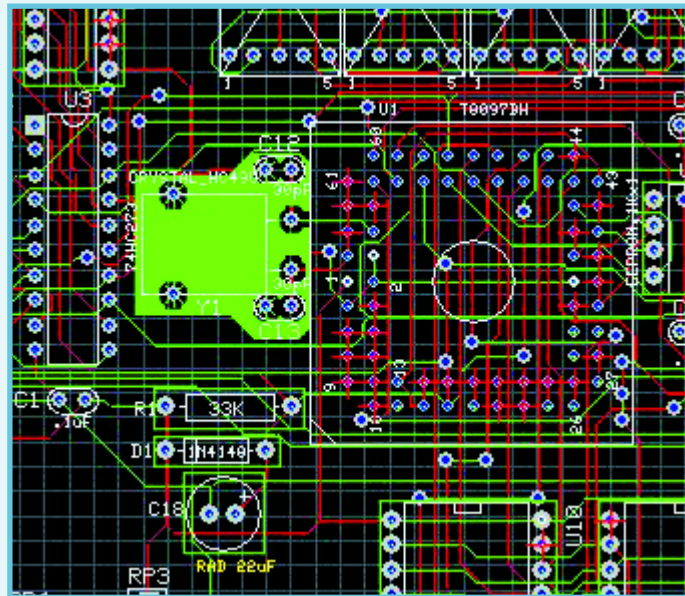
If the name of the company itself is not very meaningful to our readers, the name OrCAD has a great ring. OrCAD was one of the first schematic capture programs to appear on the market, at least twenty years ago.

At about 175 MB ORCAD is the largest suite on the *Kaleidoscope* DVD. It integrates several modules, one of which is *OrCAD Capture* for schematic capture and *OrCAD Layout* for PCB design.

One of the strengths of the OrCAD approach is sharing data over the entire process, a solution that we find on more and more programs, with retroactive editing. As soon as we have a netlist we can execute placement of components and their routing. These commands, the product of optimising utilisation experience over time, have almost become the standard in the profession.

This demo version of OrCAD 10.0 includes a demo version of *OrCAD Unison Ultra* with CIS option. It offers the following functionalities:

- OrCAD Capture, one of the most pop-



ular schematic capture tools in the world with a CIS option (Component Information System) offering options for powerful searching, capture, and analysis of manufacturer specifications of components

- PSpice A/D, offers sophisticated simulations for analog and mixed signal environments
- OrCAD Layout with SPECCTRA, PCB design, backed by the router which is the industry standard.

We extracted the file that is on the DVD. All you need to do is launch the setup. A few facts about the company and a few mouse clicks down the line, and the program is installed. Reading the documentation on the Internet may

give your firewall a hard time. You need to search for examples in the sub-directory `\tools\xxx\samples` —where the xxx will depend on what it refers to: capture, layout, etc. We'll let you savour the pleasure of discovering it yourself.

CADint PCB 2.11.15.0

This large program (close to 15 MB) from CADint (Sweden) in the 3.70.1 version installs with no problems. The demo version is similar to the others, somewhat limited, especially with respect to back-up. The possibilities for the schematic capture section are very substantial. You may be surprised to see that the various models are not drawn on the grid in such a way that

there is a progressive shifting of lines as the connection studs drift, but as soon as the grid is turned off, we no longer have this parallax problem. A number of commands are intuitive.

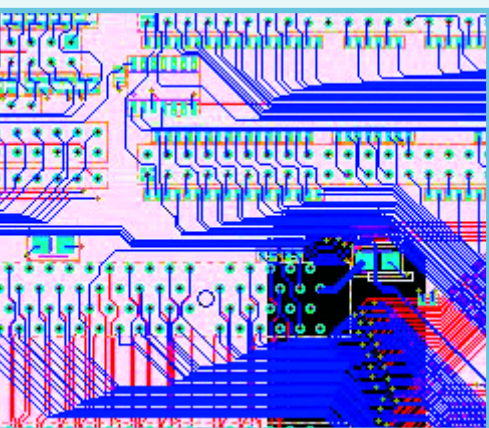
CADint PCB also has a PCB design section capable of powerful functions such as automatic via generation, manual routing with optional Bezier curves, calculation of trace length and corresponding impedance, pin and group swap. Free to all academic institutions.

Eagle 4.15

Eagle (Easily Applicable Graphical Layout Editor) from CADSoft exists in German and English; there is also a version for Mac and another which works in Linux. The current version is 4.15.

Eagle is a complete program integrating a schematic editor, a PCB editor and an auto-router — all in the same user environment. This avoids having to convert the netlists in order to get from schematics to the PCB design, whose maximum size, in the professional version, is 1.6 m square, at 0.1-micron resolution.

At first, opting for the German version shows up at the user interface level and the Help menus, but not the stan-



standard program commands, still in English. This should not bother our readers.

Eagle is a highly appreciated program all over the world due to its high-performance *Light version*.

We did not try the Mac version or the Linux version.

DesignWork Professional Version 4.3.1

The version presented on the DVD also supports simulation and VHDL.

The installation goes smoothly.

The interesting thing is that we have direct access to examples which clarify the capabilities of the program.

We also have the option of importing a certain number of third-party formats. Components are placed directly; you just need to move it inside the window and place it on the work surface.

You will need some time in order to scroll through the 1,000 components that make up the total list.

Placing the connections is extremely simple, just click on the first pin and there you go, the wire at the end of the cursor, to the destination pin.

Documentation is provided.

EPD Windows 7.2.31

This Edainnovation software has the special ability of working with an AutoCAD engine.

You will be requested to enter a license when you launch *EPD Windows* for the first time.

To do that, you will need to be online, or it may be possible by telephone.

The system automatically creates the body of the message with the serial number.

As a general rule, you will receive a response with the evaluation license file for 15 days called *EPDWINDEMO.pwd* within 24 hours. You need to transfer it to the program directory (typically `C:\Program Files\CAD DESIGN SOFTWARE\acad`) where you will give it a new name, *CDS.PWD*. When restarting the program, ignore the error message and continue.

The environment is very unusual. The graphics are surprisingly sophisticated.

The entire program seems powerful and at an industrial level, but no doubt requires an approach other than the one we are used to for this type of program.

When reducing the size of the window you are working in, the program automatically recentres the object. Fortunately, there are a good number of examples on which to practice and gain experience.

Design Suite 8.1

Who does not remember the good old Ulticap and Ultiboard from the late 80s? They have really evolved since; acquired by Electronic Workbench, a simulation specialist, it is very popular in colleges and was itself recently taken over by National Instruments. Strategic alliances are being concluded and are moving ahead at a fast pace. Electronic Workbench is proposing a demo version of their *DesignSuite 8.1* software. It integrates four programs: *Multicap* for schematic capture, *Multisim* for simulation, *Ultiboard* for PCB design, and *Ultiroute* for routing.

You are required to enter some personal data during installation before being able to continue. The program makes sure your profile is updated.

The installation proceeds smoothly. Two options open when starting the program: Multisim 8 and Ultiboard 8.

The countdown of the number of days in use begins as soon as the installation is terminated. You have 45 days.

The program automatically proceeds to verify that the program is current. It then opens automatically.

The work area is attractive. Placing the parts is simple. Select, confirm and then place. Placing the interconnections is even easier — click at the start and click at the end.

The program incorporates powerful tools like a digital oscilloscope to display signals during a simulation.

The probes provide precious information about what's happening in real time where the probe is connected (virtually).

If only for the number of examples offered, this is a superb demo.

The second program, *Ultiroute*, as one might suspect from its name, is a high performance router, and a peerless one, too.

Here at Elektor we have been asked many times which software our designers use. Well, they are Ultiboard and OrCAD fans, having used extensively customised versions of these programs for close to twenty years to design Elektor's superb PCB boards that are well-renowned in the industry.

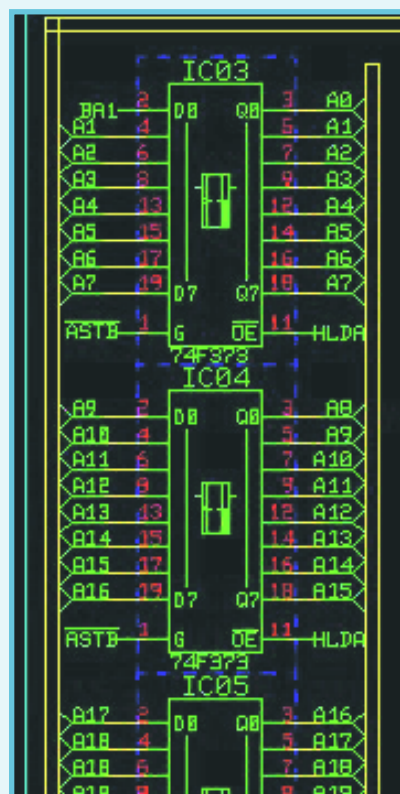
FreePCB 1.2

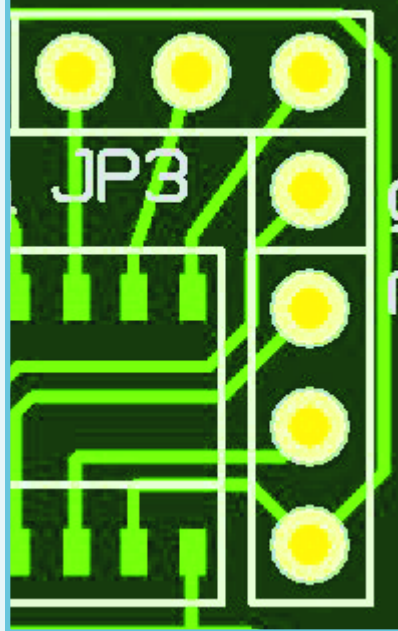
This program comes in the form of a compressed file which is self-extracting. You must extract it in an optional directory. Enter the subdirectory\bin of the directory thus created and launch the program.

As the name indicates, it is a PCB design program. It is used to import and export netlists and to use rats nests.

The manual is in .html format and easy to access.

Although it is a simple program, having function key reminders simplifies use. The digital keys are intuitively placed.





Target 3001! V 12

Here we have the exclusive honour of presenting the most recent version of this multifunction software, **version 12**. Unfortunately, we were not able to test it, the file literally arrived 1 hour before the *Kaleidoscope* DVD went into mass-production.

We are referring to version V11 which you can run by clicking on the icon *tarv11*.

Target 3001! does not contain modules. Rather, it is a program integrating schematic capture, simulation, PCB design, auto-routing, auto-placement, CEM analysis, and even front panel design.

The version presented here is a version limited to 100 pins/blocks, two layers and the simulation of 30 signals.

It should be noted that the developers of this software authorise free distribution of the demo version, it is however not allowed to use the software for commercial purposes.

Do not be alarmed by messages announcing that the software only lasts six months...

The program offers a large number of examples giving a good idea of its capabilities. It is true that the limit of 100 pins is relatively restrictive.

CirCAD '98 4.20†

This one, too, installs with no problem under Windows. *CirCAD* shows technology as it existed about ten years ago, as the original creation date indicates.

It is not simple to obtain a zoom... Once you have discovered the technique, it is rather impressive to see what could be done at the time. Remarkably, each pin has its own number, so there should be no more excuses for connection errors. Opening

a reference window lets you zoom precisely on the desired area.

Among the interesting aspects of *CirCAD* we found '*reverse engineering*' which allows you to turn a scanned image of a PCB all the way back to a schematic!

KiCAD

KiCAD is an Electronic CAD suite comprising several programs:

- *KiCAD project management* utility
- *EESchema* (Schematics)
- *PCBNEW* (Printed Circuits)
- *CVPCB* used to create the association list for PCBNEW indicating connections between components appearing on the schematic(s) and the corresponding modules (physical form) for the printed circuit.
- *GERBVIEW* (view GERBER files).

The program comes in zipped format and should be extracted into a separate directory, an operation which takes a certain amount of time, considering the size of the of the relevant files.

Next, we enter the directory that we created and start *eeschema.exe* for schematic entry. As they show on the screendump, there are some relatively complex schematics...

There is even a 3D viewing option — superb! Just move the mouse to any spot on the board surface to see it turn in space and see the components in 3D. The board's centre of gravity is identified by an axis.

Different executables exist for Linux and Windows (98, XT, NT).

The files generated by the various operating systems are perfectly compatible between one system and another.

Proteus 6.8

Proteus is a complete software tool, composed of *ISIS* for schematic capture and *ARES* for PCB design.

ISIS

The examples available are interesting, an EPROM programmer and emulator. The work surface is well thought out, all of the necessities are within reach: Components, General View, etc. Worth the detour, especially since the prices are reasonable.

ARES

You will need a bit of patience before



being able to use the PCB design program.

The quantity of subdirectories that the Labcenter Electronics directory contains is impressive. You need to go into the subdirectory\Proteus 6 Lite\SAMPLES\Schematic & PCB Layout to find all the necessary. Any new manipulation causes the (annoying) hourglass to appear.

The autorouter demo is impressive, although professional users might have some reservations that a non-specialist might not notice.

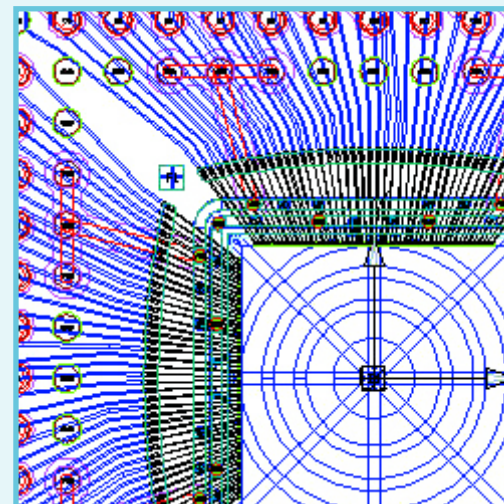
Proteus, in its complete version, can simulate an enormous number of components, LEDs and LCD displays using Atmel and Microchip controllers.

McCAD Schematics MAC PCB-ST

As you can guess from its name, a version for Mac computers exists. As far as we know, this was one of the first or the first CAD program written for the Mac. Given the growing success of Windows, a version for the PC was added later. When requesting the program installation execution, you will see two executables and one document appear. Begin by reading the latter. Install the schematic capture program, then the one for PCB design. This operation runs smoothly. Then you will find five tools, from *Schematics* to *PCB-ST*, also to include *PakEdit*, *Schematic Tools* and *Gerber 2*. We begin in 'lite' mode.

The familiarisation process for the schematic capture program is quick due to the simplicity of the commands and available functions.

The capacity of the PCB design pro-



gram is quickly exhausted. Eventually it will report that too many windows are open, and finally crash. Good documentation is sadly missing even for basic things like IC shape placement (however, a large .pdf file exists with 368 pages for schematic capture, we really feel the Mac influence there).

PCB elegance 3.0

The installation goes quickly. No example is available to get an idea of the final result. The familiarisation process requires some time but then the moves are logical. The components are gently placed on the grid.

Working on the components is simple, in the traditional way — with a few shortcuts to speed up things. Clicking on 'repaint' from time to time brings back colour to the grid.

The block diagram can be used to pass from one module to another and clarifies interconnection issues.

This is a general program that gives the impression of having the ability to build interesting things. It comes with a lot of documentation — and a lot of it is needed.

MerliPCB

Here, we find a very special program — not exactly a schematic capture program, but a design program that works under Corel Draw version 7 and up. As a number of our readers have one of the versions of this program, we included Merlin PCB Designer on the *Kaleidoscope* DVD.

WinSchem 6.1 WinTypon 6.1

This is a set of software in French for our French readers. We will not spend too much time on it.

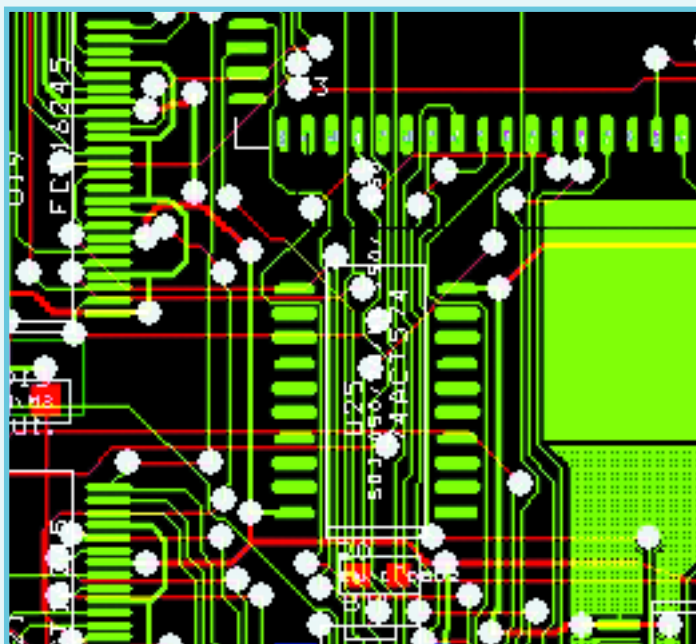
Winschem is the module for creating and preparing final copies of electronic schematics. Users are given a lot of freedom to design their schematic, the

program not imposing constraints. *Winschem* can also serve as a basis for transfer to *WinTypon*, the program used to make the printed circuit.

WinTypon is the module for drawing or making a final copy of a printed circuit drawing. Help bubbles in conjunction with keys help speed up the familiarisation process. *WinTypon* can be used as a PCB design tool, pure and simple, but it is also possible to access it from the *Winschem* schematic capture software for data transfer.

PCB Designer 1.3.3

PCB Designer is one more 'old-fashioned' program, not much bigger than 400 kB. The installation only takes a



few seconds, opening a new design takes little more time than that. Its only function, as its name indicates, is that of a PCB design program. You simply place the components as you like and then interconnect them with the traces. Please note that this works for double-sided designs. The available options are primitive but how many printed circuits were created this way? You have to opt for the widest zoom if you want to centre the pads well.

Easy-PC 9.0

Number One Systems offer you a demo version of *Easy-PC for Windows Schematic Capture & PCB Layout*, as well as a demo version, in version 2.0, of *Easy-Spice*, an A/D mixed-mode sim-

ulator that can be found in the *Extra/Simulators* directory of this DVD. It's a good idea to start with a large screen since there is no way of minimizing either the Help screen or the Control Panel (it's all or nothing). The program offers a huge number of possibilities. Several examples help you understand the program's options. The system checks to see if there is a new version.

Easy-PC also provides a Reverse Engineering function, going back from a PCB design to a schematic.

Other new features include a 3D display and a verification feature to see if sufficient space exists between two superimposed boards, a *Push-aside* function that, as its name indicates,

'pushes aside' parts so that you can squeeze in another, and *Track Necking* that automatically reduces the width of a trace when it must be connected to a thinner block.

Pad2Pad Version 1.6.0

Simple installation with no problems. Program start-up is illustrated by a nice little PCB.

This is a program for designing PCBs only. If we can believe the examples offered, this

program can handle very complex PCBs (one of the lesser examples is an SMD with 248 contacts).

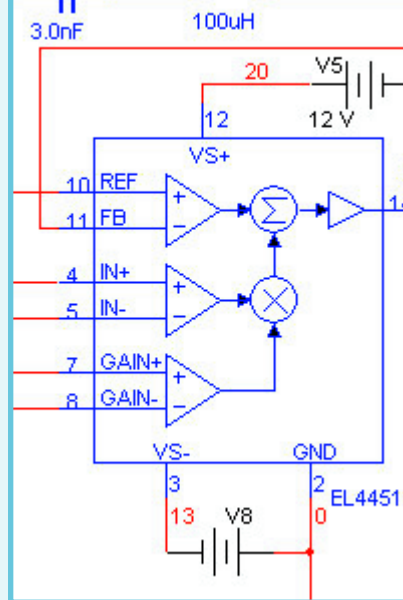
PCB123 2.1

Installing PCB123 is very easy. There are two shortcuts on the desktop,

- *PCB123 Schematic*, the schematic capture program and
- *PCB123 Layout V2*, the printed circuit design program.

PCB123 Layout offers several basic models for the PC104 and the Rabbit, in particular.

Editing is enabled with automatic activation of the component or the trace that is found just under the cursor. One click on the part makes all of the relevant information appear.



PCBEditor Version 3.0

There is a setup available to install the program. This demo version does not allow you to save files. If we can believe the About page, development was finalised four years ago in 2001.

Placing the component forms is accomplished without problems. SMD components are even placed automatically on the underside of the board.

Placing connections is easy to do. The traces have different angles, depending on the distance to be covered. An angle of 45° is used for the shortest, at 1/20th of an inch.

Documentation is found in close to 75 well-organised pages.

The professional edition can handle larger work surfaces — 32 × 32 inches instead of 15 × 15.

The system requirements are very modest, 10 MB HD space, works in Windows 95, 98, NT, 2000 but also on XP as we were able to confirm.

The PCB Editor *Help* function is well-designed and can be activated directly.

Ranger XL Ranger2 XL

Launch Autorun or, if you prefer, the setup. You will see a double installation option. You should start by reading the README file since it contains some very interesting information. If you have any installation problems, you always have the possibility of going into the corresponding directory and launching the setup of the relevant program.

Using these two programs is simple. Open a new project and search for the forms of the various components used. The most important examples cannot be edited within the limitations of

these demo versions. Nothing like having a bit of fun while familiarising oneself with a program.

BoardMaker 3

The installation is quickly completed. Libraries can only be installed once the program is installed (and that seems logical). This process overwrites the library files installed during installation of the principal program. Familiarisation with the program is smooth. It is easy to modify the size of the work surface by playing on the edges of the central panel. Moving the board design works pretty smoothly. A lot of thought has gone into tool symbol fonts.

Opening a second project in parallel with the first one causes a brutal program shutdown, forcing you to start the whole thing up again (this may be due to insufficient memory). The symbol fonts are relatively bold, but you have direct access to edit them. If you get too close to the work surface edge, the view shifts, which is surprising if not annoying.

You can open the schematic by entering directly. Several demo examples help to visualise how the program functions. A large manual is online. Impressive! Alas, no online Help.

EDWinXP 1.40

The Swedes at Visionics sent us a CD-ROM with the latest version of their product, EDWinXP, version 1.40. We put it on the DVD. The *Autorun.exe* file has some errors during program installation. A better option to use is *Setup.exe*.

If, during installation, you should get one or several messages telling you that some file could not be copied, don't panic, just click on Restart to continue. The installation process takes some time due to the size of the program but it is a visual delight.

Vutrax 12.5c

The installation is somewhat complex, using DOS windows. Installing close to 2,000 files is not restful. The program interface shows its age but doesn't interfere with utilisation. The directory of examples is well-stocked. The traces are not as fine as they might be with Windows but then a number of our readers, we know, do



not have Windows! The Help file is very large.

Using the program requires getting used to. Ctrl+Z, allowing you to undo your steps, is a useful function.

Do not bother to change the destination directory if you don't want to encounter terrible problems. Pressing <F1> during activation of an option opens an explanatory window.

Closing the program takes you back to the principal Menu from which we can choose what we would like to do and the choice is grand!

CADStarExpress7.0

CadStarExpress from Zuken requires entering your password. Installation is completed smoothly, you can even get by without entering your personal information. The trial version is limited to 300 pins and 50 components.

Launching the program automatically opens a .pdf file of about twenty pages of instructions.

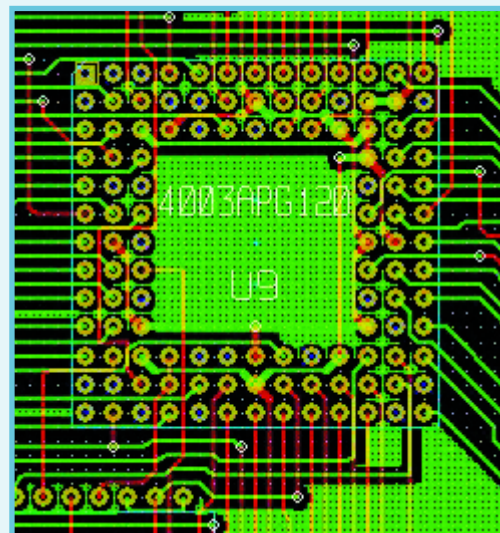
It is no doubt worth the time to print it out in order to follow it step-by-step and create your first schematic and printed circuit that can be viewed even in 3D.

The program always opens with the project last handled.

There is an impressive number of Undo's in order to re-establish a situation bungled by a series of not to clever manipulations. As in most programs, the ohm symbol is difficult to display and is replaced by an E.

Example 3, based on the 80C552-4 microcontroller is impressive. Component rotation is possible at an accuracy of 1/1000th degree!

The PCB design works well. One can see that this is a professional program.



Extras

Besides the different programs we have discussed in this article, there are others that do not fit into the defined framework of PC design & schematic capture software but seemed interesting enough to deserve to be on the DVD. We gave them their own directory: *Extras*.

Let's mention some of them here. *AsicAtoms* supplies symbols for OrCAD Capture and Pspice; *Psim6.1* libraries, a *Powersymtech* simulator, another from Number One, *HyperLynx* from Mentor Graphics, etc. and lastly, we should not forget to mention a recent version of the Acrobat Reader to enable you to read all of these wonderful .pdf files accompanying many of the programs.

The Extras directory has a very French feel, as the majority of programs we received came from that country. We leave you to discover them on your own.

A missed opportunity?

Some companies we addressed preferred to decline our offer to put their programs on the DVD, mainly for technical reasons.

Among these companies was Altium (ex-Protel). We did include them in the list of links, however for the sake of completeness. We would have liked to further garnish our Simulator section but some of the large companies like PsPice and B2Spice did simply not respond to our invitation e-mails.

Thank you's

The author would like to thank Cadence Design Systems, Inc, for allowing *Elektor Electronics* to distribute OrCAD software with this printed issue.

A hearty thank you also goes to all of the other companies, even if they did not request formal recognition. Readers will find the list in the table below, entitled 'Where to find what'.

Finally

As a matter of course, some programs will be available in a new version by the time this article and the accompanying DVD appear. Some companies put out weekly updates!

Undoubtedly the best solution if you would like to know more about any one of these programs, is to have a look at their website. Please refer to the list of Internet addresses and if you have not found the response there to all of your questions, you can send an e-mail to the address indicated on the website, to the contact closest to you geographically (preferably locally, to avoid language problems).

We cannot guarantee that all of the programs will work on all machines. We verified them on several PCs in our offices and lab with no major problems.

We, the editorial staff, obviously await your e-mails and other correspondence pro or against one or another program. Be inquisitive and, time permitting, explore the various programs gathered on this DVD. You may have a few surprises!

(050274-1)

Where to find what?

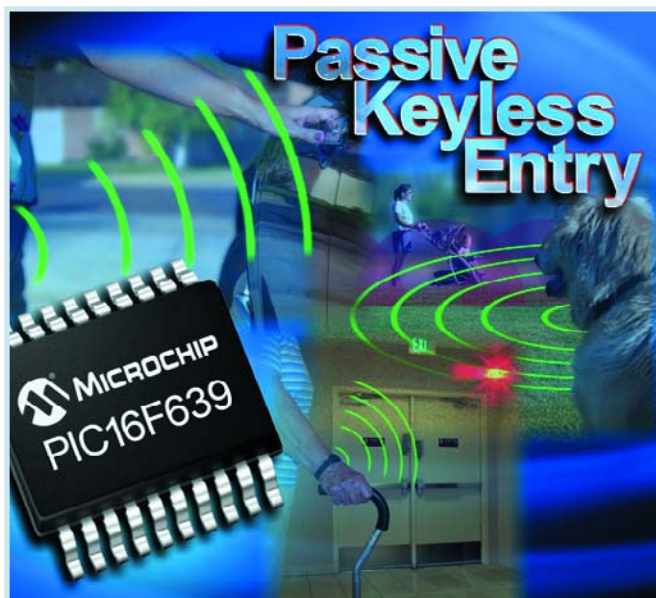
Abacom	www.abacom-online.de/	Niche Software	www.niche.co.uk/
AMS	www.advancedmsinc.com/	Number One Systems	www.numberone.com/
Asic Atoms	www.asicatoms.com/	OrCAD (Cadence)	www.orcad.com/downloads.aspx
Baas	www.baas.nl	Pad2Pad	www.pad2pad.com/
Bartels AutoEngineer	www.bme.ie/bae/bae/	PCB123	www.pcb123.com
CADint PCB	www.cadint.se/	PCB Editor	www.waldherr.com/
CADSoft	www.cadsoft.de/	Seetrax	www.seetrax.co.uk/
CirCAD	www.holophase.com/	Target 3001 !	http://ibfriedrich.com/
DesignWorks	www.capilano.com/	Tsien	www.tsien.info/
Edainnovation	www.edainnovation.com/	Visionix	www.visionics.a.se/
Electronic Workbench Europe	www.electronicworkbench.com/	Vutrax	www.vutrax.co.uk
Friedrich (Bureau)	www.ibfriedrich.com/	Zuken	www.zuken.com
Free PCB	www.freepcb.com/		
KiCAD	http://iut-tice.ujf-grenoble.fr/kicad/	Other interesting links for companies that do not appear on the DVD.	
Labcenter Electronics	www.labcenter.co.uk/	Altium	www.altium.com/pcad/
Layo1PCB	www.baas.nl	Ariadne	www.cad-ul.de/html/ariadne-start.htm
McCAD	www.mccad.com/	Circuit Maker	www.circuitmaker.com/
Mentor Graphics	www.mentor.com/products/pcb/pads/	Pulsonix	www.pulsonix.com/index.asp
Merco electronics	http://www.merco.nl/	The PCB Designer	www.cad-design.com/
Merlin PCB	www.geocities.com/MerliPCB/		
Micrelec	www.micrelec.fr/lyctech/tele.htm		

PIC micro for passive keyless entry

Microchip have announced a Flash, 8-bit PIC microcontroller with a 3-channel, low frequency (LF) transponder analog front end and an integrated KEELOQ cryptographic peripheral. This is combined with Microchip's low power nanoWatt Technology in a small 20-pin SSOP package.

The PIC16F639 is ideal for a wide range of security and other general-purpose applications. These include automotive Passive Keyless Entry (PKE) systems and Tyre Pressure Monitoring (TPM) systems, property and identity authentication systems, and security systems such as electronic locks, remote security sensors, etc.

The three-channel transponder analog front end provides bi-directional LF communication, as commonly used in PKE and RFID applications. KEELOQ technology is based on a proprietary, non-linear encryption algorithm that creates a unique transmis-



sion on every use, rendering code capture and resend schemes useless.

The PIC16F639 microcontroller also includes:

- Precision 8 MHz internal oscillator with software clock switching

- 3.5 Kbytes of Flash program memory, 128 bytes of RAM and 256 bytes of EEPROM
- Two analogue comparators
- One 8-bit Timer, one 16-bit Timer
- Ultra Low Power Wakeup (ULPW) and Enhanced Watchdog Timer (EWDt)

- Software-selectable Brownout Reset (BOR) and Programmable Low Voltage Detect (PLVD)
 - Wakeup Reset (WUR) feature
- The integrated LF analog front end includes:
- Three channels for omni-directional 125 kHz wireless communications
 - 3 mV input sensitivity
 - Programmable antenna tuning
 - Support for battery-less operation

The PIC16F639 is supported by Microchip's development tools, including: the MPLAB® Integrated Development Environment (IDE), MPLAB ICE 2000 in-circuit emulator, MPLAB PM3 universal device programmer, PICSTART® Plus low-cost development system, MPLAB ICD 2 in-circuit debugger/programmer and the new PICkit™ 2 Starter. In addition, a complete PIC16F639-based Passive Keyless Entry Reference Design (part # APGRD001) is available.

www.microchip.com/keeloq

(057174-2)

USB has never been so easy...

The AVIT Research USB to TTL Serial Cable is designed to allow simple rapid connection of microprocessors to the USB interface. It allows legacy products using PC serial ports to be easily upgraded to USB. The cable contains a USB to serial converter chip and is terminated in a standard 0.1" pitch header. The serial data is at standard TTL levels allowing it to interface directly to the micro-processor of the intended application.



Main features of the product:

- Simple and easy way to give USB support to your designs.
- Connects directly to microcontroller UART or I/O pins
- PC communicates as a standard serial device.
- All circuitry integrated into sleek, moulded USB plug.

- Allows devices that previously used a MAX232 type level shifter to connect to a PC through a USB connector and appear as a standard COM port.
- Compatible with legacy software.
- Cheap and simple way of

- upgrading existing products.
- 4-way connector (Gnd, RX, TX and +5 V).
- Supports data rates up to 921.6 kbaud allowing a significant performance increase over a standard serial port.
- Allows devices to be powered from the USB port.

Driver software for the PC is provided and is compatible with Windows 98se, 2000, ME and XP. The drivers cause the USB device to appear as a standard serial port capable of baud rates up to 921600. This allows all windows programs to access the serial device exactly as if it were a built-in serial port.

The USB to TTL Serial Cable also provides a 5-volt supply to the target system which can supply up to 50 mA. This allows many embedded applications to be run without a separate power supply. The output is fully protected against short circuits.

Avit Research,
<http://www.avitresearch.co.uk>

(057146-7)

Bluetooth product developed in three months using innovative tool



A wireless telephony access point for home use — a product that is one of the most sophisticated deeply-embedded Bluetooth systems ever developed — has been created in record time with the aid of a new development tool from Cambridge Consultants. Designed in just three months by Clipcomm, for Korea Telecom's 'OnePhone' fixed-mobile-convergence (FMC) service, the new BlueStation series uses Bluetooth to allow mobile phone users to make or receive calls over the fixed wireline phone network when at home. This facility allows mobile phone users to use the same phone and number wherever they are, and reduce call charges. With Bluetooth becoming commonplace on mobile phones, this telephony concept is attracting worldwide attention, and a similar scheme called BT Fusion is currently being launched in the UK by BT. Conventionally, sophisticated Bluetooth products employ a Blue-

tooth chip operating under the control of another processor, which manages the overall system and the user interface. However, the recent introduction of Cambridge Consultants' xIDE for Interface Express tools gives users unrestricted access to the full native power of the XAP proces-

sor core inside CSR's market-leading Bluetooth device, BlueCore. This allowed Clipcomm to develop a single chip solution that performs both Bluetooth and system control functions, saving the cost and real estate required for a separate processor and memory - an important consideration for a cost-sensitive consumer product like this. Clipcomm's new access point

product is an embedded system handling up to four handsets and two public network lines. It also has to support multiple instances of CTP, HSP, GAP and SDP Bluetooth Profiles (cordless telephony, headset, generic access and service discovery respectively). In addition,

which allows it to perform line echo cancellation as well - providing a very high performance solution. With the aid of xIDE for Interface Express, the complete embedded system was developed in just three months. Clipcomm purchased three licenses for the software development kit (SDK) which provides a complete compile/make/debug toolchain for the XAP processor, plus a library of qualified Bluetooth Profiles. Another major aspect of the tool-suite is its integrated development environment (IDE), which simplifies software projects through a highly-graphical software project management environment and user interface.

... the same mobile phone and number wherever you are ...

product is an embedded system handling up to four handsets and two public network lines. It also has to support multiple instances of CTP, HSP, GAP and SDP Bluetooth Profiles (cordless telephony, headset, generic access and service discovery respectively). In addition,

which allows it to perform line echo cancellation as well - providing a very high performance solution. With the aid of xIDE for Interface Express, the complete embedded system was developed in just three months. Clipcomm purchased three licenses for the software development kit (SDK) which provides a complete compile/make/debug toolchain for the XAP processor, plus a library of qualified Bluetooth Profiles. Another major aspect of the tool-suite is its integrated development environment (IDE), which simplifies software projects through a highly-graphical software project management environment and user interface.

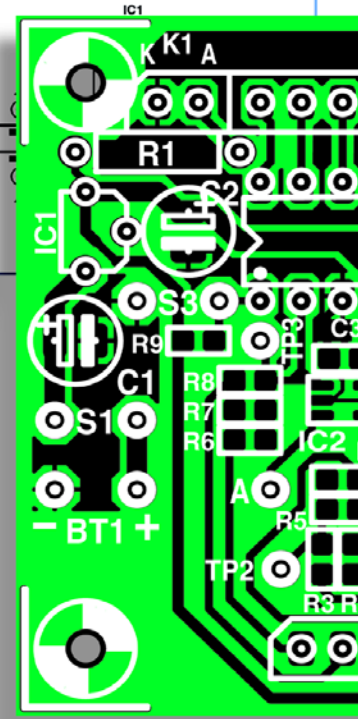
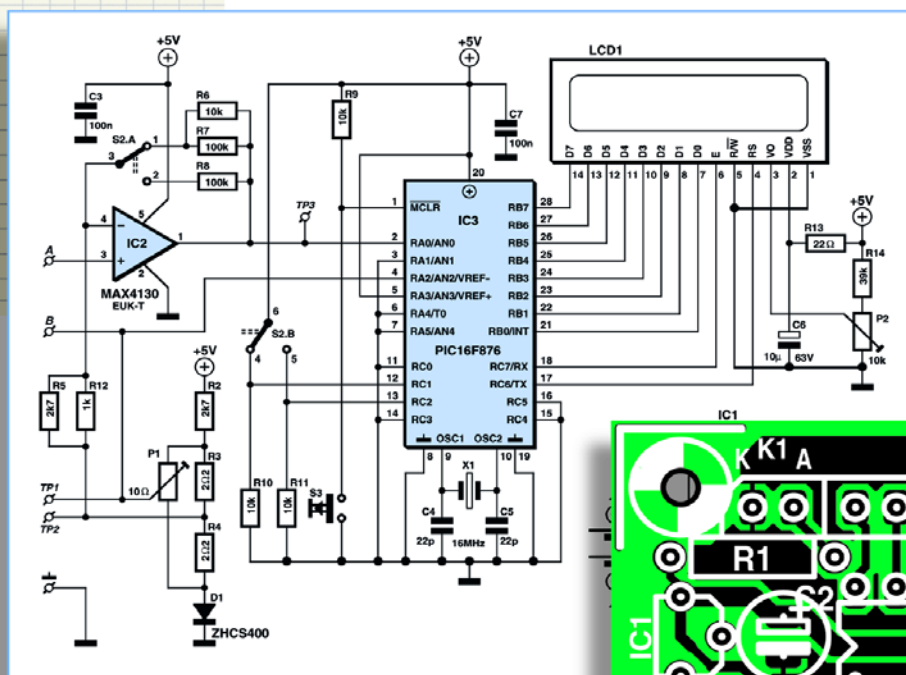
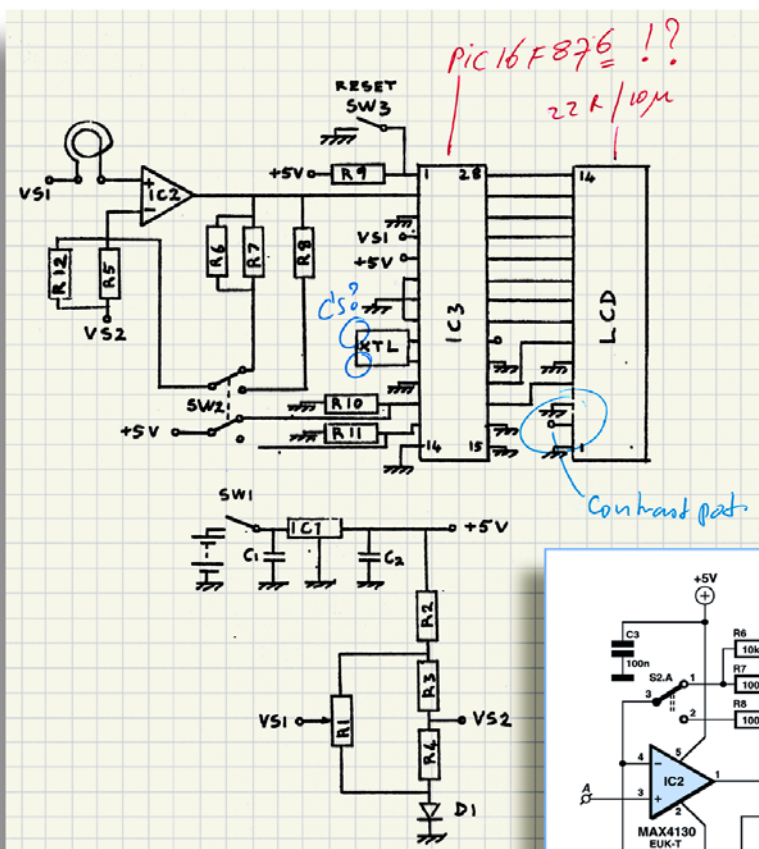
**Cambridge Consultants Ltd,
Science Park, Milton Road,
Cambridge, CB4 0DW, UK.
Tel: +44 (0)1223 420024;
Fax: +44 (0)1223 423373;
www.cambridgeconsultants.com**

(057146-4)

From Concept to

The champion's long road

Christian Tavernier



Whether building constructions that you find every month in **Elektor Electronics**, coming up with new electronic designs in a research lab, or making your own application from scratch, the steps to follow are the same. Until a few years ago this was an entirely manual process, but things are different today and many software options are available to assist the electronics designer in this task.

Reality

Some are very expensive and prices can be out of reach for hobbyists or for small businesses, but many others are proposed at a very affordable price or may even be available for free.

This article will give you an overview of the topic while describing, step by step, the process from an application idea to building a printed circuit, showing and explaining each solution available, with its advantages, as well as its disadvantages.

A problem well stated

is a problem half solved or so goes the saying. The first step, not to be taken lightly, consists simply in defining one's needs as precisely as possible. For example, if you want to make an amplifier, you must specify impedance values and input and output levels, the source voltage, the operating frequency range, etc.

In this area, no software can help you. However, consulting manufacturer catalogues of finished products, careful reading of your past Elektor issues, or consulting the Internet may all be of precious help by showing you "what is typical" in the field relating to your

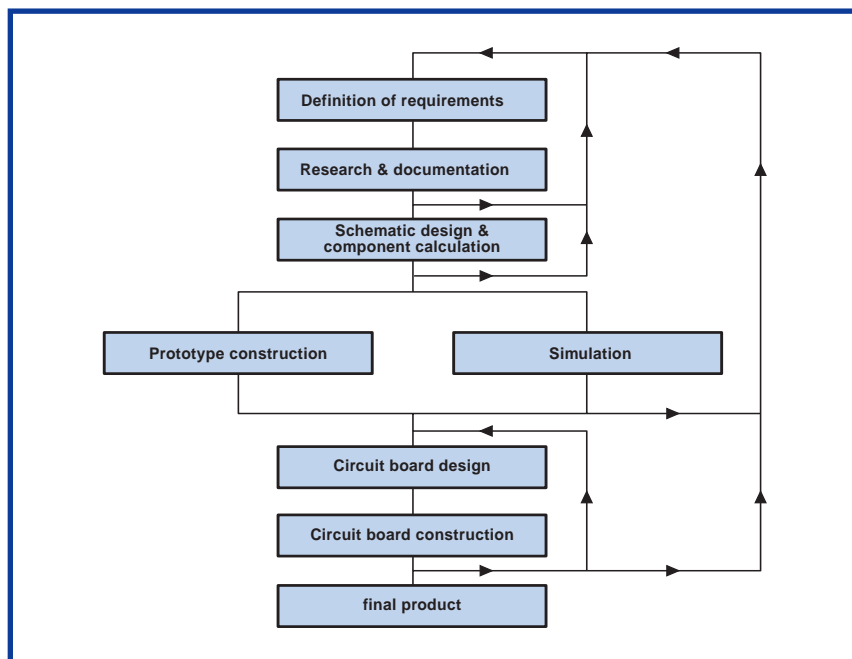
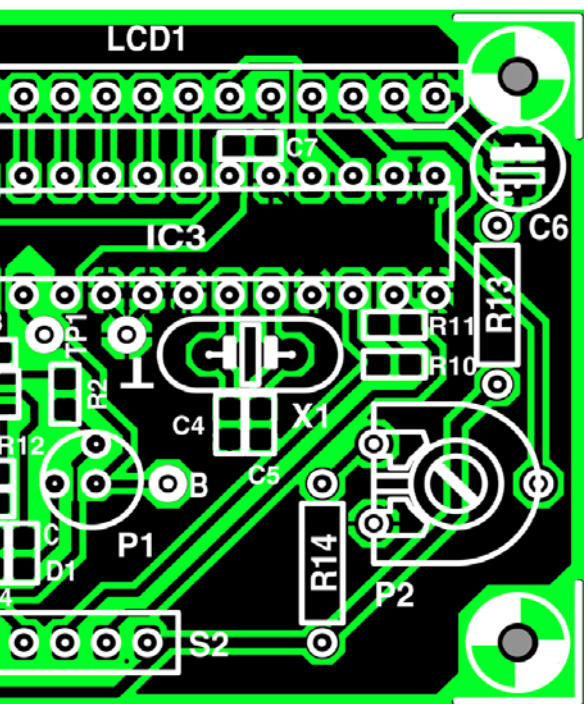


Figure 1. The flowchart shows that the path from idea to finished product involves more steps, actions and decisions than we might have thought at first.

project. Unless you have specific constraints, or because your application needs to be integrated into a larger unit which has its own requirements, it will still be possible to modify these specifications once your work has advanced.

That is one of the specialties of electronic design: be prepared for many 'feedback loops' or loopbacks. They should be part of the design, as shown in the flowchart in **Figure 1**. In the example of our amplifier, you may discover that the source voltage you initially selected is too weak for the output power you need. So, you should increase it, which means you then have to review your choice of integrated circuits or transistors because they may not be able to withstand this new voltage. It's like going in circles, but that's part of electronic design.

The Internet is the first place to look

Even if it's amusing for all electronics engineers to design schematics from scratch, using existing designs for

inspiration, even if only as a base to improve on or to avoid reinventing the wheel. In this field, the Internet provides endless inspiration, thanks to two different sources of information: the first, the websites of semiconductor manufacturers. You will find all of the technical sheets on their products, of course, but also many application notes which, even if they do not resolve your problem, can give you tips or a starting point. All of this information may be downloaded for free and is available in Adobe pdf format, which all PCs can read, thanks to Acrobat Reader.

Secondly, websites of electronics hobbyists who often give you access to the fruit of their work. With one search on Google (www.google.com) you can quickly see that which may interest you, but you should know that there are websites which have set themselves up as true encyclopaedias of schematics. Without a doubt, the most well-known is **Discovercircuits.com** [1], a compilation of more than 11,000 circuits classified into 500 categories. In addition, we can also cite **Sam Electronic Circuits** [2], a Greek website

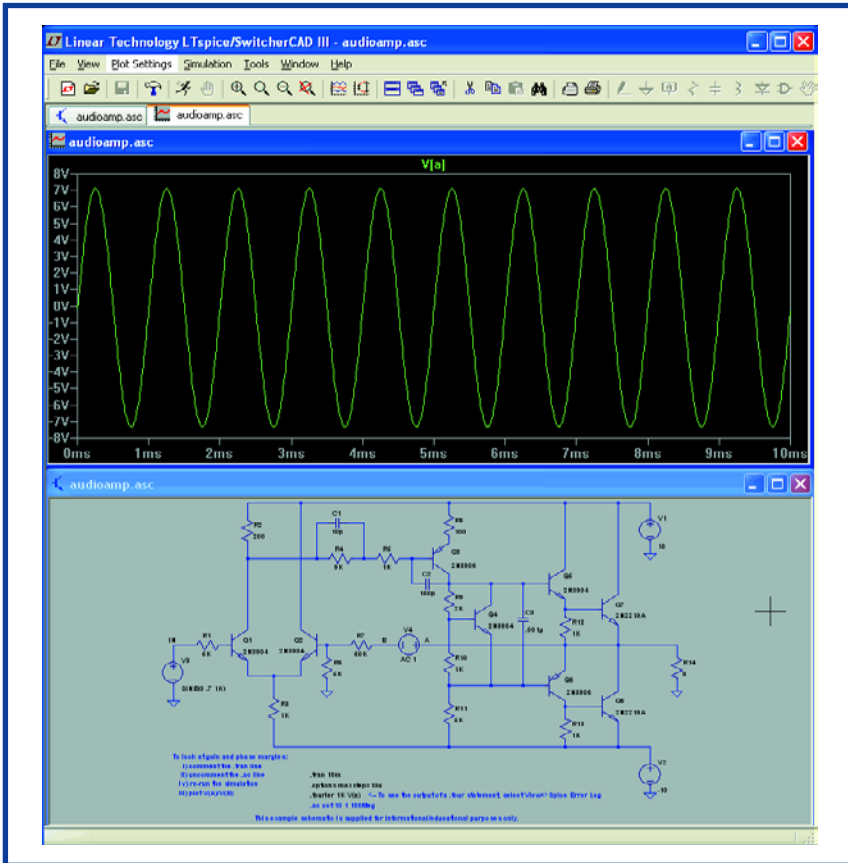


Figure 2. Some manufacturers make high-performance tools available, like LT with their Switchercad.

with a comprehensive list that is also available in English. As in the previous case, these websites may not provide you with 'ready-made'

schematics but they can certainly offer you a starting point. Be aware that although the manufacturers' application notes are usually reliable, written

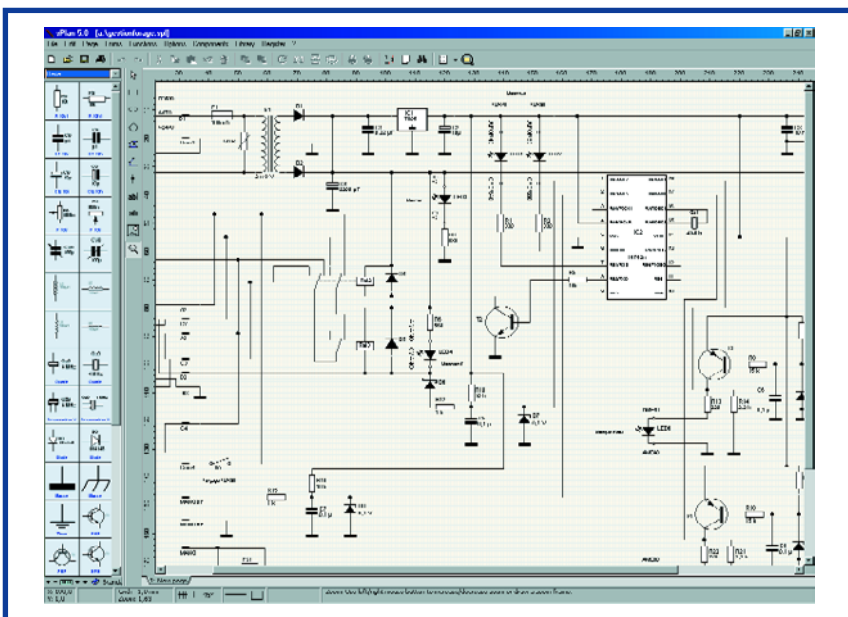


Figure 3. Due to the high resolution of our screens and the capabilities of the latest operating systems, schematics have become real masterpieces.

by developers in their own laboratories, the same is not always true for the quality of schematics on hobbyist websites. Sometimes the schematics are poorly designed and may not be reproducible, or it is possible that errors were unintentionally inserted during website creation. If a functioning e-mail address is available, the webmaster may be able to help.

Of course, the majority of this information is in English and those who can read this very article have somewhat of an advantage in this field, but it's all relative: Internet sites are much easier to read than Shakespeare's Macbeth!

Calculation and design software

Whether you found a part of the schematic or no schematic at all in the previous step, you will have to design the complete schematic for your application. That is obviously where the most difficult part of the operation lies because you absolutely must have a minimum of knowledge in electronics to be able to do that, and no software can replace what one does not possess, even if, as we shall see in an instant, various products can assist you in the most delicate phases. If you know nothing about operational amplifiers, you will not know how to assemble them in order to create a specific function and no software can do it in your place.

In that case, even if you are an experienced electronics engineer, you may be lacking in certain areas of competencies that are so specialised or delicate that you hesitate taking that risk. However, numerous products can help you with that, but you must find them, which is not always an easy thing. The websites of the major semiconductor manufacturers (yes, them again!) do provide resources that may still be unknown to many electronics engineers.

So, for example, if you are required to design a switching power supply, try the excellent **Switchercad** by Linear Technology that will not only help you design your switching power supply but will allow you to simulate it as well, in addition to a multitude of other circuits (see **Figure 2**) — we will revert to that below.

We should also mention **Simple Switcher**, by NS this time, also intended for switching power supplies but based on SIMPLE SWITCHER® regulators from this manufacturer. It is not as complete as Switchercad but much simpler to use.

If you are looking for all sorts of filters, you can use **FilterLab** from Microchip with which you can calculate and simulate a multitude of filters in cutoff frequencies and transfer functions defined by you. Or, if you prefer, you can always use **Filtercad**, from Linear Technology, clearly a very generous company that offers more in the way of available functions at the cost of a more sensitive familiarisation process. We should not forget to mention **Noise**, a little-known product from (again) Linear Technology which, as indicated by its name, is used to calculate the noise produced by an amplifier based on LT circuits and to select the one which is best adapted to your constraints. Finally, you should be aware that if you use any particular integrated circuit, a program may exist that is entirely devoted to calculating the components that need to be used in conjunction with it. This is the case, for example, with the MAX274 from Maxim and its filter design software or with the MAX5003 and its isolated low-power design software.

Along with these very professional products, a large number of 'small' programs are available, such as the simple R-C filter calculation program or the one which indicates which parts to use with a good old 555 chip in order to generate a particular signal. The Internet is there to help you, but since these programs are not always easy to find, use the *Elektor Freeware and Shareware 2000 CD-ROM* for PCB design and schematic capture software, and the **DVD** accompanying this issue that will put the large majority of them at your fingertips for a very modest sum (or even for free).

Schematic capture or pencil and paper?

It's done! The calculation of your schematics has been completed, but it still looks like a mess scribbled on graph paper. Before going to the next step, we need to make a clean copy using schematic capture software. You have several choices, depending on how you would like to proceed afterwards.

If you plan on testing your circuit using a prototype, wired on prototyping board or veroboard, for example, you can choose any schematic capture software, whether or not it is associated with printed circuit design software.

As far as we are concerned, we use the excellent product from the German firm Abacom which is called **sPlan**. This inexpensive software is available all over Europe in German, English, and in French. It was written by an electronics engineer for his peers and therefore the familiarisation process is very easy and accomplished in less than an hour. Moreover, the resulting schematics look as good (or almost) as the ones you see in *Elektor Electronics* (**Figure 3**).

If you plan to design your printed circuit layout automatically or semi-automatically, you need to select software called electronic CAD, which includes a schematic capture module used in conjunction with a printed circuit design module. Otherwise, you will have to verify that the schematic capture software is capable of exporting a 'netlist' (in other words, a wire list) and that the printed circuit design software is capable of importing a 'netlist' in the same format.

If you do not plan to wire the model but prefer to test your software by simulation, you must choose a schematic capture program created for that purpose. Considering what's currently available on the market, that means you must capture the schematic with the module that's integrated into the simulator. Even though certain simulators have impressive capabilities, their schematic capture is often a real nightmare.

What can we expect from simulation?

We discussed this with many hobbyists who had never tried a simulator before or had never seen any demonstrations at shows, and we noticed that almost all of them made the same mistake: believing that a simulator, if you paid the right price, can simulate everything and that they can skip the prototyping stage. Let's put it very clearly, at the risk of upsetting some simulator publishers: that is false! Of course, a simulator is useful and may sometimes make prototypes superfluous, but that is not always the case. Therefore, selecting a simulator is a complex problem; the best simulators are actually very expensive products (setting you back several hundred to several thousand pounds) that even a small business may not be able to amortize.

Fortunately, there are some solutions, albeit incomplete, that can still be use-

ful if you do not want to or cannot invest. For both of these cases, here are some criteria indicators for choosing a simulator.

Nowadays, a good simulator should be a mixed-mode simulator, analogue and digital.

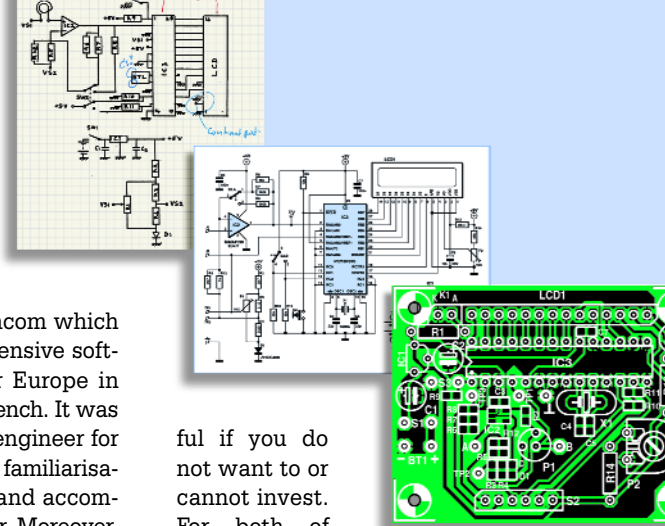
Increasingly, these technologies are combined into one application and the simulator must therefore be able to support all of them.

In order to be able to use it with a maximum number of components, the model library must be as wide-ranging as possible and it definitely must be extensible. Find out its contents before purchasing, but don't be fooled: no simulator on the market contains the latest integrated circuit in its library (if only because time is needed to integrate the model into the software), nor will it contain that rarity you found at an obscure manufacturer's site, unknown to the simulation software publisher.

Moreover, if your application integrates one or more programmable logic devices (PAL, GAL, EPLD) or worse, a microcontroller, things may become seriously complicated because this is the area in which simulators have the most problems. Some of them, such as the mighty **Proteus VSM** from Labcenter Electronics, may even simulate a certain number of programmable logic devices and microcontrollers, but this will obviously impact the purchase price (a software module will be charged for every additional microcontroller family that exists).

Our viewpoint may seem discouraging. We reap no pleasure from being negative, we just want to be realistic. Therefore, we must consider that the simulator will let you simulate parts of your application but rarely in its entirety. Except, of course, if it is very simple, but, in that case, simulation may not be useful. Still, if we had to give you some guidance, we would recommend two products.

First, **Electronics Workbench**, also known under the name of **Multisim**, is a high-performance product intended for both professional and educational uses



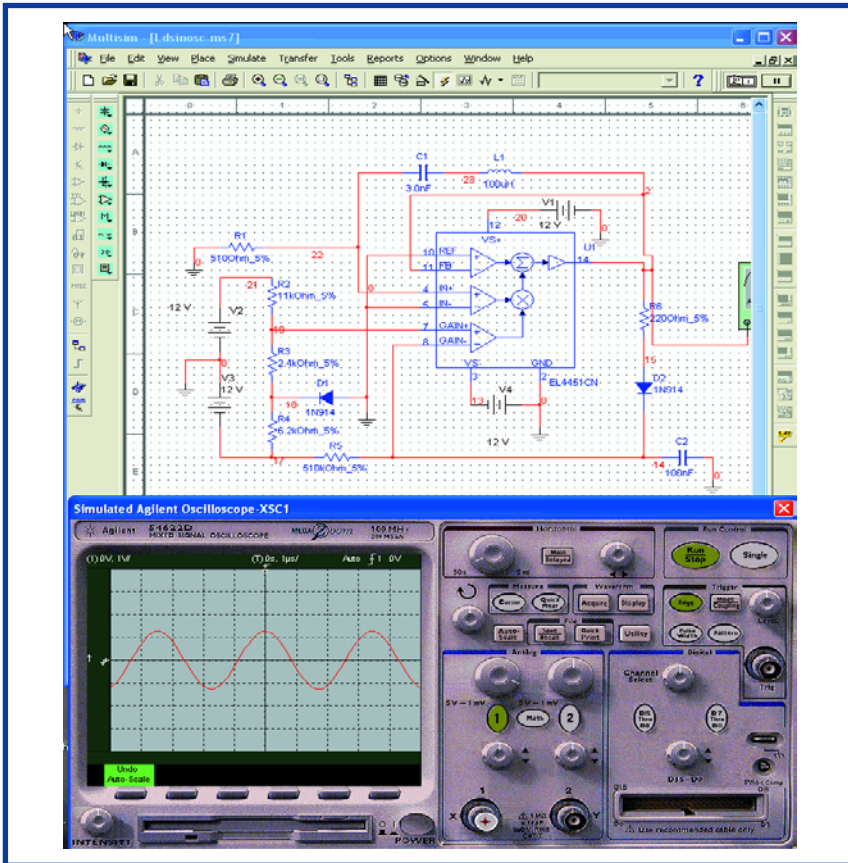


Figure 4. The (virtual) oscilloscopes included in the current simulation software are just as good as their “real” equivalents.

and has great virtual instruments on which we can view the signals just like in the real world (Figure 4). These simulations are reliable, even for unstable

circuits or for spontaneous oscillators. Secondly, **Proteus VSM**, one of the few which can simulate microcontrollers and also has a very ‘visual’ display

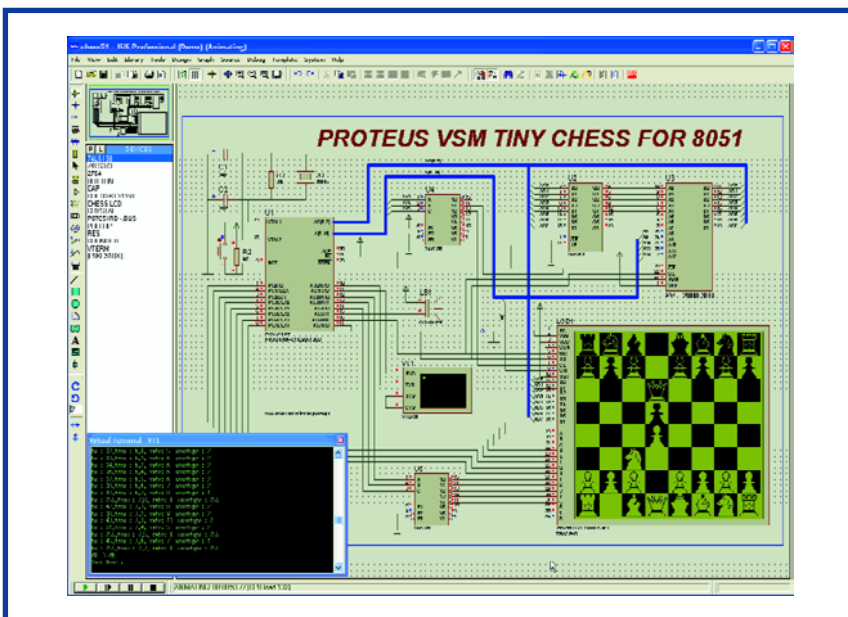


Figure 5. The power of some of the most recent simulators can be observed in this figure. Here, Proteus VSM simulates a chess game, based on a 8051 microprocessor.

mode which really brings the simulation alive on screen. At a more affordable price than **Multisim**, it is still rather capricious and sometimes ‘crashes’ for no reason while simulating ordinary circuits, though it perfectly simulated an electronic game of chess, based on a 8051 microprocessor with graphical output, which gives us a good idea of its calculating power (Figure 5).

Both of these programs have demo versions that you will also find on the free DVD accompanying this issue. In some cases they have limited trial periods and/or limitations in terms of functionalities, but you can get an idea of their capabilities. Depending on your specific example, if it is not too complex, you might even be able to simulate your application.

If your budget is too tight, you can still benefit from simulation thanks to a certain number of integrated circuit manufacturers who, (thanks again!) are making these very respectable products available to you for free. In particular, **Switchercad** from Linear Technology, which we have already mentioned, not only simulates switching power supplies but actually has a true mixed-mode simulator, useful for many schematics.

But the most spectacular of the products remains **Webench** from National Semiconductor. This tool, which can only be used online from the NS site (a fast Internet connection is recommended), covers the complete design phase of one electronic application from the input of your initial constraints up to the final construction which includes calculation, schematic design, simulation, and design of the printed circuit. Even a thermal simulation module is planned to evaluate the performance of the radiators and other coolers. Of course, this product only recognises NS integrated circuits, but this manufacturer’s catalogue is so vast that you just might find exactly what you need. In any case, go to the site and try it. It’s free and it’s really worth the detour!

Spice or no spice?

Before closing this section related to simulation, please be aware that the majority of today’s simulators are based on a Spice kernel which is ... free! Yes, Spice was developed a long time ago at UC-Berkeley and was graciously made available to the international community (see our address

book [4]). You can theoretically use the Spice simulation function without spending a dime.

But, and there is always a but, Spice is a program that only works with text files. So, your schematic must be described in the form of a 'netlist' (yes, again) in order to be accepted as input to Spice, and the stimuli that you want to apply to it must be described in the same way. In return, Spice will supply you with a ... text file, all you need to do is decode it. We do make it to the finish line, as this user did, but it is not easy or pleasant.

Commercial simulators that depending on a Spice kernel wrap this kernel with a graphics pre-processor in order to capture a real schematic and take care of translating it into Spice language; in addition, it also has a graphics post-processor that can transform Spice's boring tables of numbers into attractive curves. These two processors explain why this product, originally free, but clumsy, is now much more user-friendly, but sadly no longer free. The fact that practically all of the simulators are compatible with Spice presents quite an advantage: the possibility of easily adding new component models to the simulator. Actually, all large semiconductor manufacturers supply Spice models with their components at some point, all that is left to do is integrate them into the simulator to increase the size of the simulation library. These Spice models are ... text files, of course, but it is rather easy to integrate them into the simulation software.

Printed circuit design

Whether you wired a model or your construction was simulated, you succeeded in fine-tuning it, even if you had to return more than once to the starting line and fix your schematic, or your initial performance choices, or both. Now, your last step is to design the printed circuit.

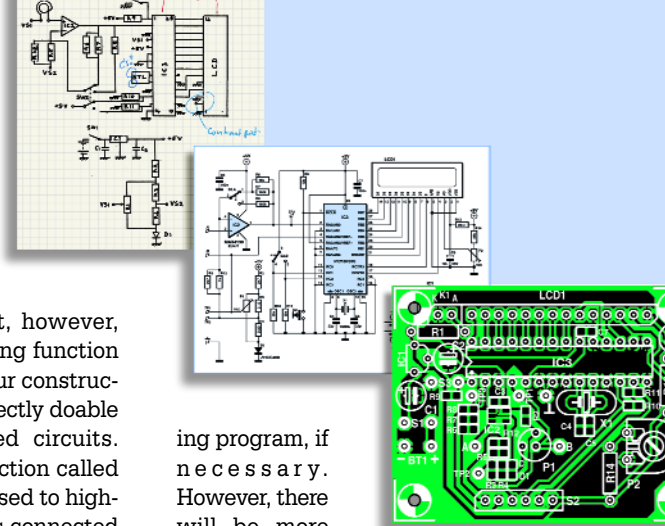
Here, again, two or three different approaches are possible but your choices may be limited, depending on your previous options. Let's look a bit closer at this.

If you have captured your schematic using independent software, such as **sPlan (Figure 6)** discussed earlier, or if your schematic is still at the paper stage; you may choose any printed circuit design software. This program will most often be autonomous software, such as the excellent **Sprint Layout**,

also supplied by Abacom and available in three languages all over Europe. You will not, however, benefit from the auto-routing function and will have to design your construction by hand, which is perfectly doable for single or double-sided circuits. Some programs offer a function called assisted routing which is used to highlight all of the components connected to the same trace helping to detect unwanted or lost connections, or to automatically track the traces, but only one by one. This solution has the advantage of being very economic and available to all hobbyists and small businesses.

If you have used schematic capture software included in a CAD package (**Eagle, Orcad, Altium** — formerly Protel — etc.), transferring the schematic to the routing software is practically automatic and you can immediately begin to design the printed circuit. Most often you will choose to place your parts 'by hand' on the construction according to your preferences and the software takes care of routing all of that, or at least it attempts to do it, we will come back to that subject. This solution is obviously much more costly than the previous one because electronic CAD programs cost a few hundred to several thousands of euros. Please note, however, CadSoft's excellent initiative, the **Eagle** editor which, proposes a 'light' version of its CAD software for free (for non-commercial use). The functionalities are identical to those of the complete version but the size of the printed circuit is limited to 10 cm by 8 cm, surely sufficient for building some interesting projects.

If you have chosen to simulate your construction, you must have used the schematic editor that was provided with the simulator to capture your schematic. In order to avoid having to re-capture it in another program, all simulators worthy of that name are associated with printed circuit design software (**Ultiboard** for Multisim or **Ares** for Proteus VSM). You will then find yourself in the previous case. If, extraordinarily, the routing software provided with the simulator is not what you need, be aware that it is always possible to export a netlist from one simulator (since that is exactly what the schematic capture software of the simulator does) that you can then import into another rout-



ing program, if necessary. However, there will be more manipulations than if it had been integrated.

The myth of auto-routing

During the selection of an electronic CAD program or a printed circuit design program, many people fixate on the auto-routing function, so much so as to make it the principal criteria for choice. If you are building multi-layer circuits, it is obvious that you should choose a program with a high-performance auto-router, but, for the majority of projects that interest us and which are either single or double-sided, the auto-router is much less useful, possibly even problematic.

Starting from the schematic (more exactly from one's netlist), the auto-router seeks to automatically draw the printed circuit traces in your stead. That is obviously very relaxing for the user but several errors may result.

First, the majority of auto-routers do not excel, or, to put it more bluntly, perform poorly in single-sided mode because they are intended to work, by default, at a minimum, in double-sided mode so that they can easily resolve problems of trace crossing.

So, if you are building a single-sided circuit, which is easier to make on your own than a double-sided circuit, forget the auto-router.

Also, the auto-router cannot take into account certain technical constraints of your project. In the field of radio frequencies (RF), for example, it might place traces side by side which will interact and cause oscillations or other parasitic phenomena.

Finally, if you want it to work correctly, an auto-router takes a long time to parametrise: minimum and maximum trace or pad sizes, minimum size of insulating areas between traces, the many parameters you must provide it before beginning any auto-routing work.

In summary, for simple or double-sided circuits of reasonable complexity, such as those a hobbyist or a small business might build, the auto-router is not indispensable and unnecessary.

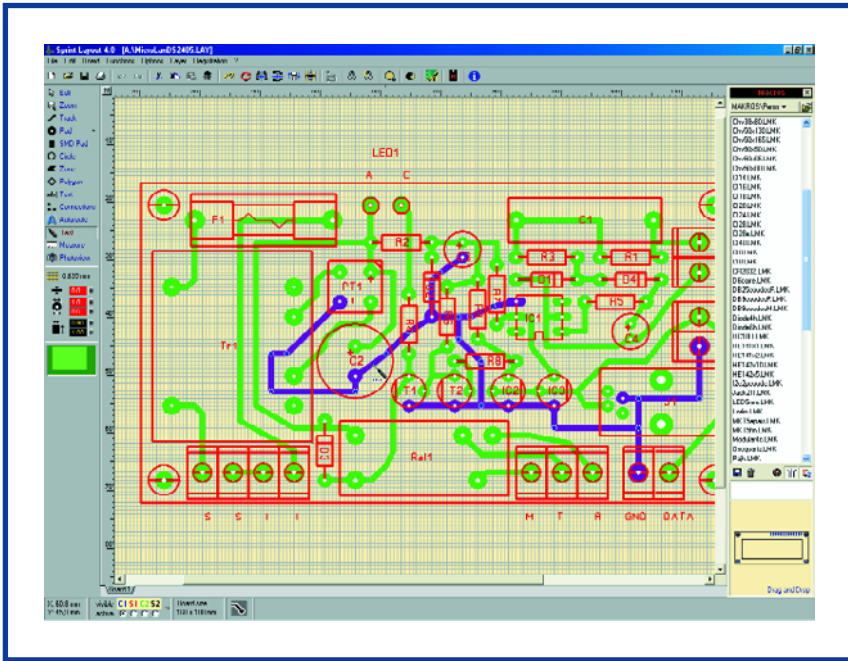


Figure 6. The author has been working with Sprint Layout for years, designing his own PCBs.

A common yet specific example

Very often we find in a magazine, like *Elektor Electronics*, a circuit that is really what we need but we would like to make a few modifications: move a connector, replace one part by another which is not the same size, etc. Then the only problem is drawing the new printed circuit. Of course, the most courageous may recopy the published design completely and make small changes to necessary areas, but it is tedious and worse, a source of errors, because it is very easy to forget one or more traces.

Therefore, please note that, to our knowledge, the only low-priced product of this type is the **Sprint Layout** software from Abacom. Used to finely tweak any previously scanned image under the design of a printed circuit in progress. This function is ideal for making the type of modifications mentioned above since you only need to scan the design published in the magazine that we copied over with great

ease, as if we had applied tracing paper to the screen.

Editor's note: the most recent version of Easy-PC also supports Reverse Engineering functionality.

Almost there...

With your construction designed once and verified twice (better than once!), you are ready to move on to building the printed circuit. Here there are still two options to choose from depending on the software you have used and the financial means available.

If you have worked with an electronic CAD package and you are targeting industrial production, even in a small quantity, your software is able to provide Gerber files, something that printed circuit manufacturers are capable of interpreting. You give them these files and a few days later you will receive a PCB to match your design. All that is left to do is to populate it, wire it up and test it.

If you are a hobbyist or if you work in a small business and would like to make just one or two circuit boards for a reasonable price, you can do so very easily, regardless of the type of software you used so far.

The simplest solution today is to have your printed circuit design printed on a transparency and use this transparency directly as film for the etching of your circuit board. The best results are obtained with a laser printer and appropriate transparencies but if you only have an inkjet printer, you can still make it work if you use transparencies especially intended for inkjet printers. In the other case, the transparency does not retain enough ink and the blacks are not dense enough which causes multiple tiny broken tracks. You can also have your design made in double and superpose both transparencies to make 'the film' insulation of your printed circuit, which further decreases the risk of tiny broken tracks. The author has been doing just that for years with a low-end laser printer and can guarantee that circuits can be manufactured in this way without problems by putting one trace between two neighbouring integrated circuit pads without the slightest tiny broken line.

We hope it doesn't happen but this final step sometimes leads to one or several loopbacks as our flow chart shows so well, because whatever safeties and controls you have built-in, some errors may only show up during the testing of this printed circuit: interaction between traces or between components, component tolerances that make the construction unstable. The only thing left for you to do is to start over again. If that is the case, don't forget to correct all of your documents when you modify something in your project. If you have been using CAD software, verify that it can withstand back annotation. In other words, if you make a modification to the printed circuit, it is automatically reflected in the schematic (the software functions retroactively as compared to normal operation which explains the origin of the term back annotate). If you use independent software, it is your responsibility to do that by hand if you want your design file to correspond to the finished product.

(050275-1)

Internet Links:

- [1] Discovercircuits.com - www.discovercircuits.com
- [2] Sam Electronic Circuits - <http://users.otenet.gr/~athsam>
- [3] see the 'Kaleidoscope' article elsewhere in this issue — it offers many links to manufacturers of schematic capture software, PCB design and simulation
- [4] <http://bwrc.eecs.berkeley.edu/Courses/icBook/tools.html>

Christian Tavernier

www.tavernier-c.com — contact@tavernier-c.com

Voltage mode or How to choose an

In many applications ordinary opamps are outperformed by their more exotic cousins

Von Thomas Ußmüller and Ulrich Tietze

There are three other designs of opamp besides the familiar type with high input impedance and low output impedance. In this article Ulrich Tietze (co-author of a well-known book on circuit design) and Thomas Ußmüller find out what these other types can offer in terms of bandwidth with the help of a few essential circuits.



current mode?

opamp

What would electronics be like without operational amplifiers? Almost every electronic device that uses analogue circuitry will contain one or more of these components. It is hard to believe that the first opamps were constructed from valves: as in so many things, developments were led by military requirements, for example in applications such as controlling the aim of guns during the war. The first commercial opamps did not come onto the market until 1952. The valve opamp designed by George A. Philbrick Researches was fitted with an octal socket (see photograph).

Shortly thereafter the first opamps made from discrete transistors were produced. The first integrated opamp was the $\mu A702$, designed for Fairchild by Robert Widlar. It was first available in 1962. Six years later came the $\mu A741$, which has maintained a hold on the market ever since. These important devices are now produced in their millions, and they are used in applications ranging from audio and video amplifiers to active filters and sensor signal conditioners.

There are four basic types of operational amplifier: the device can have a high or a low input impedance inverting input and a high or a low output impedance output (see **Figure 1**). The most widely-used type, familiar to practically all electronics engineers, is the voltage-mode, or 'VV' type. This has a high input impedance and a low output impedance (the non-inverting input has a high impedance on all device variants). In many circuits, however, one of the alternative types, called 'CC' (current-mode), 'VC' (voltage-to-current, or transconductance) or 'CV' (current-to-voltage, or transimpedance) amplifiers, can offer a significantly higher bandwidth. We shall now examine this claim in more detail with the help of a couple of simple circuits.

Let's build an opamp

First of all we shall show how the four types of operational amplifier can be constructed from basic blocks. To make the comparison as fair as possible we shall use the same components in each case. We decided to use a technique, common in CV- and CC-type operational amplifiers, called push-pull class AB (also known as 'current on demand') operation.

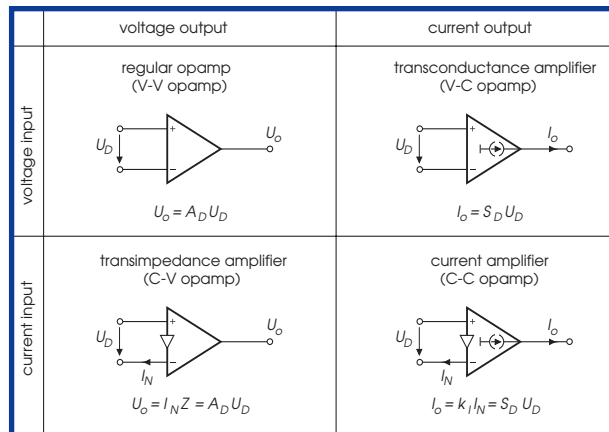


Figure 1. The four different types of operational amplifier shown schematically.

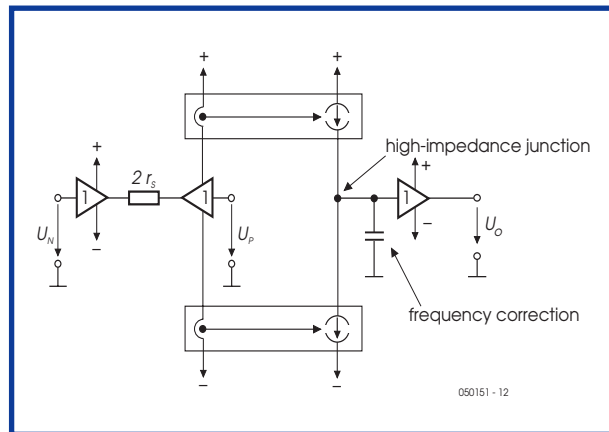


Figure 2. In class AB push-pull operation the voltage-mode amplifier uses only two different basic blocks: a voltage follower and a current mirror.

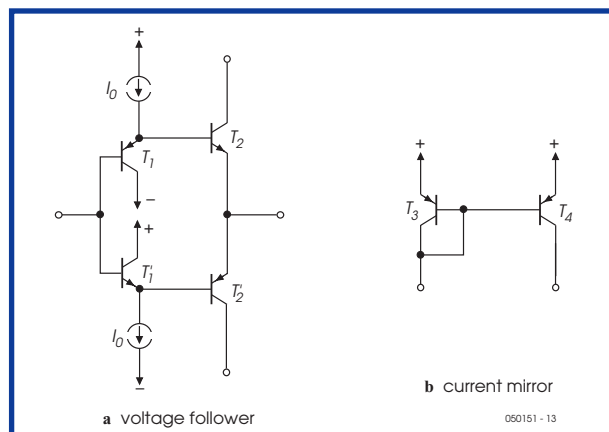


Figure 3. Internal circuit of the voltage follower and current mirror.

Figure 4.

Model of a real voltage-mode operational amplifier. The differential gain is determined by resistors r_s in the impedance converter at the input and by the impedance at the high-impedance point.

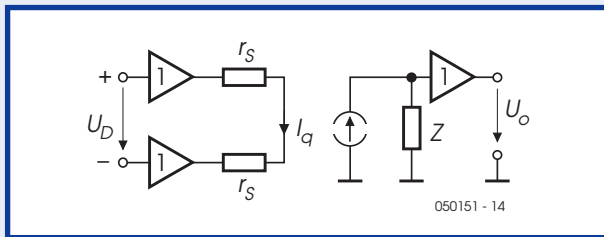
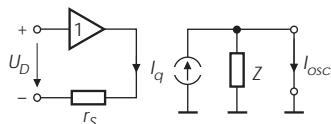
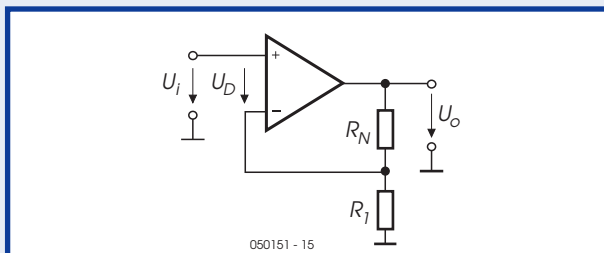
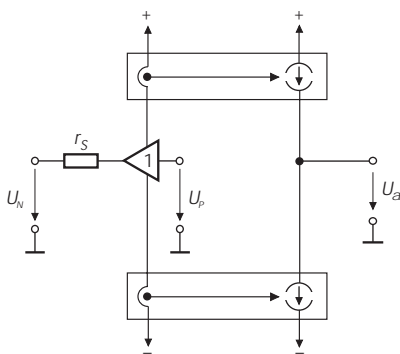


Figure 5.

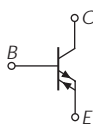
Connecting a voltage-mode operational amplifier for a bandwidth test ($R_N = R_1 = 100 \Omega$).



a model



b equivalent diagram



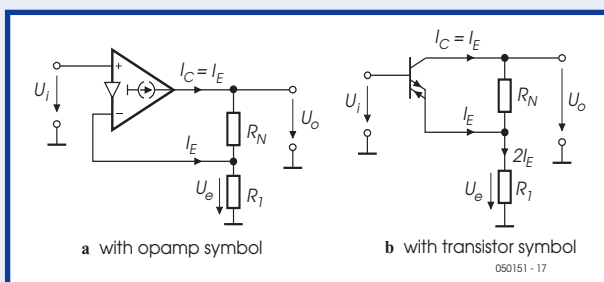
c circuit symbol

Figure 6.

Model of a current-mode operational amplifier.

Figure 7.

A current-mode operational amplifier used as a non-inverting amplifier with negative voltage feedback ('direct feedback'): $R_N = 200 \Omega$, $R_1 = 100 \Omega$.



As we can see from **Figure 2**, there are only two basic circuits that we need: a voltage follower and a current mirror. The innards of these circuits are shown in **Figure 3**.

The two complementary voltage followers form the necessary differential amplifier at the input to the opamp. The signal is taken in the form of a current drawn from the power supply from the voltage followers to the current mirrors. The advantage of push-pull class AB operation when compared to a conventional differential amplifier is that, despite the low quiescent current, almost arbitrarily large peak currents can be obtained. A consequence of this is that there is no slew rate limit. The output of the two current mirrors is the high impedance point of the circuit. If necessary, a capacitor (called a 'compensation capacitor') may be connected to ground from this point. The final part of the operational amplifier circuit is a voltage follower which provides a low impedance output.

Voltage mode under the microscope...

A voltage-mode operational amplifier is characterised by high-impedance inputs and a low impedance output. It operates as a voltage-controlled voltage source. The ideal amplifier exhibits a gain of

$$A_D = U_o / U_D = \infty \text{ (infinity).}$$

Real opamps have a differential gain of between 10^3 and 10^6 . From the model of a real amplifier (**Figure 4**) we can see that the differential gain is determined by the transconductance resistances $r_s = 1/S$ in the impedance converter at the input and by the impedance at the high impedance point.

To investigate its bandwidth the operational amplifier is connected as shown in **Figure 5**. The feedback resistors have values $R_N = R_1 = 100 \Omega$. This gives the circuit a gain of

$$A = 1 + R_N / R_1 = 1 + 100 \Omega / 100 \Omega = 2.$$

Without a compensation capacitor we get considerable overshoot in the step response of this amplifier, and so we connect a compensation capacitor to the high impedance point. This succeeds in suppressing overshoot but, unfortunately, also reduces the bandwidth of the circuit. The frequency response of the circuit is shown in **Figure 8**.

... and current mode

A current-mode operational amplifier has a low-impedance inverting input and a high-impedance output. Exactly as with other types of operational amplifier, the non-inverting input has a high impedance. The model is shown in **Figure 6**. Another common name for this type of amplifier is 'diamond transistor', which derives from the similarity of its behaviour to that of an ideal bipolar transistor. Commercially-available current-mode operational amplifiers include the MAX435 and MAX436 from Maxim. The well-known OPA660 from Burr-Brown is sadly no longer produced, but comprehensive documentation is available on the Internet: see the links below. To use a current-mode operational amplifier as a non-inverting amplifier, it can be connected in the same way as a voltage-mode opamp with negative voltage feedback: see **Figure 7**. This circuit arrangement is also

called 'direct feedback'. In our simulation we set $R_N = 200 \Omega$ and $R_1 = 100 \Omega$. The gain is again given by

$$A = 1 + R_N / 2 R_1 = 1 + 200 \Omega / (2 \times 100 \Omega) = 2.$$

The cutoff frequency of the voltage amplifier can be determined from **Figure 8**.

An alternative arrangement for constructing a non-inverting amplifier from a current-mode opamp uses the opamp's ideal transistor properties. It can be connected in an emitter-follower configuration with negative current feedback (**Figure 9**). Here the resistor values are $R_C = 200 \Omega$ and $R_E = 100 \Omega$. The gain is then

$$A = R_C / R_E = 200 \Omega / 100 \Omega = 2.$$

This arrangement is termed 'current feedback'. The bandwidth is similar to that achieved using negative voltage feedback (see **Figure 8**).

As well as the amplifiers we have described here, we have also simulated a VC opamp and a CV opamp, which we can construct from the same basic blocks. Again, we examined the bandwidth of each circuit; we have collected the results together in the **Table**. There are only small differences in bandwidth between the various amplifier configurations. Voltage-mode and VC opamps require a capacitor to correct the frequency response, thus ensuring stable operation of the amplifier.

Which is the best opamp to use in an integrator?

An important operational amplifier circuit is the integrator. It finds practical application in, for example, filters ('biquads'). **Figure 10** shows the circuit of an integrator using a voltage-mode operational amplifier, which can operate at a frequency of up to 2 MHz. Above that frequency is a region where the gain increases again (see **Figure 11**). The reason for this behaviour can be seen from the equivalent circuit (**Figure 12**). Capacitors C and C_1 are effectively a short-circuit to high frequencies. A voltage divider is thus formed by R and r_o , giving an output voltage of

$$U_o = r_o / (R + r_o) U_i.$$

The current-mode operational amplifier can be connected as an integrator as shown in **Figure 13**. As the frequency response clearly shows, the bandwidth of this circuit is much greater than that of the voltage-mode integrator. The upper frequency limit is around 24 MHz (see **Figure 11**). Also, there is no frequency range over which this configuration exhibits differentiating behaviour. This is because the parasitic capacitance at the high impedance point of the current-mode operational amplifier is connected in parallel with the integrator capacitor and does not cause a parasitic 'pole' on the frequency response.

In summary we can say that the current-mode operational amplifier is much better suited to use in integrator circuits than the voltage-mode opamp: firstly, its bandwidth is some ten times greater, and secondly, it does not exhibit differentiating behaviour at higher frequencies.

And finally: a filter

Finally we shall look at the frequency response of a second-order Butterworth low-pass filter. The filter was

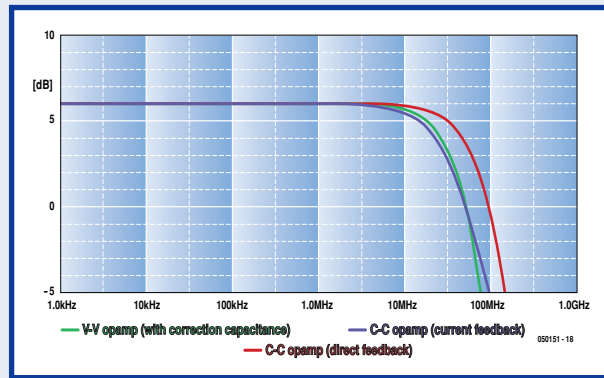


Figure 8. Frequency response of the amplifiers. Shown are a voltage-mode opamp with compensation capacitor (green), a current-mode opamp with direct feedback (red) and a current-mode opamp with current feedback (blue).

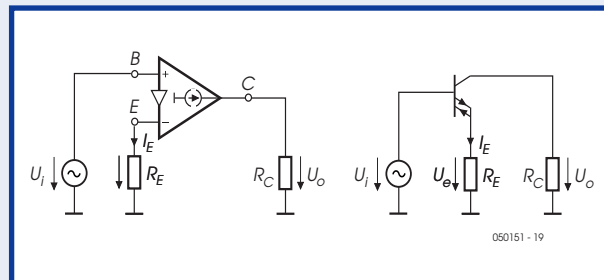


Figure 9. A current-mode opamp connected as an ideal transistor in an emitter-follower configuration with negative current feedback: $R_C = 200 \Omega$, $R_E = 100 \Omega$.

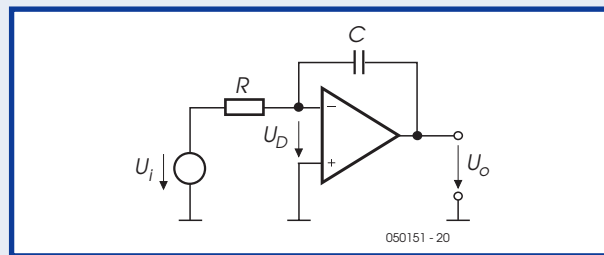


Figure 10. Circuit of an integrator using a voltage-mode operational amplifier.

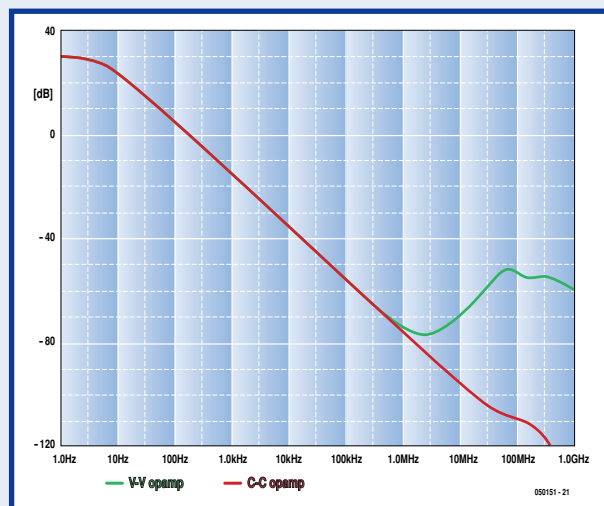


Figure 11. Characteristic curves of integrators realised using voltage-mode (green) and current-mode (red) opamps. The current-mode circuit is clearly superior.

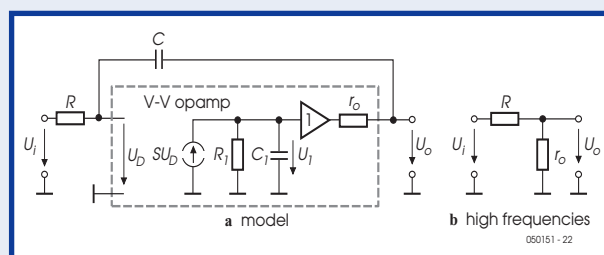


Figure 12. Equivalent circuit of an integrator constructed from a voltage-mode opamp. Capacitors C and C_1 are effectively short circuits at high frequencies.

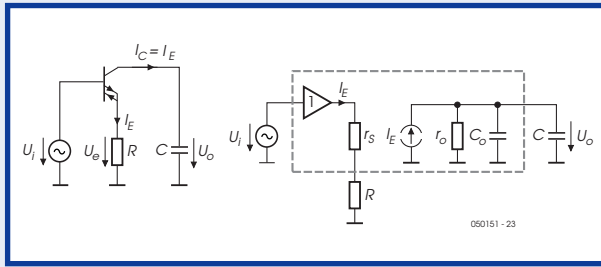


Figure 13. Circuit and model of an integrator using a current-mode operational amplifier.

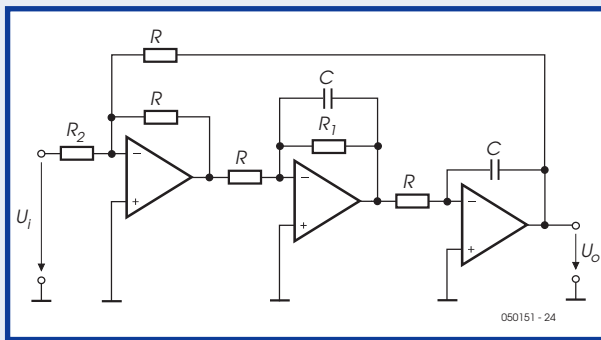


Figure 14. Second-order Butterworth low-pass filter realised using voltage-mode integrators.

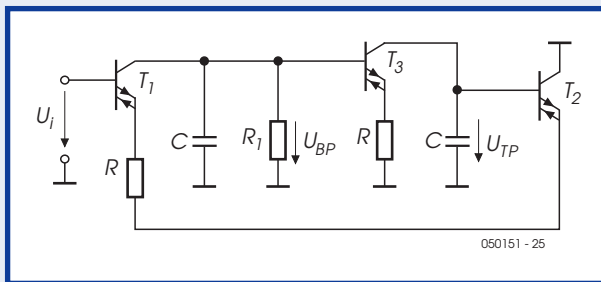


Figure 15. Second-order Butterworth low-pass filter realised using current-mode integrators.

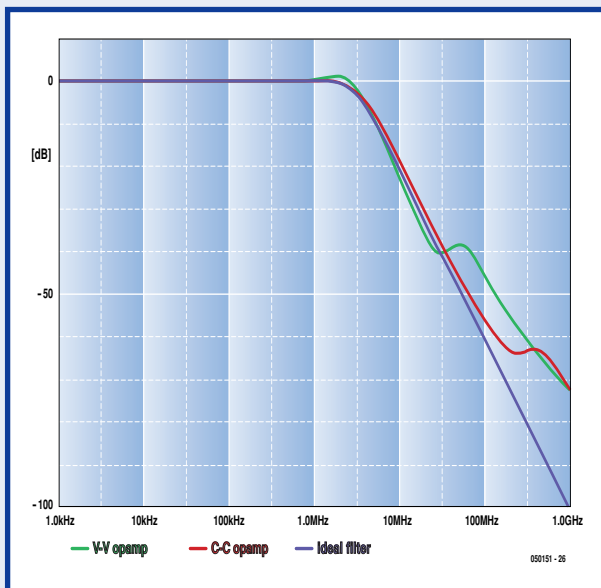


Figure 16. Characteristic curves of the low-pass filters. The integrator filter using current-mode operational amplifiers (red) follows the ideal characteristic (blue) more closely than the voltage-mode integrator filter (green), and is suitable for use with frequencies of up to 100 MHz.

Bandwith comparison

Amplifiers based on different opamps

Amplifier	Bandwidth
VV	32.1 MHz
VC	49.4 MHz
CV	59.1 MHz
CC (direct feedback)	58.1 MHz
CC (current feedback)	29.5 MHz

realised using voltage-mode integrators (see circuit in **Figure 14**) and with current-mode integrators (see circuit in **Figure 15**). An ideal filter was used as a benchmark. The cutoff frequency chosen was $f_c = 3$ MHz. Looking at the frequency responses (**Figure 16**) we can see that the filter using voltage-mode integrators (green curve) soon departs from the ideal characteristic (blue curve). At frequencies above 30 MHz the filter is no longer usable. The integrator filter using current-mode operational amplifiers (red curve) follows the ideal characteristic more closely than the filter using voltage-mode integrators; it is suitable for use with frequencies up to 100 MHz.

Conclusion

The conclusion is that a 'normal' voltage-mode operational amplifier is adequate for most applications. In particular, when used as an amplifier, the difference between the types is not great as long as the same circuit configuration is used. In integrators and integrator filters, however, current-mode operational amplifiers outperform voltage-mode opamps in respect of bandwidth by a factor of around ten.

(050151-1)

Reference

Tietze, U. and Schenk, C.:
Electronic Circuits: Handbook for Design and Application,
Springer, ISBN 3-540-50608-X / 0-387-50608-X

Web links

- www.eetimes.com/anniversary/designclassics/opamp.html
- http://en.wikipedia.org/wiki/Operational_amplifier
- www.uoguelph.ca/~antoon/gadgets/741/741.html
- <http://focus.ti.com/lit/an/sboa071/sboa071.pdf>

Remote control by mobile phone

Switch & Control using DTMF signalling

Heikki Kalliola

Many of you will have an old mobile phone gathering dust. The circuit shown in this article gives the old mobile a new lease of life, enabling it to switch an electrical appliance on and off in or around the home. The circuit will even talk to you on in reply to predefined commands.

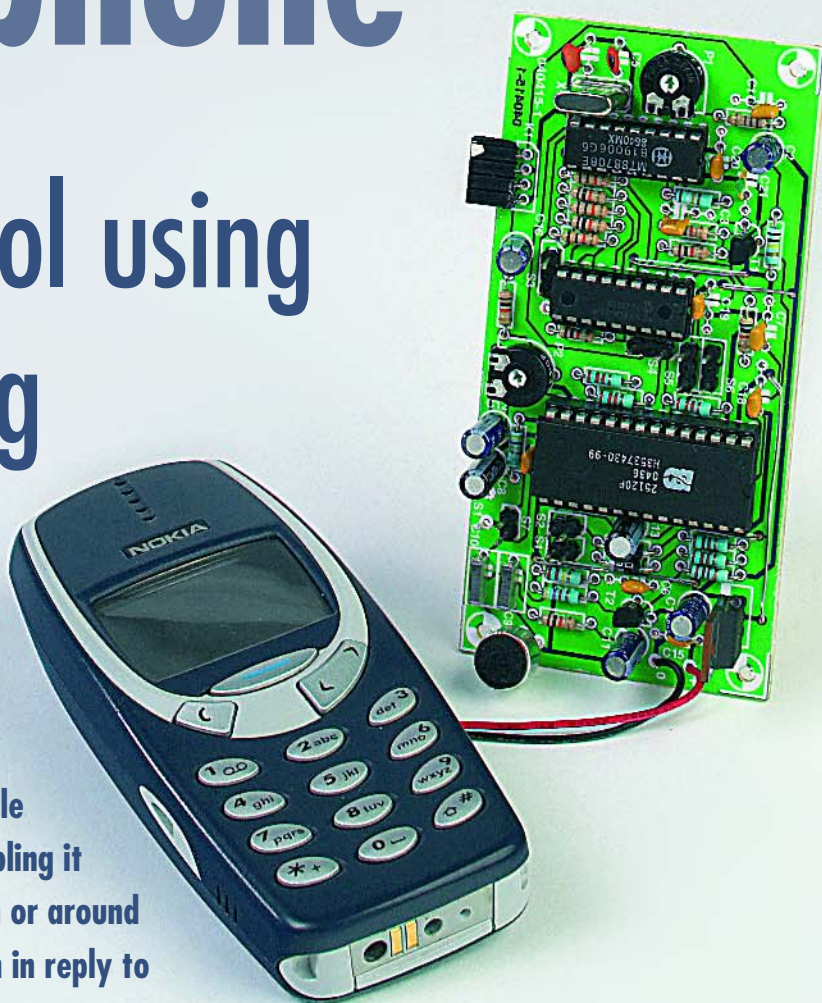
Commercially available equipment for telephone-controlled switching usually employs special GSM modems in combination with SMS messaging ('texting'). *Elektor Electronics* once published such a circuit for home construction, see 'SMS-Controlled Mains Sockets' in the January 2003 issue. Unfortunately, GSM modules are fairly

expensive so the idea comes to mind of using a discarded or obsolete mobile phone you may have lying around somewhere.

The above functionality is within reach with the present circuit. The mobile phone is linked to it via the connector normally used for an external microphone and earphones (some restricting

conditions apply in this respect, see the inset).

If your old mobile employs a prepaid card (i.e., it is not on subscription with a telco) then it allows the worldwide remote control to be implemented inexpensively. After all, you then only pay for the call on the calling phone, not the receiving phone.



The overall complexity of the circuit is greatly reduced by the use of a PIC microcontroller. The addition of a speech synthesizer chip allows the control unit to return spoken information to the user on the calling phone.

The system

The schematic of the Remote Control by Mobile Phone is given in **Figure 1**. The connector for the headset is plugged into the mobile phone and the wires to the earphone pieces are connected to the input of a DTMF decoder IC type 8870 (IC1). The received signals are amplified by T1 and then applied to pin RB2 of the central microcontroller, a PIC16F84 (IC2).

DTMF stands for *dual-tone multi-frequency*, a tone coding/decoding system that's been in use on wired telephone sets for almost 30 years now. Each key on the DTMF keypad is linked to a combination of two (non-harmonically related) tones in the audible range. At the receiving side, the tone pairs are decoded into the matching number (0-9, #, *).

For our application, the mobile phone is

set to **auto-answer mode**. In operation, the PIC micro will continuously monitor the level at RB7 (pin 13). A ring signal will wake up the PIC from its sleep state and cause it to activate the speech chip (IC3) which supplies a 'spoken' message to the mobile phone via the microphone connector of the headset. At the calling end, the user hears the welcome message and is prompted to enter the system password (MSG1). The level of the audio signal may be adjusted on preset P2. The selection of spoken text stored in the speech chip is determined by the logic state of control lines A6, A7 and A8.

Next, the program running inside the PIC will wait for certain DTMF tones entered by the user on his/her phone (mobile or landline). The received DTMF signals are decoded into the matching numbers by IC1, which supply binary output information on outputs Q1-Q4, with output STD (pin 15) flagging reception of a valid code to the PIC.

Having received a valid 4-digit password (which is being compared to the

one stored in the PIC's memory), the system will request a function code (MSG2). Depending on the information received in response to MSG2, the micro will pull its RB5 control line logic Low or High. This signal enables a relay to switch a suitably wired up electrical appliance in your home to be switched on or off (for example, a coffee machine or the central heating). The relay may be a solid-state device (which allows direct control from RB5), or a traditional solenoid-based switcher that's controlled via a transistor buffer.

The action executed at the receiving side is confirmed with a spoken message (MSG4 or MSG5). The speech chip contains yet other text messages (MSG3, MSG6 and MSG7) for the processing of situations 'out of the ordinary'.

The speech chip

The spoken messages are supplied by a type ISD25120 speech synthesiser which has storage capacity for about two minutes worth of speech. Different memory areas inside the ISD25120

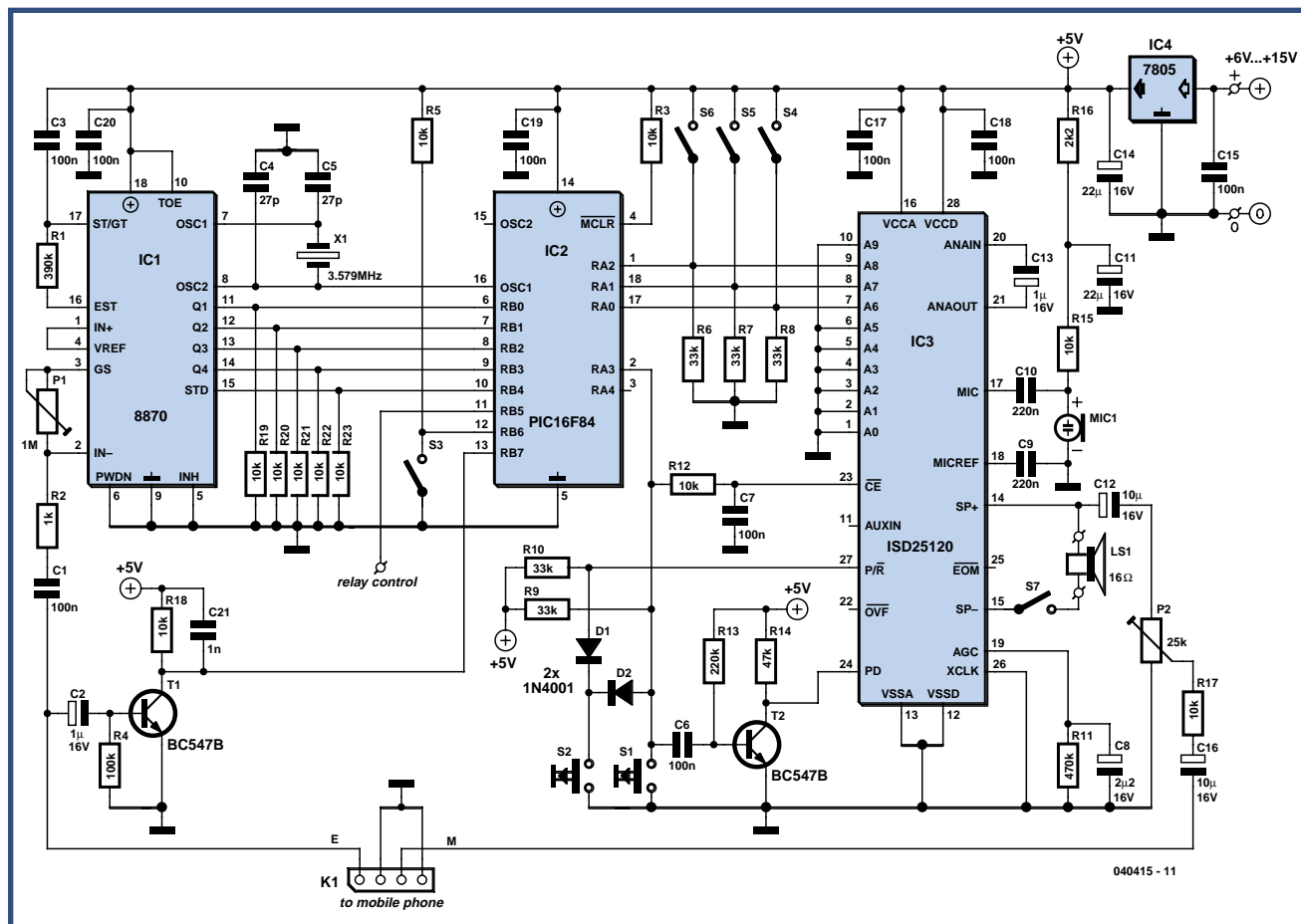


Figure 1. The circuit contains just three ICs: a DTMF decoder, a PIC micro looking after the internal control functions, and a speech chip.

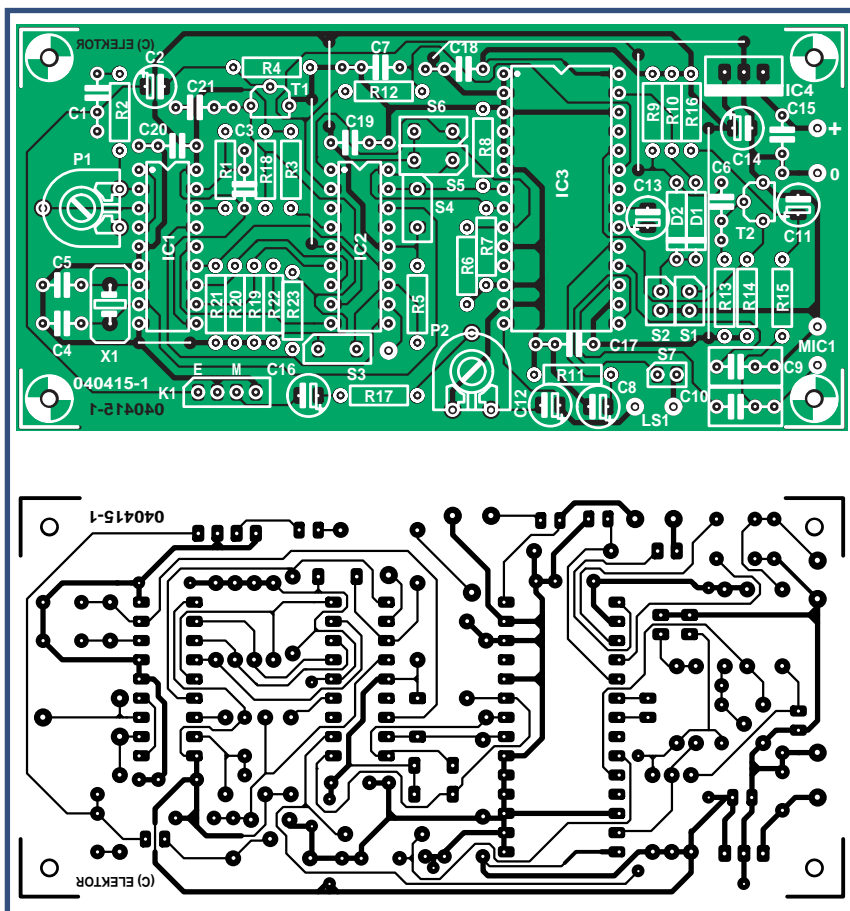


Figure 2. Copper track layout and component mounting plan of the PCB designed for the remote control. Be sure to use sockets for the ICs, as the PIC has to be removed during recording of the spoken messages.

COMPONENTS LIST

Resistors:

R1 = 390k Ω
 R2 = 1k Ω
 R3,R5,R12,R15,R17-R23 = 10k Ω
 R4 = 100k Ω
 R6-R10 = 33k Ω
 R11 = 470k Ω
 R13 = 220k Ω
 R14 = 47k Ω
 R16 = 2k Ω
 P1 = 1M Ω preset H
 P2 = 25k Ω preset H

Capacitors:

C1,C3,C6,C7,C15,C17-C20 = 100nF
 C2,C13 = 1 μ F 16V radial
 C4,C5 = 27pF
 C8 = 2 μ F 16V radial
 C9,C10 = 220nF
 C11,C14 = 22 μ F 16V radial
 C12,C16 = 10 μ F 16V radial
 C21 = 1nF

Semiconductors:

D1,D2 = 1N4001
 T1,T2 = BC547B

IC1 = MT8870BE

IC2 = PIC16F84A-20/P (programmed,
 Publishers' order code **040415-41***)

IC3 = ISD25120P (Winbond, Digikey #
 ISD25120P-ND)

IC4 = 7805CP

Miscellaneous:

K1 = 4-way SIL pinheader
 LS1 = 16 Ω miniature loudspeaker
 MIC1 = electret microphone capsule
 S1,S2 = pushbutton, 1 make contact
 S3-S7 = on/off switch, 1 make contact
 X1 = 3.579MHz quartz crystal
 Mobile phone with headset (see text)
 PCB, Publishers' order code **040415-1***
 Disk, PIC source and hex code files,
 Publishers' order code **040415-11** or
 Free Download

* see Elektor Electronics SHOP pages or
 www.elektor-electronics.co.uk

may be selected using address lines A0-A9. Here, the memory has been divided into eight chunks of about 15 seconds that can be individually addressed over A6, A7 and A8. This allows the PIC to select any individual spoken messages and have it played back by pulling the **CE** control input Low.

The spoken messages may be recorded and checked using a microphone and a small loudspeaker connected to IC3. Switches S4, S5 and S6 select the relevant memory area. The relevant message is then recorded on the chip via MIC1 while S2 is pressed. The pushbutton should be released at the end of your message, which can be listened to by briefly pressing S1 (this requires switch S7 to be closed). If you are not satisfied with the result, the text may be re-recorded.

During recording and playing back the messages (for checking) the PIC has to be removed from its socket, else IC3's address lines cannot be set manually. The inset 'Messages' gives examples of texts you could use for the voice feedback, including the associated address settings.

Password changing

In an application like this, where equipment is controlled from a remote location, it is essential to prevent unauthorised access. In any case, password protection is required.

You, the user, may pick your own 4-digit password using the number keys on your mobile phone's keypad. For this to work you have to open switch S3 (which is closed during normal use). In that case, the four numbers entered on the calling phone after the welcome message has finished will be accepted as the new password, and subsequently stored in the PIC's RAM memory. If mains outages are common in your home, we recommend powering the circuit from a battery system with mains charging, else the password will be lost.

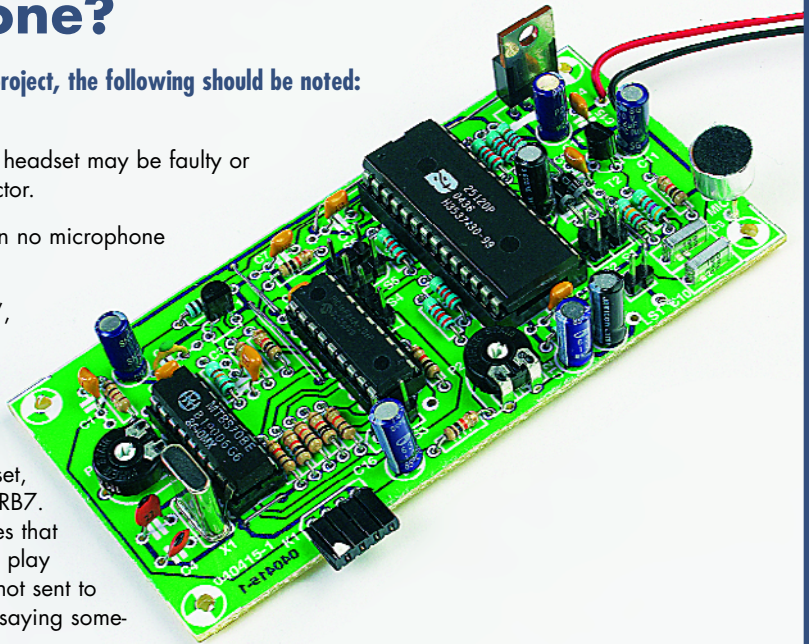
Construction

The above article text having largely described the operation and practical use of the circuit, you should know the function of all switches and other controls and use this information to fit the relevant parts on the case in such

What mobile phone?

When selecting a mobile phone for use with this project, the following should be noted:

1. You will require a mobile with a headset. The headset may be faulty or incomplete — basically we only need the connector.
2. Some mobiles will not detect the headset when no microphone is connected.
3. The mobile has to support 'auto-answer mode', that is, it has to be able to answer an incoming call on the first ring signal. Some mobiles will not show this option until a headset is actually connected.
4. The ring signal has to be audible in the headset, too, as it its function is to 'wake' the PIC via T1/RB7. The first pulse edge on the PIC's port pin indicates that DTMF odes are imminent and also causes IC3 to play back the welcome message. If the ring signal is not sent to the headset the circuit may also be activated by saying something into the calling phone.



a way that the unit is easy to operate in practice.

The source code of the PIC control program written by the author is available as a free download from our website, as well as on floppy disk (order code **040415-11**). Those without access to a suitable PIC programmer may also buy the PIC micro ready-programmed as order code **040415-41**.

The PCB artwork is shown at true size in **Figure 2**. With all components fitted

(insert the ICs in sockets) the controls may be connected up via short lengths of stiff wire: S1, S2, S3, S4, S5 and S6, the electret microphone capsule and (if required) the loudspeaker (together with S7). The latter parts are mainly needed to check the spoken messages, switch S7 allowing the loudspeaker to be switched off during normal use. Next, the mobile phone link may be hooked up on connector K1. The output relay for the electrical appliance you wish to control (literally from anywhere

in the civilised world!) is connected to the terminals beside S3 on the board. The circuit is happy with a 6-15 VDC supply voltage from a mains adaptor capable of supplying 100 mA or so.

There is only one adjustment in the circuit: preset P1 determines the input sensitivity of the DTMF IC. The setting is not critical however and the circuit will usually work with the wiper at mid travel.

(040415-1)

Messages

Message	Address			Text
	A8 (S6)	A7 (S5)	A6 (S4)	
MSG1	0	0	0	This is Joe Blogg's remote control by mobile phone. Please enter system password
MSG2	0	0	1	Correct. Please enter function code. ONE to switch on appliance, TWO to switch off appliance.
MSG3	0	1	0	Available time for code entry elapsed. The connection will be broken. Bye!
MSG4	0	1	1	Appliance switched on. The connection will be broken. Bye!
MSG5	1	0	0	Appliance switched off. The connection will be broken. Bye!
MSG6	1	0	1	Password incorrect, please try again.
MSG7	1	1	0	Password successfully changed. The connection will be broken. Bye!

Programming in pictures

Coding without keywords

Burkhard Kainka

Microcontrollers are particularly hard for beginners to program. Graphical user interfaces help to show interdependencies, but they have their limitations. We look at a couple of systems and some simple application examples.

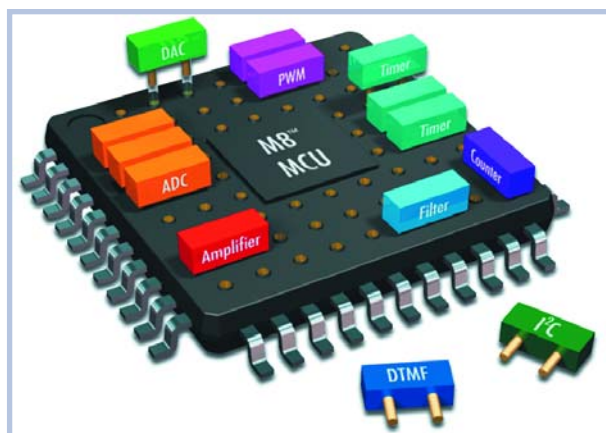


Figure 1.
Schematic diagram of a
PSoC and its
functional blocks.

Everyone has had the experience of recognising a face, but being unable to put a name to it. It is just the same in programming: commands and keywords, register names, the names of all the various control flags: it is a lot to hold in the mind at once. It is often easier to think in a 'graphical' way, for example describing the operation of a microcontroller in terms of a flow chart or an equivalent circuit diagram, before embarking on the work proper. This is almost the same as using a graphical interface: instead of typing in program text, you can simply click on logic functions and connections, draw lines between objects with a sweep of the mouse, and set properties of various elements by simply selecting from a list.

Some manufacturers of microcontrollers have recognised this demand. In the struggle for market share it is no longer enough just to compete on performance figures; the product must also be easy for potential customers to use. The value of making a user's first experience with a

new chip a success makes the (high) development cost of graphical programming tools entirely justifiable.

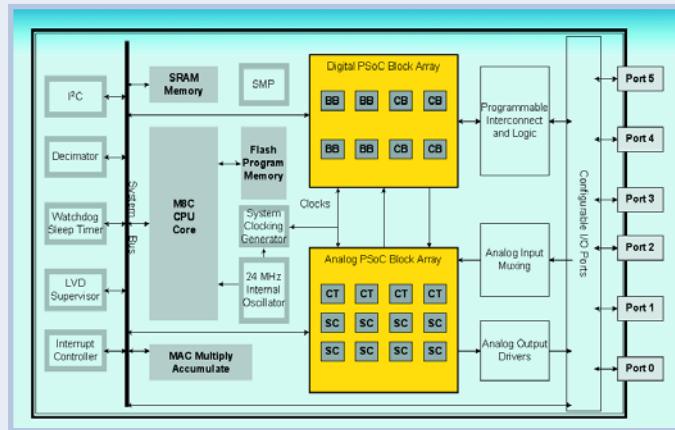
We frequently have the choice between a graphical and a text-oriented programming environment. In general the graphical environment will seem simpler since in many cases decisions have already been made behind the scenes to make it harder for us to make mistakes. This can, however, be a limitation in more complex projects. The immediate experience with graphical programming environments and the ready-made projects supplied with them is indeed positive: it looks so easy that anyone could use it. However, this overlooks the fact that the interface is hiding many decisions. It is possible, for example, that objects have been given various properties that can only be revealed by clicking on them with the mouse. If you take a working example project, it is not always immediately obvious how many properties and decisions are hidden away in the background. Perhaps only when you try to build your own project from the ground up will it become apparent that you are standing before as great a mountain of information as if you had used a text-based interface.

In other cases you might embark confidently on an ambitious programming task, seemingly made child's play by the graphical interface. Then suddenly an insurmountable obstacle crops up. The problem could be solved with three lines of program code; but unfortunately the whole project would then have to be restarted from scratch in text form.

Whether a graphical programming interface is worth using, and whether it makes work any easier, can only be decided easily in a handful of cases. Nevertheless, it is worth familiarising oneself with this style of programming. The reports of our experiences below will give an overview of the possibilities. In this article we will not be

The PSoC family

The figure shows the PSoC architecture divided into four main blocks: the processor core, the digital peripherals, the analogue peripherals, and system resources. Configurable global buses allow these to be connected to form a complete system. One member of the PSoC family is the CY8C27443, which can use up to five I/O ports as analogue and digital inputs and outputs. There are eight digital blocks (8 to 32-bit timer, counters, PWM generators, up to two full-duplex UARTs, SPI masters and slaves etc.) and twelve analogue blocks (A/D converters up to 14 bits, D/A converters up to 9 bits, programmable amplifiers, filters and comparators etc.). The device has an M8C processor core running at up to 24 MHz and has an internal 24 MHz/48 MHz oscillator, 16 kbytes of Flash program



memory and 256 bytes of data RAM. The PSoC Designer software is free.

covering the microcontroller building-block system **E-Blocks**, whose **Flowcode** programming interface (shortly to become available at <http://www.elektor-electronics.co.uk>) is suitable for use even by beginners: more on this system will be found elsewhere in this issue.

PSoC Designer

The PSoC family from Cypress is a series of microcontrollers equipped with an extra mixed-signal array. Each PSoC controller includes highly-reconfigurable analogue and digital blocks (see **Figure 1**) with programmable interconnections. This architecture allows the user to match the external interfaces of the device to the requirements of the application. The device also includes a fast CPU with in-circuit programmable Flash ROM, RAM, and configurable ports.

Programming and connecting the various blocks is very complex, since there are countless configuration possibilities, as well as many constraints. Finally, hidden behind the configuration of each PSoC block and the connections that form a circuit is a large number of registers and corresponding switches. Programming all these bits by hand is a job beyond mere mortals, and so the only

practical approach is to develop the circuit graphically. The user is assisted in this by 'PSoC Designer', a complex programming environment which even includes an assembler and a C compiler. Generally, most of the time is taken in constructing the circuit. With small projects at least, the amount of text-oriented programming is relatively small.

The quickest way to start using the system is to take a ready-made PSoC project and expand and alter it. In this way all the necessary settings for a project will have already been made correctly. In our case, we took the example project `asm_example_dac_adc.SOC` configured for the CY8C27443. The program turns the device into an adjustable sine wave generator, taking data from a lookup table of sine values and passing them to a D/A converter. An A/D converter measures the voltage at an analogue input and sets a timer appropriately so that a potentiometer can be used to control the frequency of the output sine wave. To expand this project, create a new project with its own name, making a clone of the example project.

Once the project has been loaded it is possible to select between the Application Editor, which shows the assem-

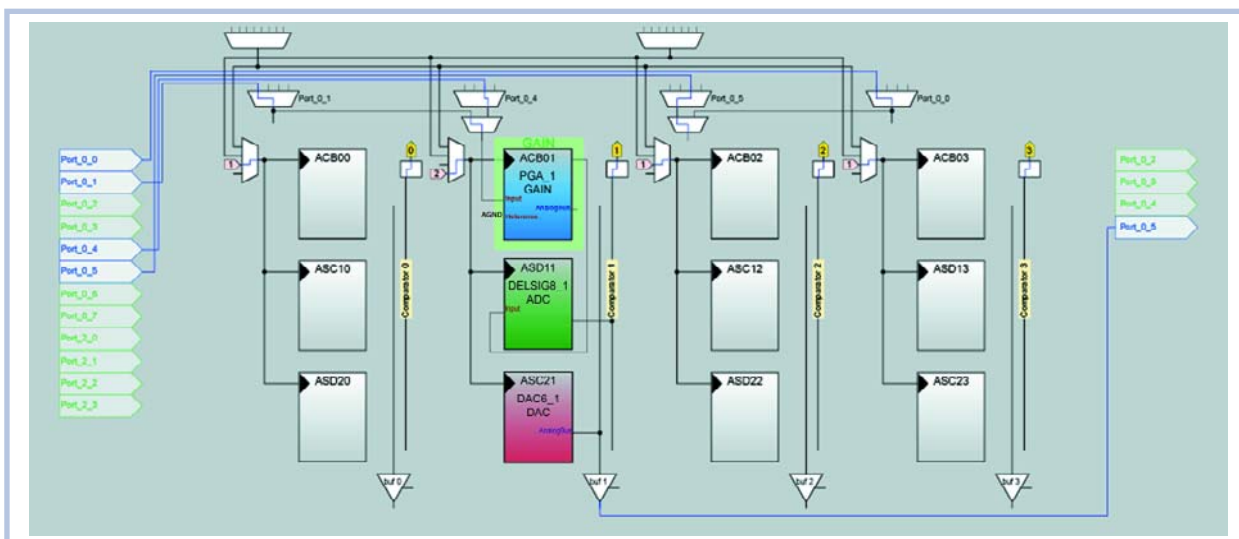


Figure 2. The analogue blocks that are in use are highlighted in colour in PSoC Designer's Device Editor.

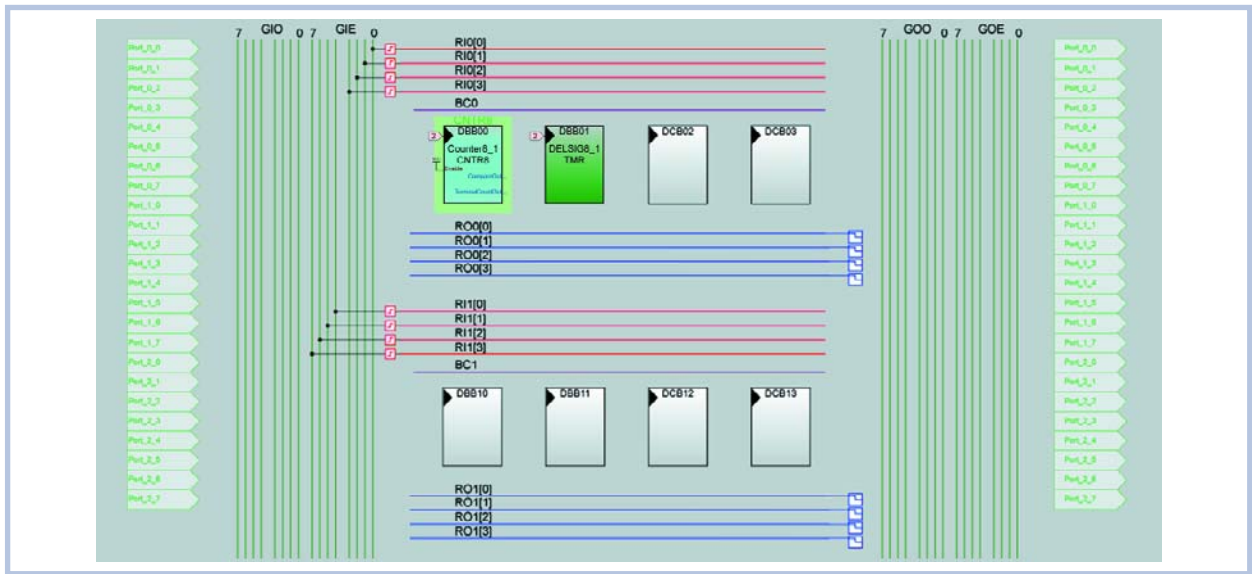


Figure 3. The sine wave generator project also uses the digital block Counter8_1.

bler source code, and the graphical Device Editor. **Figure 2** shows a section of the Device Editor screen (here showing only the analogue PSoC blocks and their connections). Four configured blocks can be seen: PGA_1 is a programmable amplifier, here used as an input buffer with gain set to 1. The input is connected to pin Port_0_1. The output of the buffer amplifier is connected to the input of the ADC block DELSIG8_1, a delta-sigma converter with 8-bit resolution. The third PSoC block is the 6-bit D/A converter DAC6_1, whose output is connected to pin Port_0_5. In addition to the analogue blocks, this project also uses the digital block Counter8_1 (see **Figure 3**). The timer sets the rate at which the sine values are output, but is not itself connected to an output pin. The digital part of the delta-sigma converter can also be found in the digital section.

Pin overview

The complete circuit as seen from the outside has one input, where a potentiometer is connected (VR), and one

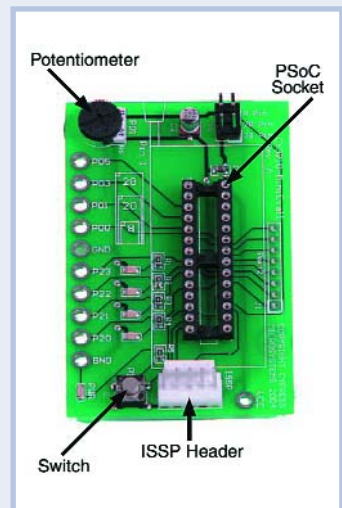
output for the sine wave signal (AnalogOutBuf_1). The selected pins can be seen in the Device Editor (see **Figure 4**). This is practically the complete circuit: of course V_{SS} and V_{DD} need to be connected, but there is no crystal and no reset circuit. The microcontroller operates from an internal 24 MHz RC oscillator with an accuracy of $\pm 2.5\%$.

Each PSoC block has a large number of configurable parameters, which can be set at the design stage. The parameters can also be changed under program control. Whenever a new block is added to the circuit, PSoC Designer adds the necessary code to the project in the form of a library module. The main program main.asm then remains very high-level and can be examined in the Application Editor view. The blocks which are used must first be enabled, which is done using standard calls. The program then runs in a simple loop (see listing).

The project can now be built using 'Build'. The result is the file 'Sine.hex' in the 'Output' subdirectory of the main project directory. This file can now be transferred to the

MiniProg and MiniEval

The low-cost CY3210 starter kit includes: the MiniProg1 programming device, which has a USB interface; a small MiniEval printed circuit board for the processor, with a 28-pin IC socket; and two PSoC sample devices, a CY8C27443 and a CY8C29466. A manual, a CD containing the complete PSoC Designer and PSoC Programmer software, and a USB cable are also included. It is possible to get started immediately with this kit. The printed circuit board includes four LEDs, a pushbutton and a potentiometer. Power can be taken from the USB port via MiniProg1, or the controller (once programmed) can be powered from a separate 5 V supply.



target system, for example using MiniProg (see box). MiniProg is controlled by its accompanying software 'PSoC Programmer' (see **Figure 5**). It is important to select the correct device family (27x43-*XI) and device type (CY8C27443). Set the acquire mode to 'Reset' and then apply power to the device using 'Power Device'. When programming has been successfully completed, set 'Power Device' to off and then on again. The necessary potentiometer is already present on the CY3210 MiniEval evaluation board (see box), connected to the correct input. The sine wave signal will now be present on pin P05, and its frequency can be set as desired between approximately 15 Hz and 260 Hz.

PSoC Express

Using the PSoC Express graphical development tool is possible to develop applications for the Cypress PSoC mixed-signal devices without writing a single line of C or assembler code: the complete program is created automatically. One reason that PSoC Express was brought out by the manufacturer was perhaps that using PSoC Designer is very efficient, but not always completely straightforward. Presumably it was not at the right level for some potential users. The purely graphical interface of PSoC Express can fill that gap, since it is very simple and motivating to use, and results can be achieved with very little effort and in a very short time. There are just four steps to using this new tool.

1. Select the input and output blocks from a catalogue. For inputs this includes such things as temperature sensors, analogue voltage inputs, switches and buttons as well as I/O pins; for outputs it includes MOSFET switches, relays, LEDs and fans.
2. Configure the function of each block using its parameters.
3. A simulation mode allows the design to be verified. Input values can be set using sliders or knobs and the behaviour of the outputs can be observed.
4. In the final step a suitable PSoC device is selected and the hex file containing the programming data is generated.

PSoC Express is of course a specialised tool, which cannot cover all the potential uses of the PSoC family. Rather, it is oriented towards constructing analogue and digital controllers and displays. If, for example, it is desired to indicate a given voltage range using LEDs or turn on a fan at a specified temperature, PSoC is the right tool for the job.

In our example we will build a display for monitoring a battery, indicating voltage over a set range using LEDs. The 'undervoltage' range is below 2.2 V, and is to be indicated by a red LED. The range up to 2.9 V is to be indicated by a yellow/green LED, and the range above 2.9 V by a flashing green LED. The first step is to create an input suitable for measuring from 0 V to 5 V in the work area (see **Figure 6**). The second step requires a 'valuator' of type 'Setpoint Region', in which the necessary threshold values are entered (**Figure 7**). Finally, the outputs are dragged onto the work area. A transfer function must be defined for each output. In each case a 'TableLookup' type is suitable. You can then

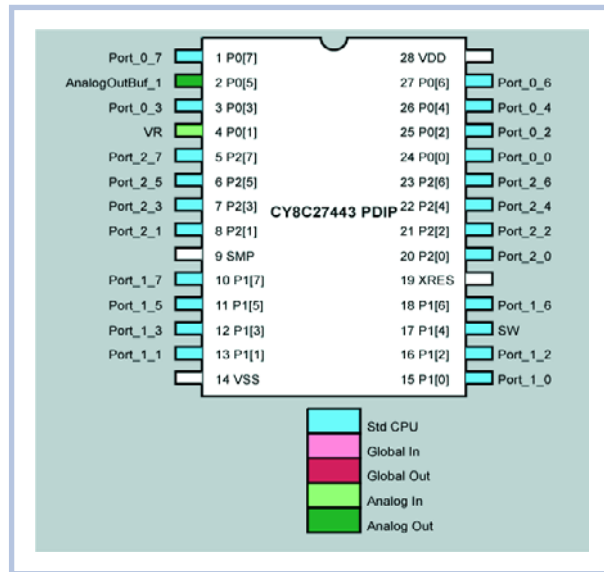


Figure 4. Device Editor gives an overview of the selected pins.

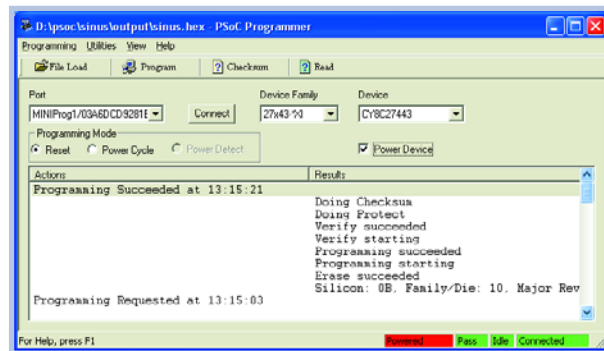


Figure 5. MiniProg (see box) is controlled by the PSoC Programmer software.

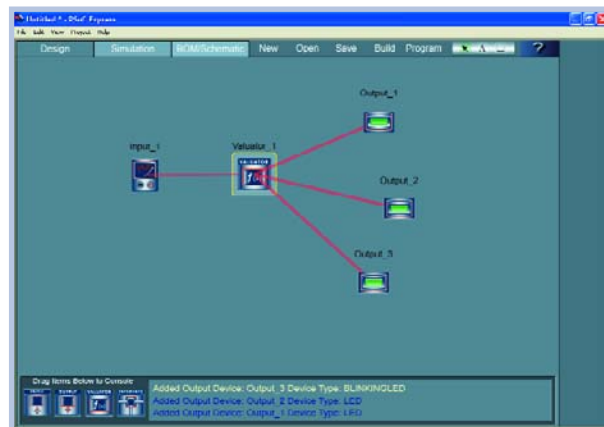


Figure 6. The work area of PSoC Express shows the input and outputs as well as a 'valuator'.

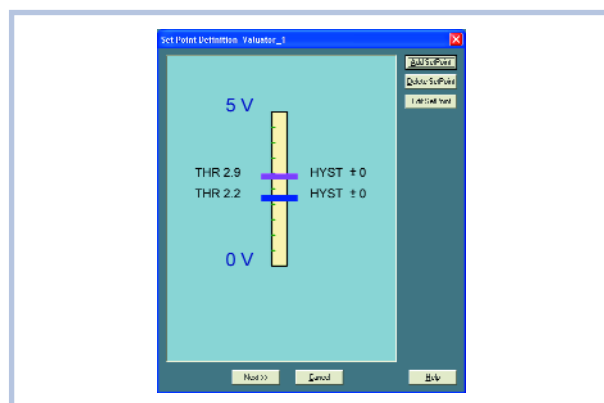


Figure 7. The desired threshold values can be entered into the 'valuator'.

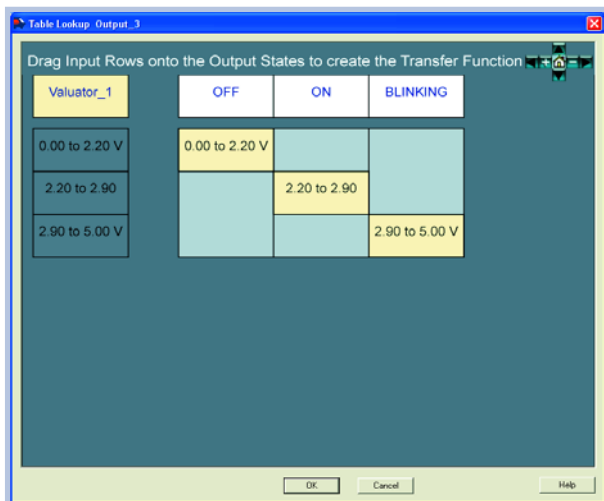


Figure 8.
The transfer function for the green LED is set using a lookup table.

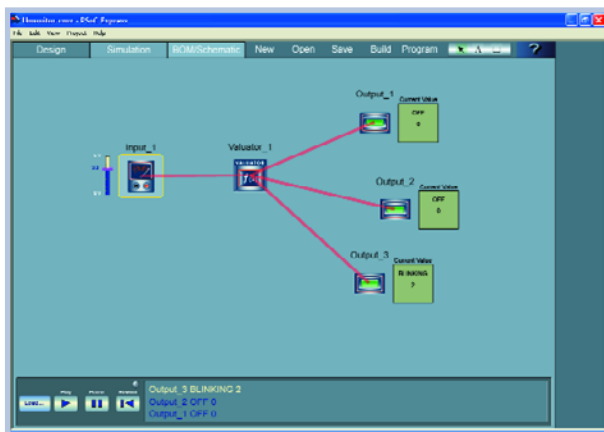


Figure 9.
The completed project can be simulated on the screen.

decide on which input or valuator the output state is to depend, and then define the individual states (**Figure 8**).

No soldering

Programming for the project is now complete, and it can be simulated on the screen, as shown in **Figure 9**. A slider is used to adjust the input voltage and the state of the outputs can be observed.

When the simulation has completed satisfactorily, the project can be built using 'Build'. At this point the controller type needs to be selected. Building the project takes a little time: behind the scenes, PSoC Express is creating a complete PSoC project in C, launching the C compiler in PSoC Designer, and creating an executable program. A circuit diagram and a data sheet with the pinout and a tabular functional description are also produced.

All that is left to do is transfer the code to the target system. We can use the MiniEval board, and control the input using the potentiometer. Output 2 and the flashing output 3 are suitably connected to LEDs on the board, so that operation can be checked without any extra soldering.

The curious user can examine the C project that has been created using PSoC Designer, if for no other reason than to learn a little about C programming and program structure. The kernel of the program is an infinite loop which reads the inputs and drives the outputs.

The program created is always built around an infinite loop which includes all the decision and control processing. It is not possible to add processes that run at specified times or to use special-purpose analogue blocks such as filters. If all that is wanted is a simple permanent linkage between input and output, this interface is, however, entirely adequate. PSoC Express is available for free download from www.cypress.com/psocexpress.

(050220-1)

Sine wave generator output loop

```

loop:
    lcall DELSIG8_1_fIsDataAvailable // if ADC conversion complete then.....
    jz    loop

    lcall DELSIG8_1_cGetDataClearFlag // get ADC result and convert to offset binary
    add  A, 0x80
    mov  [bADCvalue], A // store value for debug watch variable
    cmp  A, 0x0F // counter period less than 0x03 is invalid due to
    jnc  LoadCounter // excessive interrupt servicing
    mov  A, 0x0F

LoadCounter:
    lcall Counter8_1_WritePeriod // update DAC update rate

    jmp  loop

area lit

SINtable: // 64 entry SIN lookup table
db 31, 33, 36, 39, 41, 44, 46, 49, 51, 53, 55, 56, 58, 59, 59
db 60, 60, 60, 59, 59, 58, 56, 55, 53, 51, 49, 47, 44, 42, 39
db 36, 33, 31, 28, 25, 22, 19, 16, 13, 11, 9, 7, 5, 3, 2, 1, 0
db 0, 0, 0, 1, 2, 3, 4, 6, 7, 10, 12, 14, 17, 20, 23, 26, 29

area text
    
```


ELECTRONIC BUILDING

Modular design and construction using E-blocks

Evelien Snel & David Daamen

We often hear from our readers that they don't have enough time to solder all those parts in a constructional project. Or that there are too many complex microcontroller projects in Elektor Electronics, for which you have to be a genius to program them. It is indeed difficult to make modern technology accessible to everybody. But the best comes to those who wait ... join us to discover the numerous possibilities that E-Blocks offer!



E-BLOCKS

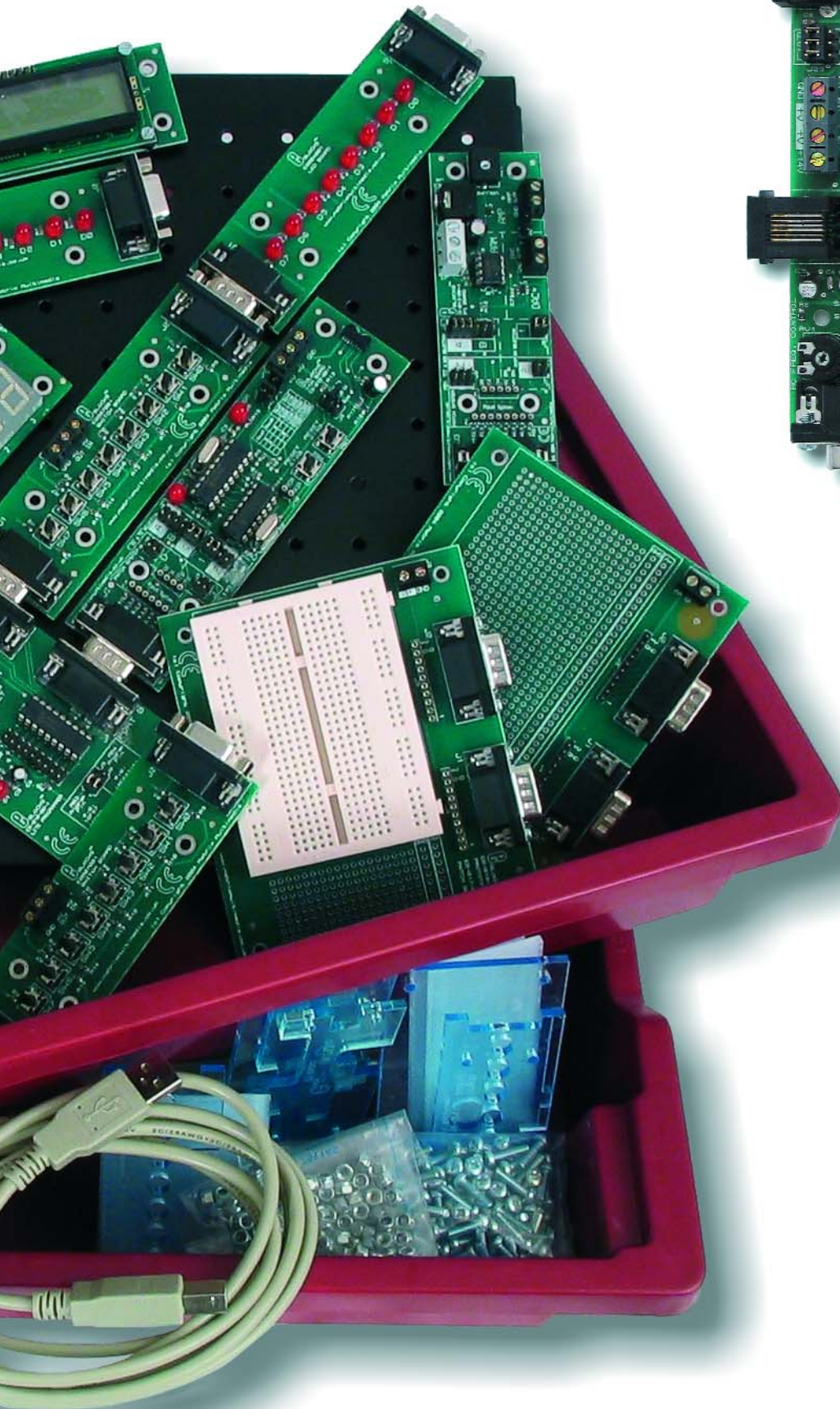


Figure 1. The USB Multiprogrammer is at the heart of the system.

Some time ago the editorial team became aware of E-Blocks. At first we wondered what we could do with them. Some didn't see the point of them, others thought they were fantastic. In due course the balance shifted to the 'yes' camp, as the designers and editorial staff began to admit that 'it was all thoughtfully and well designed'.

But why did people have all these reservations about E-Blocks? Was it due to the fact that the hardware was already completed? Was it because you no longer had to solder? Was it because you 'only' had to program?

Not everybody is convinced yet, and it will take time to get used to the idea. In any case, we decided to let you make up your own mind because there was one thing everybody agreed with: E-Blocks are very handy when you want to design something very quickly, and without much soldering. You can first build a prototype using a handful of E-Blocks, test the design and only then start with the design of the PCB.

But what exactly are these E-Blocks? They are a collection of interchangeable

able boards that can be used to quickly and easily put a microcontroller system together. Apart from several microcontroller boards, there are many other E-Blocks available from the manufacturer, our long-time advertiser **Matrix Multimedia**. There are input boards, output boards and combined I/O boards. Last but not least, a whole range of sensors is provided that can be plugged into the E-Blocks system.

USB Multiprogrammer

Let's take a look at the E-Blocks USB Multiprogrammer (see **Figure 1**). This E-block makes the perfect starting point for your own projects. The Multiprogrammer can use many different types of PICs (see inset) and also has a USB port that can be used to connect the E-Blocks system to a PC. The USB connection is not only there to program the system, but it can also be used with most computers to provide a simple system with power. In many cases there is therefore no need to use a mains adapter. There is of course a socket on the board that takes an external power source for those projects that require more current than the USB port can provide.

Plug and Play

The I/O lines of the PIC are made available along the edge of the board. There is a maximum of five ports, each with a maximum of 8 lines, depending on the type of PIC used. The connections between all E-Blocks have been standardised, making it very easy to combine the parts to form a system. A peripheral is connected to a port of the microcontroller simply by plugging in a board. This distinctive method of construction has many advantages, since in virtually all microcontroller systems the majority of the hardware consists of a combination of standard building blocks. Instead of redesigning these blocks for every prototype and then constructing them, you can now simply pick them out of a spares box. When you have completed your project you can easily dismantle it and use the individual blocks again. Or not of course: the E-Blocks are also perfectly suitable for use in a permanent design. A sturdy metal base plate has been designed for just this purpose. It has a grid of mounting holes, which correspond with the holes present on all E-Blocks.

We should however point out that a

USB Multiprogrammer

The E-Blocks USB Multiprogrammer supports the following PICs:

PIC12F629, PIC12F675, PIC12F635, PIC12F683
 PIC16F627A, PIC16F627, PIC16F628A, PIC16F628,
 PIC16F630, PIC16F648A, PIC16F676, PIC16F684,
 PIC16F688, PIC16F636, PIC16F716, PIC16F72,
 PIC16F737, PIC16F73, PIC16F747, PIC16F74,
 PIC16F767, PIC16F76, PIC16F777, PIC16F77,
 PIC16F818, PIC16F819, PIC16F83, PIC16F84A,
 PIC16F84, PIC16F870, PIC16F871, PIC16F872,
 PIC16F873A, PIC16F873, PIC16F874A, PIC16F874,
 PIC16F876A, PIC16F876, PIC16F877A, PIC16F877,
 PIC16F87, PIC16F88
 PIC18F242, PIC18F248, PIC18F252, PIC18F258,
 PIC18F442, PIC18F448, PIC18F452, PIC18F458,
 PIC18F1220, PIC18F1320, PIC18F2220, PIC18F2320,
 PIC18F2331, PIC18F2410, PIC18F2420, PIC18F2431,
 PIC18F2439, PIC18F2455, PIC18F2510, PIC18F2515,
 PIC18F2520, PIC18F2525, PIC18F2539, PIC18F2550,
 PIC18F2585, PIC18F2586, PIC18F2610, PIC18F2620,
 PIC18F2680, PIC18F2681, PIC18F4220, PIC18F4320,
 PIC18F4331, PIC18F4410, PIC18F4420, PIC18F4431,
 PIC18F4439, PIC18F4455, PIC18F4510, PIC18F4515,
 PIC18F4520, PIC18F4525, PIC18F4539, PIC18F4550,
 PIC18F4585, PIC18F4586, PIC18F4610, PIC18F4620,
 PIC18F4680, PIC18F4681

system built this way with E-Blocks will always be physically larger than is strictly necessary, but the extremely short development time certainly makes up for this.

Programming: frightening or challenging?

Part of the design naturally involves some programming of the hardware. For some of you this aspect of the design is a bit of a challenge that makes the project more interesting. For others this is the main reason not to get involved with a microcontroller project in the first place.

In any case, even those of you who are

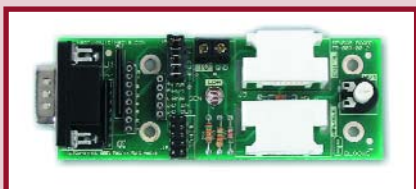
comfortable with microcontroller design are not necessarily expert programmers. Programming is a completely different branch of design compared to electronics. It would therefore be very useful if there was a way that made the programming simpler and also reduced the development time. Matrix Multimedia has designed a programming suite especially for use with E-Blocks that satisfies both of these requirements.

Flowcode

Even if you have no prior experience of programming microcontrollers, you'll find that you will be able to program

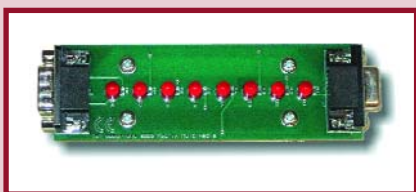


A few of the available E-Blocks modules



Sensor interface board

It's not just the E-Blocks that have standardised connectors; all E-Blocks sensors have them as well. They can be connected to the E-Blocks system via the sensor interface board. This interface board already contains an LDR sensor.



LED board

This board has eight LEDs that are individually driven via an I/O line. When a '1' is written to an output line, the LED will turn on. This board can be linked to another board.



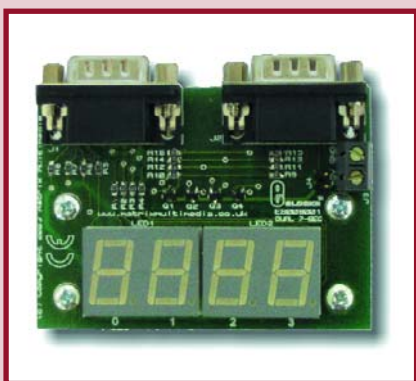
LCD board

This board has an LC display with two lines of 16 characters.



Switch board

This board contains eight pushbuttons. When a button is pressed, a '1' will appear at the corresponding I/O line. This board can be linked to other E-Blocks.



Quad 7-segment display

This board contains four 7-segment displays. The displays are multiplexed, so only 2 I/O ports are required to control the 32 segments.



SPI board

This E-Block has an 8 KB SPI (Serial Peripheral Interface) compatible serial memory chip as well as a D/A converter.

even complex systems with the help of Flowcode. It is literally 'programming with the mouse'. You choose the required functionality from a menu and in this way create a program flow-chart.

When the flowchart has been completed, the Flowcode program can be simulated on the PC. You can then see on the screen how peripherals such as LEDs and LCDs behave. This all happens before a single line has been compiled or assembled and has been sent to the microcontroller. Flowcode also make this process of the design a lot easier. The Flowcode automatically includes all the extra required software routines.

This doesn't mean that all background processes remain hidden from view. Flowcode first translates a flowchart into C, after which another process converts this C code into assembler. At both of these stages you are able to make modifications. Not only can you inspect the code generated, but you can also edit it. It all depends on your preferences: do you want to quickly create a prototype or would you rather program in C, or even assembler? Whichever choice you make, Flowcode won't get in your way. Quite the opposite.

From idea to system

The usefulness of this system is ultimately determined by the possibilities it offers. We could go into detail about the inner workings of the USB Multiprogrammer and which processors are supported. But we feel it's more important to know what you can use it for and have picked a few examples. How about a system that uses mobile phones for remote measurement and control (**Figure 2**)? If you connect a microphone to the A/D converter of the processor you could use it as a wireless baby monitor. With the addition of your own GPS module you could turn it into a vehicle tracking system.

Somewhat simpler, but no less interesting, is the ultrasonic measurement system shown in **Figure 3**. You could use this to take measurements within the home or as a parking aid in your car.

Are you curious about the state of your health? With the help of E-Blocks and Flowcode you can build an advanced PC controlled ECG and heartbeat monitor in a jiffy (**Figure 4**).

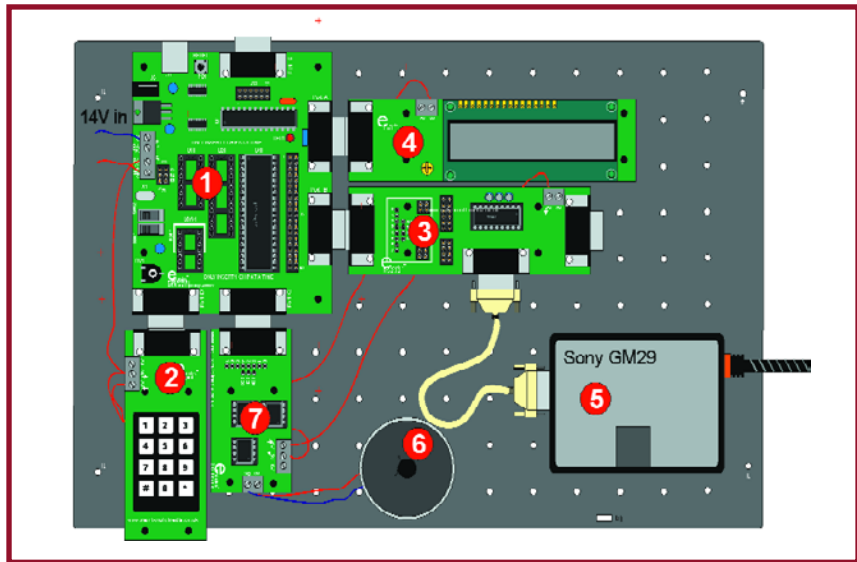


Figure 2. Mobile telephone application - USB Multiprogrammer (1), keypad (2), RS232 board (3), LCD board (4), GPRS unit (5), loudspeaker (6) and SPL board with D/A converter (7).

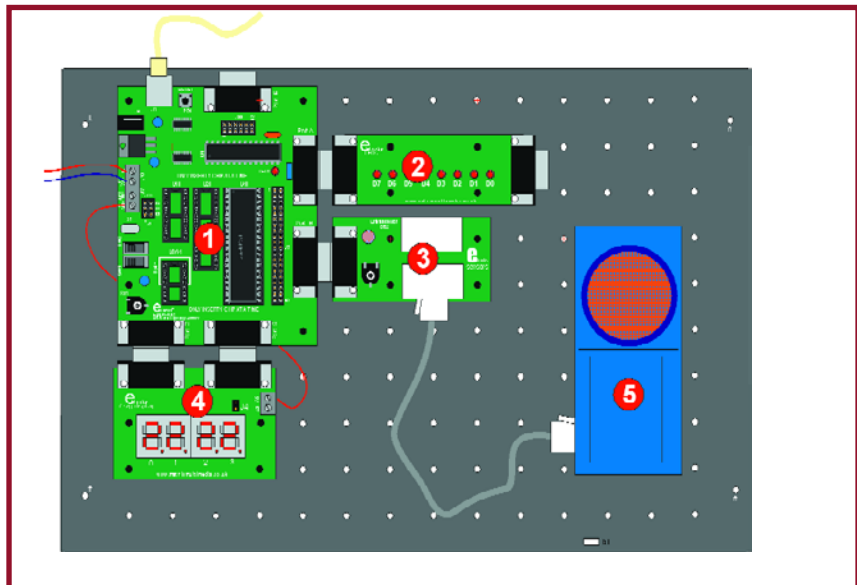


Figure 3. Ultrasonic measurement system - USB Multiprogrammer (1), LED board (2), sensor board (3), 7-segment display (4), ultrasonic sensor (5).

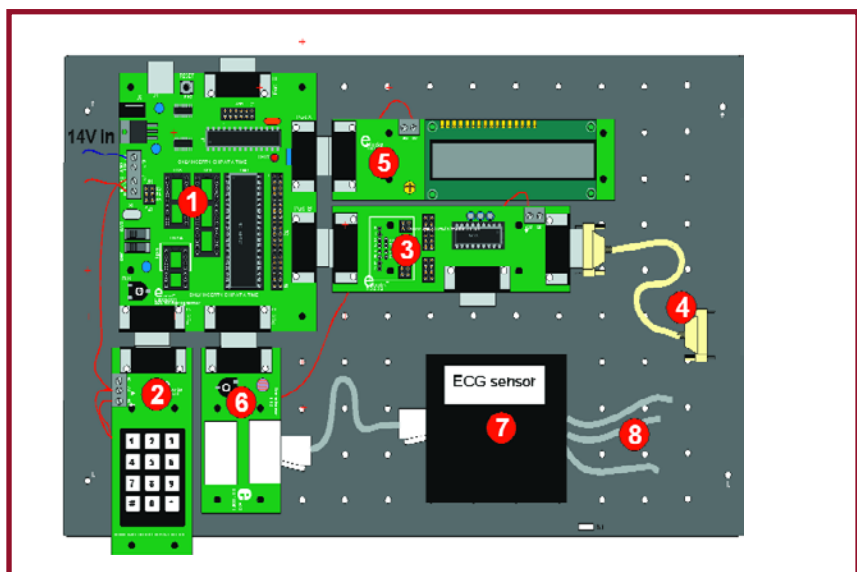


Figure 4. ECG and heartbeat monitor — USB Multiprogrammer (1), keypad (2), RS232 board (3), RS232 serial cable (4), LCD board (5), sensor board (6), ECG unit (7), ECG cables and electrodes (8).

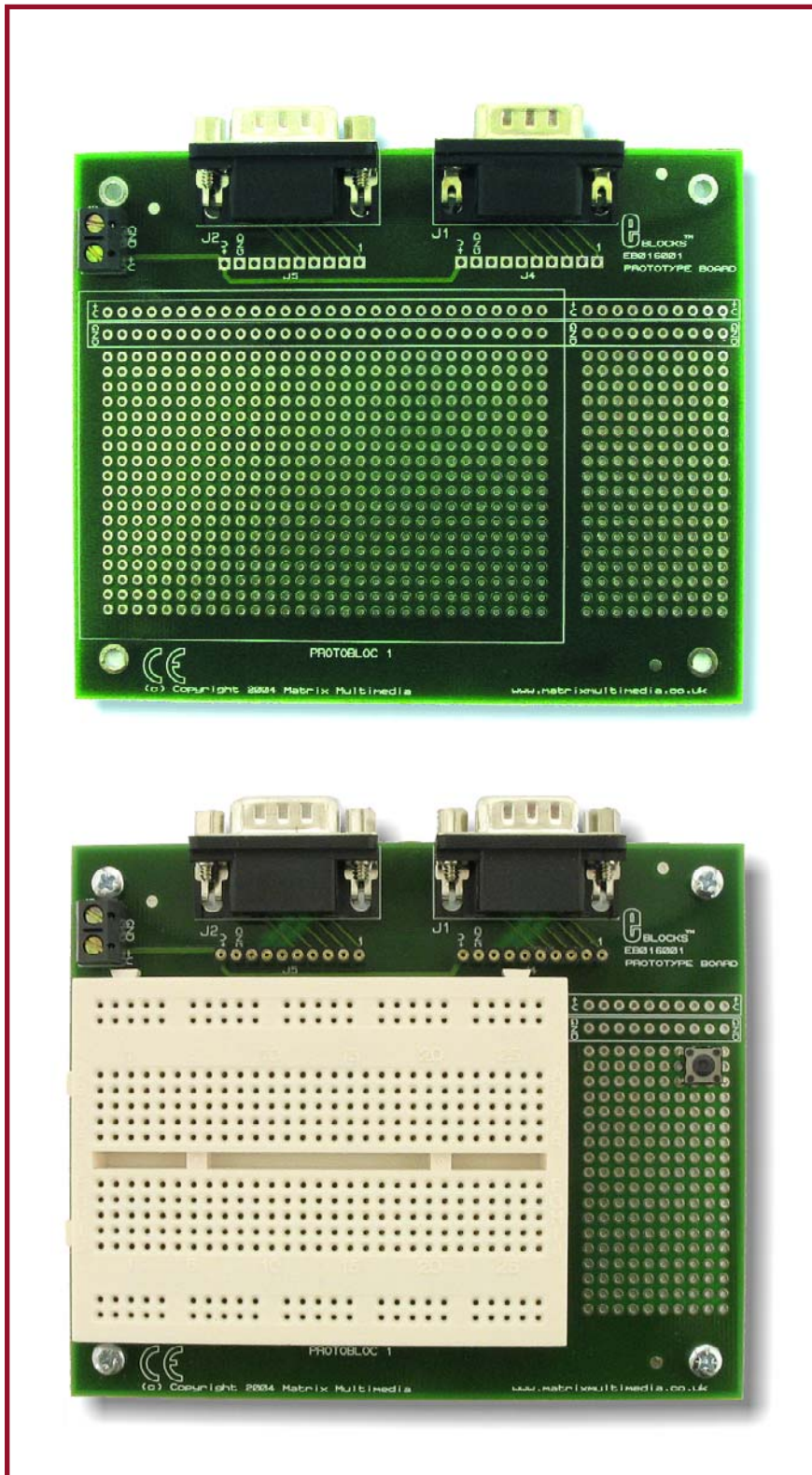


Figure 5. You can still experiment with designs even though you're using standard building blocks.

Sensors

We could continue like this for a while. The possibilities really seem to be endless. One of the biggest advantages of the whole E-Blocks system is that it isn't limited to just the microcontroller boards, displays and pushbuttons, but that a lot of thought has gone into interfacing with the outside world. The communications facilities, for example, include a GPRS unit and an Internet

board. There is also a huge range of sensors available. The list of medical sensors incorporates a respiration monitor, which can be used to measure the lung capacity, and a blood pressure sensor that provides electronic measurements of the systolic and diastolic values. There are also ECG sensors and heartbeat monitors, which can be used to display the rate and shape of your heartbeat.

There are also a number of sensors available that come under the category of process technology. These include a flow rate sensor, oxygen sensor, pH sensor and a gas pressure sensor.

Then there are sensors such as radiation detectors, accelerometers, motion detectors and magnetic field strength sensors.

Sensors that have a more everyday use such as a microphone, temperature sensor and relative humidity sensor are also included in the range.

Hit or miss?

Well, you tell us. We are now convinced that E-Blocks and Flowcode are a magical combination that adds a new dimension to the way that you approach microcontroller design.

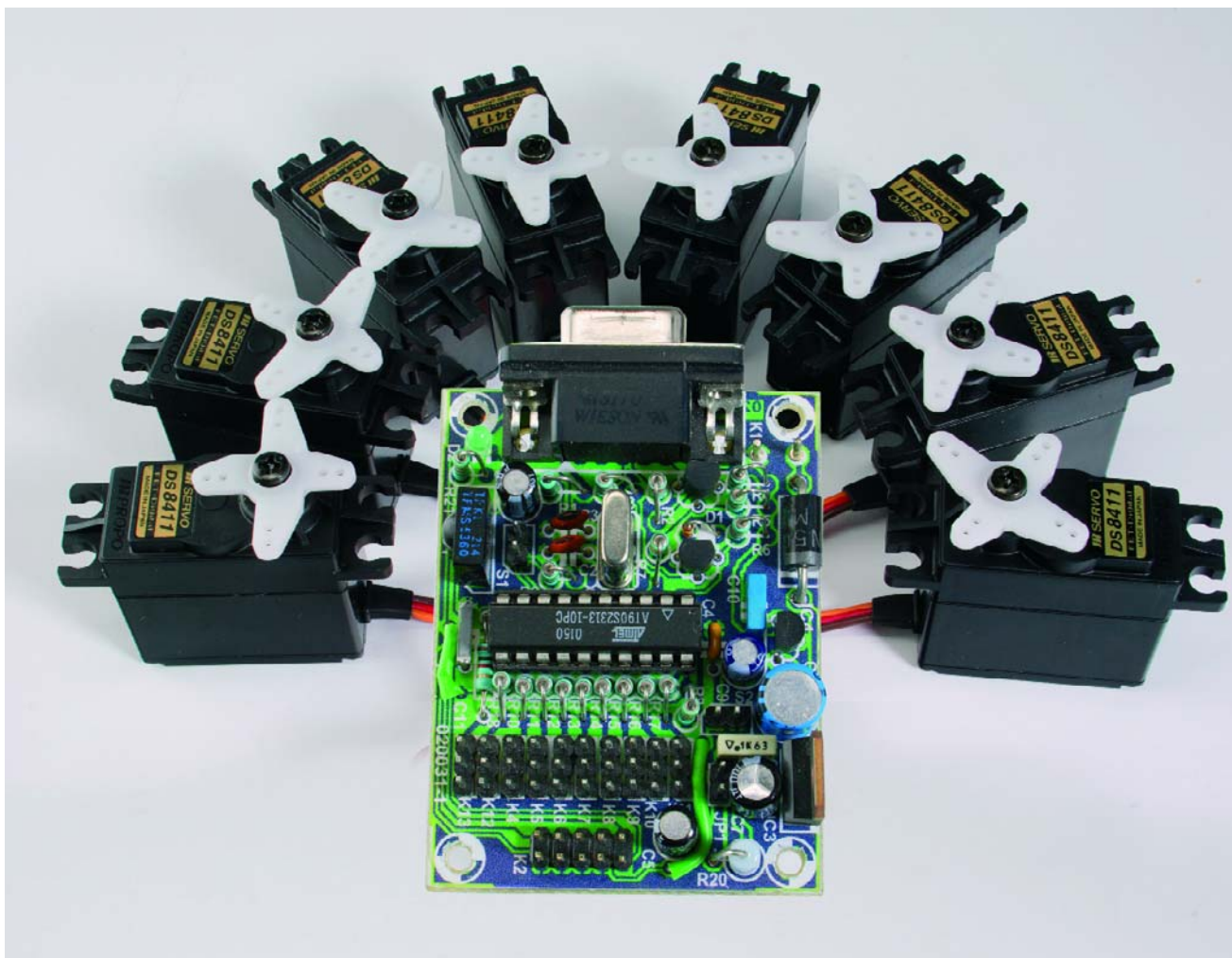
We purposely said it adds a dimension, since it is your own creativity that leads to a unique design. And even here E-Blocks can help you out: there are also two prototype boards available, which have a pair of standard connectors (Figure 5). One is like a standard experimenter's board with a matrix of holes; the other also includes a breadboarding area. These last E-Blocks complement the system perfectly. With only a little soldering and without having to choose individual components or delve into the internal registers of a microcontroller, you can still be very creative in modern electronics.

(054041-1)

The full E-Blocks catalogue, further information and special introductory offers can be found at www.elektor-electronics.co.uk/eblocks

SYNCHRONOUS SER

Willem Ouwerkerk



Model servos are frequently used in robotics projects because of their unique properties. However, a significant disadvantage is that servos often change position far too quickly. In addition, each servo moves at a slightly different speed, which results in erratic looking movements.

VO CONTROL Tracking control with ServoMagic

In these situations it would be nice if all connected servos would move simultaneously and with a user-defined speed to the next position. This synchronous servo control (called ServoMagic) is the solution for this problem. ServoMagic is intended for models and other robotics applications where it is usually undesirable that a servo changes position abruptly. Examples are a robot arm, a walking robot, various parts of a fire fighting ship or warship, etc.

Control principle

ServoMagic uses a Bresenham-like algorithm to move one or more servos in a coordinated (smooth) manner to their new position. Albert Nijhof originally designed this routine. This implementation is an 'unsigned' version.

What does ServoMagic do to change the position of servos synchronously? For each motor it is determined how much it has to travel to the new position. The longest travel (called the number of steps) is the starting point for this algorithm. Each motor will reach its endpoint in that number of steps.

A counter is reserved for each motor. Once the counter reaches zero it is reloaded with the largest number of steps. Each counter is decremented by the number of steps that the servo has to make.

The end result is a reasonably simultaneous movement of the servos. Each servo also reaches its desired final position, as is shown in the example below. An asterisk indicates that the corresponding servo has changed position and a hyphen indicates that nothing has changed.

The following table assumes the original implementation for the explanation of the algorithm.

Initial position	0	0	0	0	0
Counter init.	5	5	5	5	5
Step no.	s1	s2	s3	s4	s5
1	-5*	2-	-2*	4-	3-
2	-5*	-1*	1-	3-	1-
3	-5*	6-	-6*	2-	-1*
4	-5*	3-	-3*	1-	7-
5	-5*	0-	0-	0-	5-
6	-5*	-3*	-7*	-1*	3-
7	-5*	4-	-4*	8-	1-
8	-5*	1-	-1*	7-	-1*
9	-5*	-2*	2-	6-	7-
10	-5*	5-	-5*	5-	5-
Endpoint	10	3	7	1	2

The example above has a starting position of 0 0 0 0 0 and a finishing position of 10 3 7 1 2. From the table it can be seen that the final position will be reached in ten steps. When a counter becomes empty (negative) it is reloaded by adding the maximum number of steps (10 in this case).

All counters are initialised with half of the maximum number of steps, in this example that is 10/2=5.

The counter for Servo 1 (s1) has to be reloaded on each step and as a result this servo changes position ten times. The counter for Servo 5 (s5) changes position only twice, because the counter underflows on just two occasions.

In this way, the servos can be controlled either individually or simultaneously, with or without speed reduction, but always in coordinated fashion.

The hardware

The most important part in the circuit of **Figure 1** is an AT90S2313 from Atmel, an 8-bit RISC microcontroller, which, in this case, is clocked at a frequency of 4 MHz. This controller is provided with 2 kB flash memory, 128 bytes of RAM, 128 bytes of EEPROM and 15 I/O ports.

The remaining hardware is minimal.

For communicating with the ServoMagic module you can choose between either a simple RS232 port, which is built with a couple of transistors (T1/T2) or a 36 kHz infrared receiver (IC1) that can be connected to K3 (but not both at the same time!). Finally there is a double power supply.

A precautionary remark regarding the power supply: note that the AVR chip used here does not have a built-in brownout detector. If the voltage rises too slowly on power up, it is possible that the reset procedure of the microcontroller is not properly performed. The consequences are usually a few altered bytes in the internal EEPROM. Atmel themselves also warn about this. If necessary, fit an external brownout-detector. The successor type ATtiny2313 does have an on-chip brown-out detector — something to consider for a 'mark2' version of the circuit!

On the PCB there are also several connectors for the servos, the ISP programming interface from Atmel, a reset pushbutton and a pushbutton for demo mode. There is also room

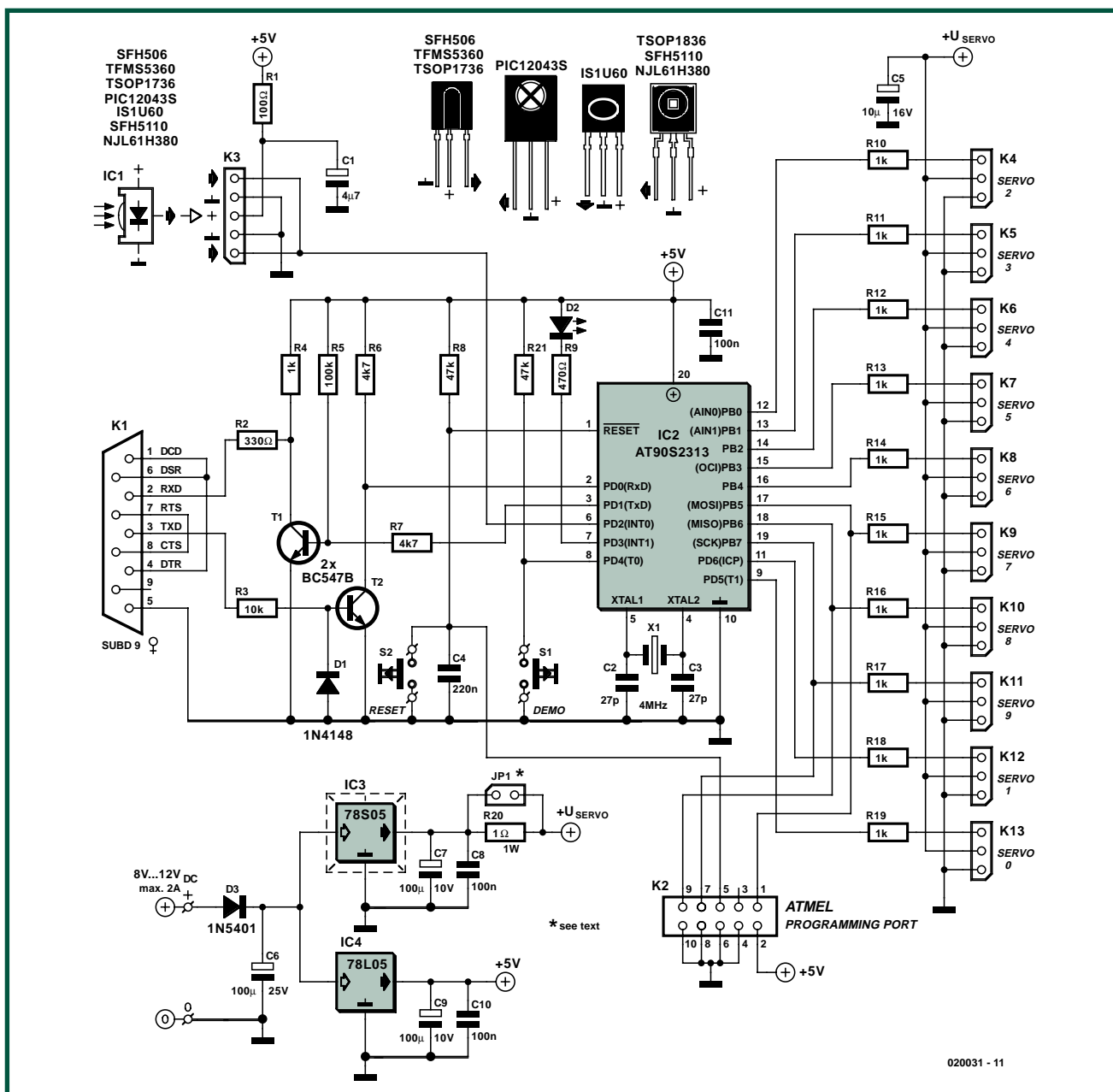


Figure 1. The circuit consists mainly of an AT90S2313.

for a right-angle female sub-D 9-way connector for the RS232 connection and also for IC1, the infrared receiver. Voltage regulator IC3 (a 78S05) will need to be replaced with a 78S06 if the servos are expected to deliver a little more power. If a lot of power-hungry servos are connected, an even bigger regulator such as the 78T05 (3 A) may be required. Don't forget to fit this regulator with a suitably sized heatsink.

Details

Each type of servo has its own begin and end position at a different pulse

width. The begin and end positions for all the servos are stored in a table in the internal EEPROM of the AT90S2313.

To make sure that each servo can be controlled with the same data (0 to 180), the calculated pulse width for each servo is scaled separately. In this way all servos can be used over their maximum range, but the programmer does not need to remember the actual range for each individual servo. The basic calculation is as follows:

$$\left(\frac{(\text{EndPosition} - \text{BeginPosition}) \cdot \text{SetPosition}}{\text{Limit}} \right) + \text{BeginPosition}$$

The result of this calculation is stored inverted in a table in RAM. The timer interrupt uses this number as the basis to generate the desired pulse width for each servo.

The EEPROM space has been allocated as follows:

- **Address 0:**
not used (Refer Atmel errata).
- **Address 1:**
initial contents for WAIT (step time).
- **Addresses 2 to 11:**
minimum pulse widths for all servos.

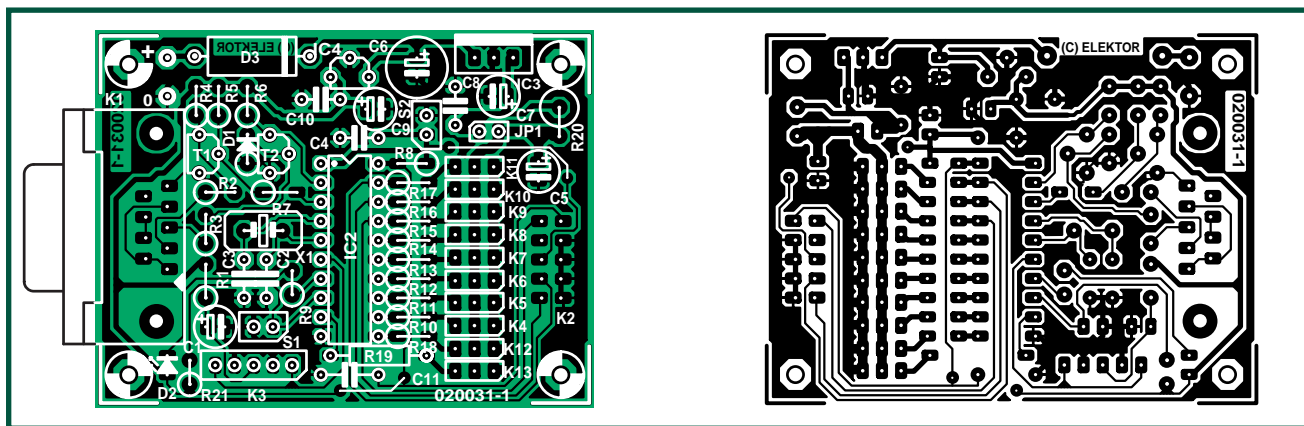


Figure 2. The PCB has been kept quite small to simply installation. Don't forget the two wire links when assembling!

COMPONENTS LIST

Resistors:

R1 = 100Ω
 R2 = 330Ω
 R3 = 10kΩ
 R4, R10-R19 = 1kΩ
 R5 = 100 kΩ
 R6, R7 = 4kΩ
 R8, R21 = 47kΩ
 R9 = 470Ω
 R20 = 1Ω 1W

Capacitors:

C1 = 4μF7 10V radial

C2, C3 = 27pF

C4 = 220nF
 C5 = 10μF 16V radial
 C6, C7, C9 = 100μF 25V radial
 C8, C10, C11 = 100nF

Semiconductors:

D1 = 1N4148
 D2 = LED, 3mm, green
 D3 = 1N5401
 IC1 = IR receiver (see text)
 IC2 = AT90S2313-10PC, programmed for RS232, order code **020031-41**
 IC3 = 78S05
 IC4 = 78L05
 T1, T2 = BC547B

Miscellaneous:

JP1 = jumper
 K1 = 9-way sub-D socket (female), angled pins, PCB mount
 K2 = 10-way pinheader
 K3 = connection for IR receiver
 K4-K13 = 3-way pinheader
 S1, S2 = pushbutton, 1 make contact
 X1 = 4MHz quartz crystal
 Heatsink for IC3, e.g., FK237, ICK35
 2 wire links
 RS232 cable, straight-through (all wires 1:1 connected)
 PCB, no. 020031-1 from The PCBShop Disk, project software, order code **020031-11**

– Addresses 12 to 21:

maximum pulse widths for all servos.

– Addresses 22 to 121:

ten rows of ten locations each, where preset positions for the servos can be stored. The first row is used as the rest or initial position for the ten servos.

Construction

Begin by fitting onto the PCB, shown in **Figure 2**, all the low-profile passive parts. Start with the wire links, followed by the resistors and diodes. If you are using an IC socket, it has to be soldered on the board before mounting the upright resistors. Continue with the transistors, capacitors and crystal, followed by the male header connectors for the servos (connections for Graupner/JR-standard) and JP1, then K2 and K1, IC4 and IC3. IC1 is optional if you prefer to use the infrared version of the soft-

ware (the *Elektor Electronics* supplied pre-programmed controller is suitable for RS232, the source code and hex file are available for the IR version). Finally connect the two pushbuttons for resetting and activation of the demo mode.

Software

The complete AVR ByteForth software can be downloaded from the *Elektor Electronics* website, filed as no. **020031-11**. The file contains several modules.

A free demo version of the compiler can be obtained from <http://\forth.hccnet.nl>.

Forth is often considered to be outdated. It is a limited language but is capable of getting the most out of many a modern microcontroller. The multitasker is capable of running three tasks on an AT90S2313, try it! First, the library file AVR-EFPFRT is

loaded containing the commands (called words in Forth) that read and write to the internal AVR EEPROM. After that the pseudo-random generator library is added as well as modified versions of the RS232 and number conversion libraries.

Once all the declarations have been carried out, the timer-interrupt routine follows, which, with variable frequency (from 40 Hz to 125 Hz), provides the servos with neat pulses.

The interrupt routine reads the pulse widths from a table that is stored in the RAM section of the AVR. The 8-bit timer-0 was selected, because in conjunction with the built-in prescaler, it can easily generate pulses between 0.02 ms and 4 ms.

The data for the routine requires five tables of 10 bytes each. The interrupt routine itself also has its data in a table, which is 10 bytes long. All this fits in the small amount of available RAM of only 128 bytes.

Based on the contents of two arrays, the data is calculated for the move-

RS232-command set

ServoMagic is controlled via RS232 and uses ASCII commands for this. Once a command has been processed, the prompt returns consisting of the following characters: ^M ^J > space

Character	Name	Instruction	Instruction string	Reply
?	Question	Print servo position	?	The position is
1 2 3 4 5 6 7 8 9 10 11				Prompt
A	Adjust	Set new position	A 2 44	Prompt
B	Begin	Set begin position	B 3 62	Prompt
D	Demo	Servo Magic demo	D	Prompt
E	End	Set end position	E 3 125	Prompt
G	Go	Go to EEPROM position	G 0	Prompt
H	Hold	Save WAIT in EE	H	Prompt
M	Move	Move several servos	M	Prompt
N	Note	Note position in EEPROM	N 1	Prompt
O	One	Move 'one' servo	O 3 40	Prompt
P	Put	Move all	P 1 2 3 4 5 6 7 8 9 10	Prompt
R	Rest	Set rest position	R 5 94	Prompt
S	Speed	Move servo fast	S 7 80	Prompt
W	Wait	Set waiting time	W 10	Prompt

Here follows a more detailed description. The data indicated with 's' is always a number from 0 to 9, data 'n' differs for each command.

? Display the present position of the servomotors, followed by the wait time. The position of the servomotors is a number from 0 to 180. The wait time is a number between 0 and 255 milliseconds.

A s n Set the new position 'n' for servo 's' (a number from 0 to 180). Servo 's' will move in small steps only after the M-command is received.

B s n Set the begin position 'n' for servo 's'. The number 'n' has to be multiplied by 16 μ s in order to generate the desired pulse width. The default is 62 ($\times 16 = 992 \mu$ s). Servo 's' is a number from 0 to 9.

D Set ServoMagic to demo mode. Every 250 ms, the servos are instructed to a new position by the random generator.

E s n Set the end position 'n' for servo 's'. The number 'n' has to be multiplied by 16 μ s to generate the desired pulse width. The default is 125 ($\times 16 = 2000 \mu$ s). Servo 's' is a number from 0 to 9.

G n Go to the servo position stored at location 'n' in EEPROM. There are up to 10 stored positions in EEPROM where 'n' can be from 0 to 9. Location zero corresponds to the rest position. The waiting time after each step has the value given by the W-command. The initial value is 10 ms.

H Copy the waiting time to EEPROM, this value is retrieved at start-up and used as the default value for the waiting time.

M Set the servos indicated with the A-command to their new position. The waiting time after each step has the value set by the W-command and is initially 10 ms.

N n Store current servo position at location 'n' of the EEPROM. Up to 10 positions fit in EEPROM, so 'n' can be from 0 to 9. Location zero corresponds to the rest position.

O s n Set servo 's' in position 'n' (a number from 0 to 180). Servo 's' moves in steps. The waiting time after each step has the value given by the W-command. The initial value is 10 ms.

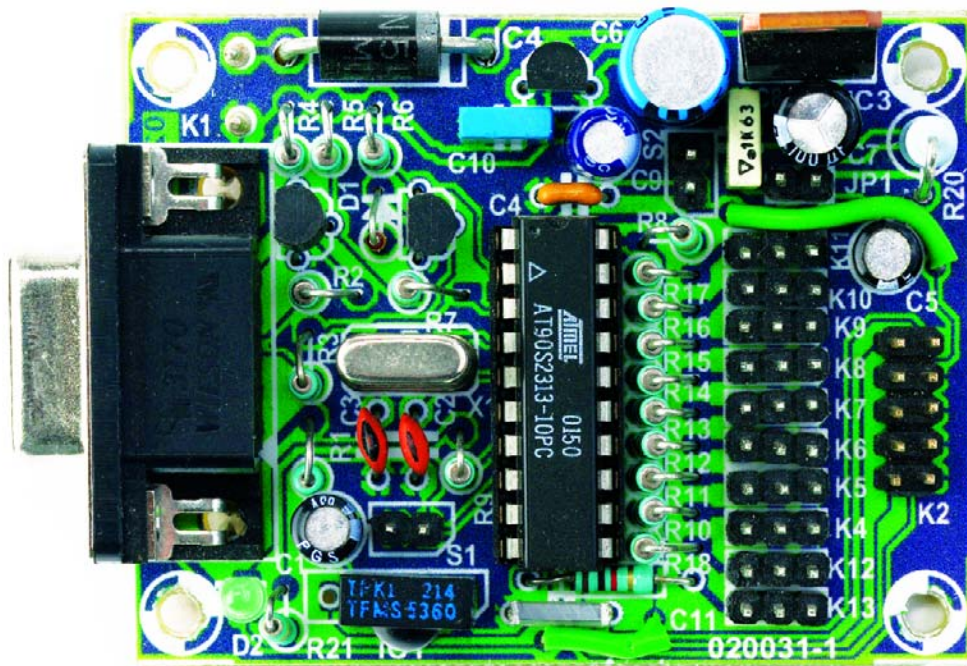
P n1 n2 n3 n4 n5 n6 n7 n8 n9 n10

Set all servos 'n1' to 'n10' in steps, as instructed by the W-command. This is initially 10 ms. The numbers 'n1' to 'n10' range from 0 to 180.

R s n Set a new rest position (starting position) for servo 's'. The number 'n' ranges from 0 to 180 (90 is centre position).

S s n Set servo 's' in position 'n' (a number from 0 to 180). Servo 's' moves in steps. There is no waiting time after each step.

W n Set new waiting time.



ment of the servos.

Finally there is the command-interpreter, which receives and processes its instructions received via RS232 or RC5. The baud rate and other settings are defined by default as 9600 baud, eight data bits, no parity bit and one stop bit. The RS232 command interpreter is described separately a little further on.

When the RC5 version is used, an RC5 decoder runs in the background. The decoder is started by a falling edge on PD2, which causes a hardware interrupt. This hardware interrupt starts a timer, which samples the incoming signal on PD2 and checks it for correctness.

Testing

Connect an RS232 cable (of the 'straight-through' type with 1 to 1 wire connections; i.e., not a null-modem cable!) and start a terminal program on your PC. Configure it for 9600, 8, N, 1 and connect a power supply. Note that no servos are connected as yet. If everything is correct, a start-up message appears at the top of the window and the LED lights up briefly. A prompt appears below the message. Type a question mark '?'. ServoMagic will respond with the current position of the servos. If this works, then the board is functional and the software is running. Remove the power supply voltage from the board and connect a

servo to K6 (servo 4). Re-connect the power supply and ServoMagic starts again; the servo should now be in approximately its centre position.

Now type: O 4 180 and then O 4 0. The servo will first go to the maximum position followed by the minimum position. Everything now functions as it should. You can now connect additional servos. Note that for demonstration purposes, the first two servos have a different begin and end position.

Once you are confident that all ranges are correct you can try the demo mode. In this mode, the random generator drives all servos to ever changing positions. The random generator also controls the speed.

Calibration

The PCB is fitted with jumper JP1 in order to easily calibrate each servo for its maximum range. Remove the jumper when calibrating and connect a voltmeter across the pins.

Now measure the voltage across R20. Once the servo reaches its limit (locks up), the voltage across the resistor will increase. You now know that that selected begin or end position is outside the mechanical range. In this way, with some 'trial and error', you can determine the full range of the servo. Maintain a small margin so that during normal use the

servo will not lock up. Repeat for each servo, one at a time. Once this is all done, a number between 0 and 180 can be used to control each servo to its full limits.

Final notes

With the controller that has been used here, a machine that contains up to ten servos can be controlled. If a larger number is required, use two ServoMagic modules or use an AVR with more I/O, RAM and EEPROM. For example, an ATmega8515 can easily drive 24 servos without any problems.

There are two versions of ServoMagic. The basic version (SMAGIC1A) employs RS232, an alternative version (SMAGIC1I), RC5. *Elektor Electronics* only supplies the RS232 version. The 'IR' version you will have to program yourself.

Both versions can be built on the same PCB. In that way, a PC, SBC or other computer can remotely control one or more machines with or without wires. The RC5 command set is fully documented in the source code file.

Note that in addition to the flash hex files there is also a file EE.hex, which needs to be programmed in the EEPROM section of the controller (this is only necessary, of course, for those who program their own controller).

(020031-1)

DELPHI FOR ELECTRONIC ENGINEERS

Part 9

The Delphi Stamp as a standalone controller

Detlef Overbeek, Anton Vogelaar and Siegfried Zuhr

The Delphi Stamp is a complete controller about the size of a large postage stamp and is based on the ATmega128 microcontroller made by Atmel. Last month we looked at the capabilities of this module. In this instalment we continue programming the Delphi Stamp under Delphi. Here we use it in a standalone mode where it provides the functions of a sprinkler system¹.

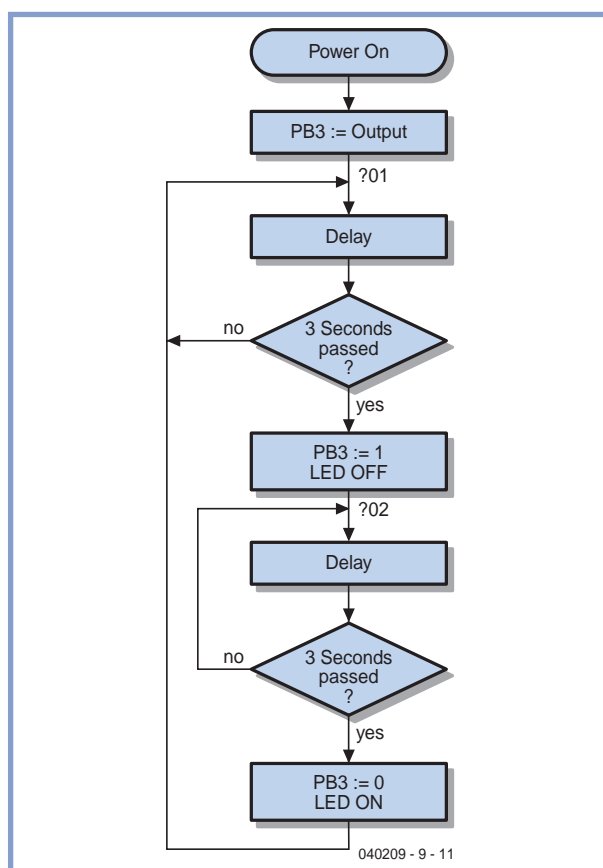


Figure 1. Flowchart for a simple LED flasher program.

The programming language used to directly program microcontrollers is assembly language (this programs the microcontroller at the lowest possible level). However, writing and debugging an assembly language program takes up a lot of time. Although it's certainly possible to use the Delphi Stamp as a development platform for assembly programs, the software tools provided let you write and test programs in Delphi, which significantly simplifies the development of the firmware.

Several templates have been included that make it easier to start writing programs. In the first example we will write an assembly program. As is usually the case, the first step is the drawing of a flowchart, which gives an overview of the functionality of the program (**Figure 1**). In this example we make an LED flash: 3 seconds on and 3 seconds off. The LED is connected via a series resistor to +5 V and Port B, output 3. The controller turns on the LED by making the output low. It is done this way because the ATmega128 can sink more current than it can source.

The ports of the microcontroller can be used as either inputs or outputs. To configure a port pin as an output a '1' has to be written to the relevant position in the Data Direction Register (DDRB). A '1' or '0' in the relevant bit of the data register PortB determines if the output is high or low. The conversion of the flowchart into code begins with a copy of the template file UMain.pas.

The assembly program uses instructions from the instruction-set for this particular microcontroller. The meaning of the instructions we've used is shown below:

1. Please note that several areas in the UK have a water shortage and a hosepipe ban may be in force. (Editor)

SBI Set a bit to '1' in an I/O register
 DEC Decrement the value in a register by '1'
 BRNE Branch if Not Equal: jump to the specified address only if the result of the last operation was not equal to zero
 SBIW Subtract Immediate from Word: reduce the value of two combined registers
 CBI Clear a bit to '0' in an I/O register

The program UMain.pas is compiled by the PasAVR cross-compiler and is then uploaded and run in the Delphi Stamp by the monitor program Mon485.

```
Link $000,$180 // AtMega128 : Code $0000..$FFFF [W]
// RAM $0100..$10FF
// EePRom $000..$7FF
```

```
Program Main;
(* Delphi Stamp. Template for writing assembly programs
Provided by Vogelaar Electronics, Bunschoten NL
Rev 0.10 23-08-05 Initial release *)

(* ===== Start vectors ===== *)

{#P LLChar ?}
{#P ABS 0}

Begin
  Asm
    SBI...DDRB,3      ; PB3 := Output
?01...DEC...R29      ; 3 Sec Delay
    BRNE..?01
    SBIW..R30,1
    BRNE..?01
    SBI...PortB,3    ; PB3 := 1, Led Off
?02...DEC...R29      ; 3 Sec delay
    BRNE..?02
    SBIW..R30,1
    BRNE..?02
    CBI...PortB,3    ; PB3 := 0, Led On
    RJMP..?01
  End;
End.
```

Sprinkler system

For our next project there are really two programs that need to be written: one that runs on the PC and another that runs on the Delphi Stamp itself (see the overview in **Figure 2**). They perform the same functions and are both written in Delphi. Hence the name for the controller board, a microcontroller that is programmed via Delphi, or strictly speaking the Delphi IDE. The units have been designed to perform a number of functions. These functions are the same in both projects, but the way they work has been adapted for the hardware. For example, the unit *Drivers* for the Delphi Stamp deals with the inputs and writes information to the outputs. The unit for the PC has the same name, but takes its inputs from the buttons and trackbars that are shown on the screen. The outputs are represented by virtual LEDs and text on the screen. The unit *UMain* also plays an important part, even though it contains very little: just one timer. It does have a very important function since it simulates what happens inside the Delphi Stamp when it starts up, which is the initialisation of the I/O and generation of a clock that periodically calls a subroutine. This pro-

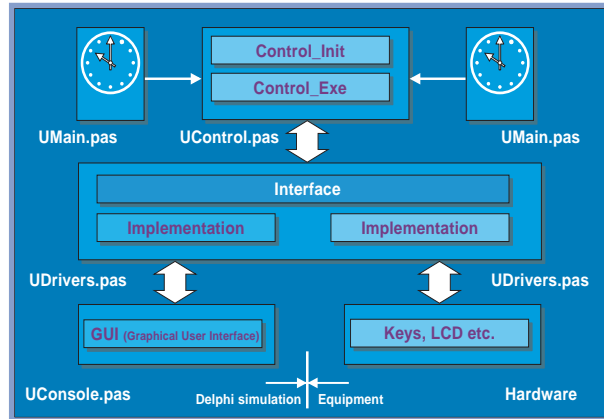


Figure 2. The project consists of two programs: one runs on the PC and the other runs on the Delphi Stamp.

vides the same functionality to the PC that the Delphi Stamp has internally. For the *GUI* unit this comparison no longer holds; this unit exists purely to provide a visualisation of the outputs that are driven by the unit *Drivers*. This makes it the counterpart of the actual inputs and outputs of the Delphi Stamp, which are the switches, potentiometer or analogue input, an LCD display and the LEDs. And finally there is the most important unit of all: *UControl*. This contains all the logic for the project and is tested on the PC. Once this unit has been debugged and it functions reliably it can be transferred to the 'Delphi Stamp project'. In here the units are compiled by PasAVR, which converts the Pascal files into a machine language file that can be stored in the Delphi Stamp. This then has the same functionality as the Delphi program on the PC. In this way Delphi has become an important part of creating a project for a microcontroller. The project is developed and tested in Delphi after which the core program (the unit *UControl*) is transferred into a new 'environment' where it can finally perform its tasks.

For the development of projects for the Delphi Stamp the units *UMain*, *UDrivers* and *GUI* need to be created once only; the code in *UControl* can then be modified for other projects. Should the hardware of the Delphi Stamp change, this can easily be replicated by extending or modifying the units.

So much for the theory, now for some practical matters. Once we know what our goal is we first have to specify the design and then create and test a simulation in Delphi. For this instalment we thought about an automatic sprinkler system for the garden (this will be handy for next summer, and you can experiment with it throughout the winter). This requires only a little extra hardware to implement on the evaluation board. For example, we could control three solenoid valves. We would like to be able to turn them on individually for specified times. It would also be useful to sprinkle the garden for short periods of time and to repeat this process several times. This gives the soil a better chance to absorb the water.

We require three outputs for the solenoid valves, which can be simulated on the evaluation board by LEDs. For the control we need two pushbuttons, which can be used with an easy menu structure. One of the buttons is already present on the board and the other can be con-

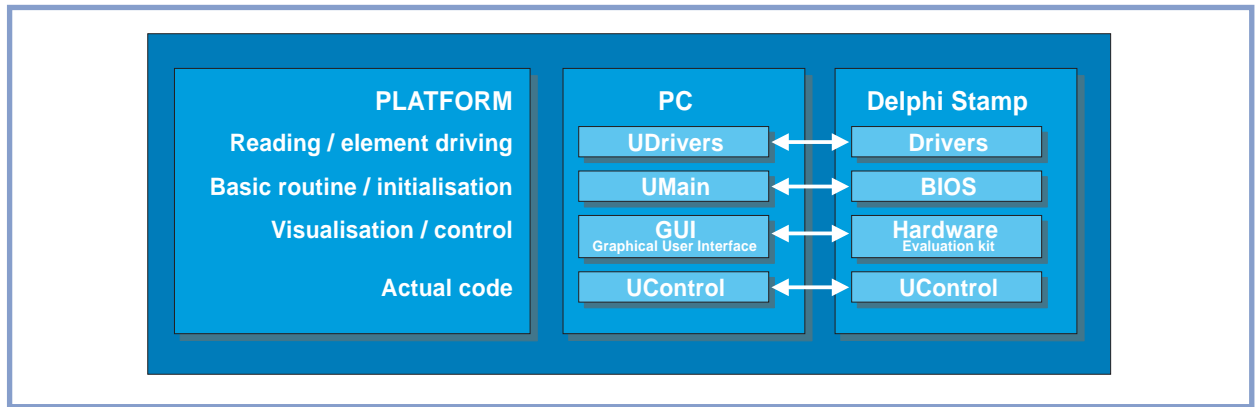


Figure 3. The relationship between the different units for the PC and Delphi Stamp.

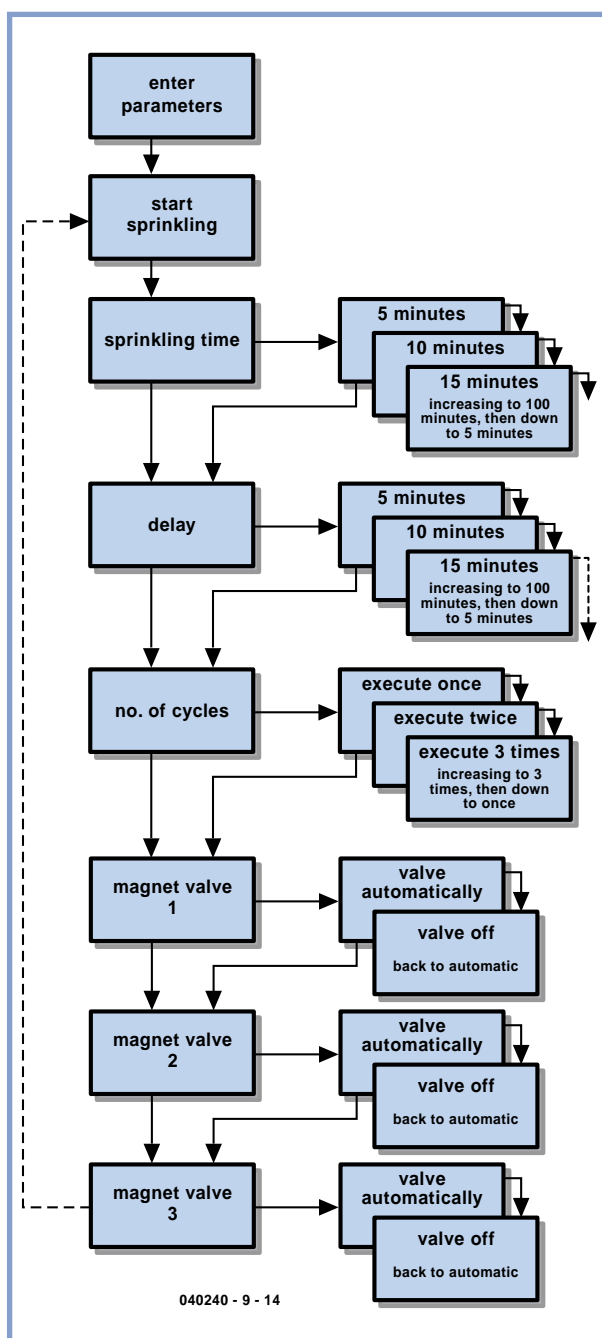


Figure 4. Part of the flowchart for the sprinkler system.

ected to the second analogue input of the board. For the menu and the display of events we'll use the LCD display on the board, which can show a 16-character text line.

We'll use the included *Start Delphi* program to create the simulation. This contains a ready-made layout of the evaluation board in Delphi and includes the required units and routines to make use of all the parts.

You should copy the contents of this directory into a new one, where the design will take place. (After copying the files you should delete the file with the extension '.dsk' because this contains the paths of the files, which would still point to the original directory. After starting Delphi this file will be recreated with the correct path to the new directory.)

In the unit *UMain* we can see that the main parts of the Delphi Stamp are simulated.

The initialisation of the Delphi Stamp happens when the data module is created. In a real Delphi Stamp this normally occurs during the start up. There is also the internal timer, which calls a subroutine (the procedure *ControlExe*) every 100 ms. This is part of the unit *UControl*, which we'll expand in a moment.

The unit *UDrivers* contains the code to simulate the hardware of the Delphi Stamp. The procedures and functions in this unit are used to read the inputs that are entered via the screen and to display the outputs generated by the Delphi Stamp on the screen.

The unit *UGUI* is the graphical representation of the evaluation board and has code that links to the unit *UDrivers*. There is no need to change anything in the above-mentioned units; everything has been designed so that it's ready to be used now and can be reused later.

The core of the project can be found in the unit *UControl*.

The procedure *ControlInit* initialises some of the variables used. On the first line is *!olnit*, which initialises the Delphi Stamp Simulation. Following this are our own variables. We have come across the procedure *ControlExe* earlier in the unit *UMain*, where the timer called this procedure every 100 ms. We have added some code to this procedure to check if some event occurred (e.g. a button pressed) or if a countdown timer has completed.

The function *GetButton* is called to check if a button has been pressed. This returns the value *True* when a button has been pressed.

The following code makes sure that the procedure *Btn1Pressed* is executed once only:


```

If GetButton
Then // PushButton1 is pushed down.
  Begin
    If Not Btn1Last Then
      Btn1Pressed; // do the actual work.
      Btn1Last := True
    End
  Else Btn1Last := False;

```

The procedure Btn1Pressed does all the actual work. The same applies to the second button except that the Analog2 input is read, and then checked to see if it has a high or low level. The procedure Btn2Pressed will then be called if necessary. These pushbuttons are used to work our way through the menu (Figure 4).

Button 1 is used to step through all the menu entries. When we reach the end of the menu it will jump back to the beginning (see var. MenuLevel).

With **Button 2** we can open a menu item and then change its value with Button 1. When the program has just been started we use LED 2 on the Delphi Stamp to indicate this.

After pressing Button 1 once we enter the normal menu and LED 2 will flash slowly. When Button 2 is pressed to enter the edit mode, LED 2 will flash quickly.

The periods can be set anywhere from 5 to 100 minutes in steps of 5 minutes. When the value goes past a 100 it will jump back to 5. The number of repetitions will also jump back to 1 after the maximum value of 3 has been reached.

The solenoid valves can be controlled automatically or turned off.

The settings are stored in an array that has the same structure as the menu. This configuration makes it simpler to expand the menu at a later date.

To show some text on the LCD displays we use separate procedures, which can be reused to show the various values. The value currently shown on the display is stored in a separate variable that is copied into the array of settings when the mode changes.

Pressing Button 2 from the menu item 'Start sprinkler' starts the process. Those solenoid valves that are set to 'Automatic' will then be driven for the selected sprinkling time.

When the same process happens in a program more than once we can save many lines of code by putting the code for this process into a functional routine. An example of this can be found in StartCycle. When enough time has passed, the routine NextValve is called, which looks for the next active valve. SetValve is then used to turn on the valve, or if a parameter of '0' is given, turn off the valve.

After the last valve has been processed it checks if the sprinkling has to be repeated. If so, the counter will be loaded with the set value and a delay begins. When this delay has passed, the cycle is started again with StartCycle. At the end of this process is closed, the control is returned and the display shows the initial start-up text.

This code can be compiled with Delphi and tested on the PC. The pushbutton on the evaluation board is Button 1, which is used to step through the menu. Button 2 is connected to pin 8 of the Delphi Stamp or J7 of the evaluation board, which is the second analogue input (the push button is connected via a 10 k pull-up resistor to +5 V). The output to the valves is simulated by the LEDs that are in a dice configuration on the evaluation board. The text on the LCD shows what is happening.

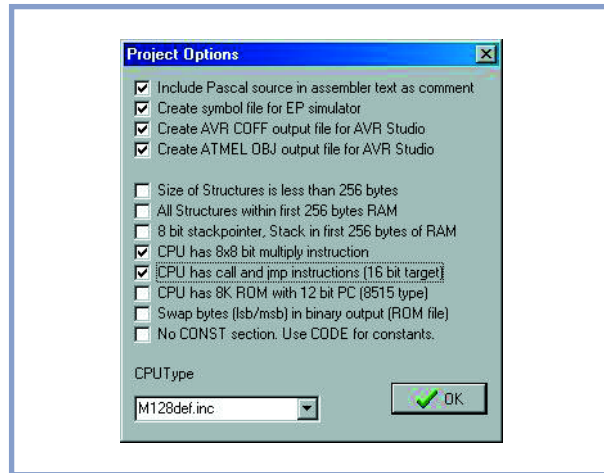


Figure 5. Make sure that the fourth and fifth options of the second group in AVRPas are ticked.



Figure 6. The monitor program M485 is used to connect to the Delphi Stamp.

We've now covered the development of the code via a simulation in Delphi. The code Ucontrol is used to drive the Delphi Stamp. The easiest way to get started is to create a separate (parallel) directory. Copy the general files for the Delphi Stamp (found in the AVR Start directory) into this and overwrite the unit Ucontrol in this directory with the unit Ucontrol that you've just created with Delphi. Next we continue with the AVRPas cross-compiler. This is also included with the Stamp in the kit. After starting this program we create a new project with **Project / New project**

Make sure that the fourth and fifth options are ticked, as shown in Figure 5, and click on **OK**.

A new window opens where a name and a location for the project have to be given.

Go to the directory where you've copied the files and give a name to the project, for example Rain. The generated files will now be stored in the same directory as the source code.

Add the main file to the project with **Project / Add to project**. Set the filter for *.pas files, select UMain.pas and click on **Open**. The file editor then opens and shows the contents of this file. Indicate that this is the main file with **Project / Project / Main file**, select UMain.pas

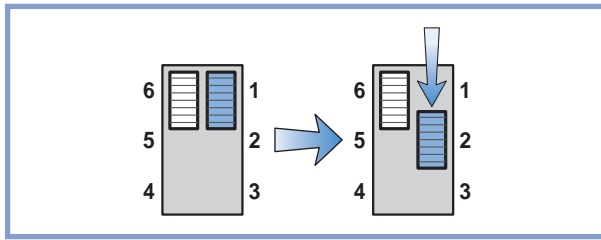


Figure 7. The right-hand slide switch on the Delphi Stamp is first pushed to the top and then one position lower.

and click on **Open**. Then open the other files in the editor with **File / Open** and add *UDrivers.pas* and *UControl.pas*. Compile these two files via **Compile / Current editor file**. Occasionally there can be a few error messages caused by some Delphi functions that are not found in AVRpas. These can usually be corrected with fairly simple modifications.

Next compile the project with **Project / Main Project file**. The code will then be generated and AVRpas can be closed.

The file with the binary code created by the cross-compiler has to be stored in the Delphi Stamp. For this we use the program Mon485 (**Figure 6**), which can make a connection to the Delphi Stamp via the RS232 port and the M485-server running on the Stamp.

Start the Mon485 program and reset the Delphi Stamp

by sliding the right-hand switch (next to the LEDs) to the top and then one position down (**Figure 7**). The LED then starts flashing at a rate of 0.5 Hz. The M485-server is now ready.

Connect the Stamp to the RS232 port and check the communication settings (probably Com1, use a speed of 9600 baud). Press the red button at the top right to make a connection. Press 'read' to show the serial number and test the communications channel. Then go to the Commands tab and press the button to the left of the input box to select the file to be downloaded. Go to the directory where AVRpas stored the created file and select the file with the '.ROM' extension. Click on the **Hex code** button to transfer the file to the Delphi Stamp. The **Run** button can then be used to start the program.

To make the program start independently in the Delphi Stamp the right-hand slide switch next to the LEDs has to be put to the bottom position.

If all goes well the unit will function identically to the simulation.

We now have functioning, standalone application running on the Delphi Stamp, which was the goal of this instalment.

(040240-9)

More information on the Delphi Stamp can be found on the websites www.vogelaar-electronics.com and www.learningdelphi.info

Solid-state Turn In

FET replaces relay

Willy Dhooge

Some motorcyclists not only enjoy tweaking the engine but also dabble in electronics. The author encountered problems with the mechanical turn indicator on his Harley Davidson bike and came up with an all-electronic alternative. As it turned out (pun intended), the little circuit is very useful for cars, too.

On motor vehicles like scooters, mopeds, motorbikes and cars, a mechanically controlled turn indicator (historically also known as a 'trafficator') arranges for lamps to flash at the side, front and back of the vehicle when the driver intends to make a major turn. For decades, the circuit controlling the turn indicator lights has been based on a relay or a bimetallic strip, which are both subject to wear and tear and hence breakdown. If failure occurs, an original control unit is often difficult to get and one soon start to think about an electronic version with eternal life.

This month's *Mini Project* demonstrates that an all-electronic replacement is easy to build from just standard components. The key element is a power MOSFET capable of passing currents up to 26 A (**Figure 1**).

How does it work?

The circuit is effectively connected in series with the turn indicator lamps.



Indicator

When the FET (field effect transistor) conducts, components D1 and C1 ensure the supply voltage remains present for the drive electronics, during the on-periods. Moreover, as a result of the on/off switching, the cathode of D1 is 'lifted' some 11 V above the battery voltage, allowing the gate of the MOSFET to switch the device into full conduction.

When the supply voltage is switched on, C2 is initially discharged, causing T3 to be kept off. The base of T2 is held low by R4, causing T2 to conduct and the emitter of T3 to be pulled down to about 2 V. The gate of T1 being connected to the emitter of T2 via R1, T2 will not conduct. Once C2 is almost fully charged (via R3, D3 and T2), T3 starts to conduct, switching off T2. Next, FET T1 is driven into conduction via R6 and R1 and the lamps will light. While T3 conducts, C2 is discharged via R5, T3 and R2 causing T3 to block after a while and T2 to be driven into conduction again. T1 then no longer receives a gate voltage and switches off.

A mini board

Although the circuit diagram seems to contain a good number of components, actually building the circuit on a PCB is neither a difficult nor a long-winded affair. Looking at the PCB artwork in **Figure 2**, the whole circuit is accommodated on a 3 × 3 cm square PCB.

Mounting the parts is not expected to cause difficulty — the FET may be mounted upright and does not require a heatsink.

The wiring diagram in **Figure 3** is intended as a guide to installing the circuit in or on the vehicle. In fact, there is little change with respect to the original configuration, but do note that connection T1 is wired to the +12 V line, and T2, to the turn indicator switches.

(050120)

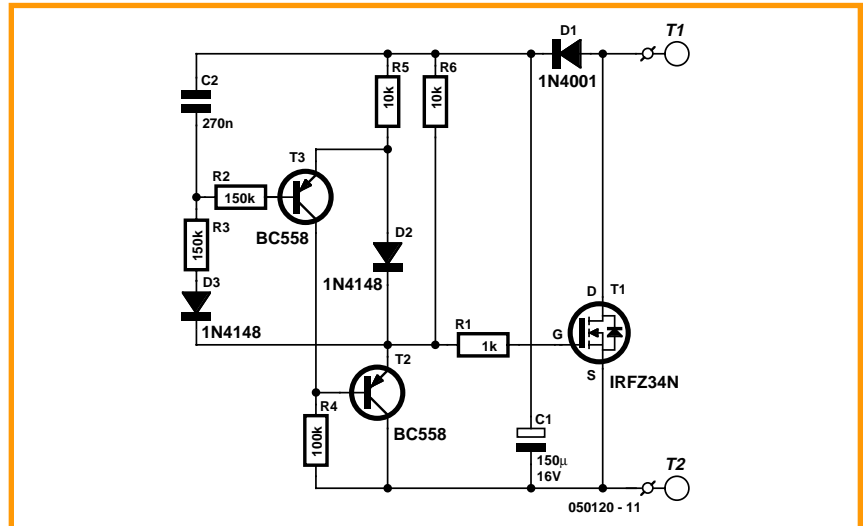


Figure 1. The circuit is designed to employ a kind of bootstrapping effect when the FET gate is driven.

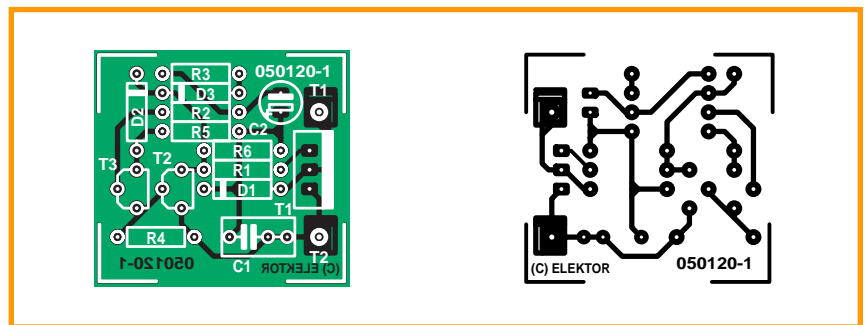


Figure 2. The circuit board is so small it can usually be fitted into the original turn indicator control box.

COMPONENTS LIST

C2 = 270nF

Resistors:

R1 = 1kΩ
R2, R3 = 150kΩ
R4 = 100kΩ
R5, R6 = 10kΩ

Semiconductors:

D1 = 1N4001
D1, D2 = 1N4148
T1 = IRFZ34N
T2, T3 = BC558

Capacitors:

C1 = 100µF or 150µF, 16V, radial

Miscellaneous:

PCB, ref. 050120-1 from The PCBShop

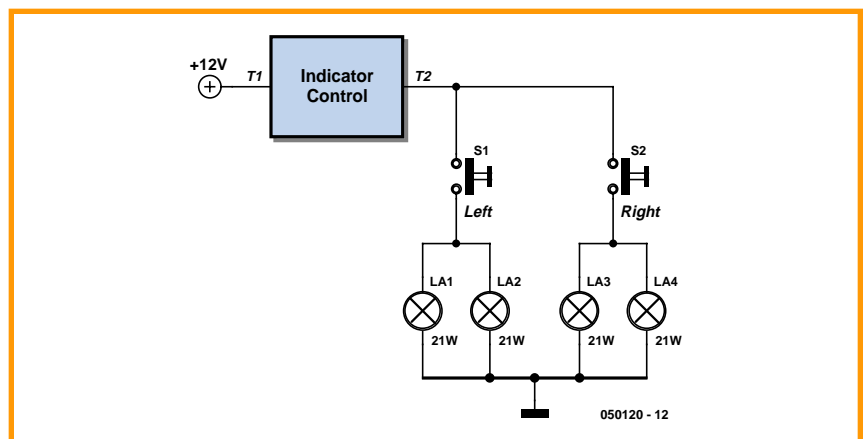


Figure 3. Connection diagram of the turn indicator incorporating the all-solid state drive circuit.

In all mains-operated equipment certain important safety requirements must be met. The relevant standard for most sound equipment is Safety of Information Technology Equipment, including Electrical Business Equipment (European Harmonized British Standard BS EN 60950:1992). Electrical safety under this standard relates to protection from

- a hazardous voltage, that is, a voltage greater than 42.4 V peak or 60 V d.c.;
- a hazardous energy level, which is defined as a stored energy level of 20 Joules or more or an available continuous power level of 240 VA or more at a potential of 2 V or more;
- a single insulation fault which would cause a conductive part to become hazardous;
- the source of a hazardous voltage or energy level from primary power;
- secondary power (derived from internal circuitry which is supplied and isolated from any power source, including d.c.)

Protection against electric shock is achieved by two classes of equipment.

Class I equipment uses basic insulation; its conductive parts, which may become hazardous if this insulation fails, must be connected to the supply protective earth.

Class II equipment uses double or reinforced insulation for use where there is no provision for supply protective earth (rare in electronics – mainly applicable to power tools).

The use of a Class II insulated transformer is preferred, but note that when this is fitted in a Class I equipment, this does not, by itself, confer Class II status on the equipment.

Electrically conductive enclosures that are used to isolate and protect a hazardous supply voltage or energy level from user access must be protectively earthed regardless of whether the mains transformer is Class I or Class II.

Always keep the distance between mains-carrying parts and other parts as large as possible, but never less than required.

If at all possible, use an approved mains entry with integrated fuse holder and on/off switch. If this is not available, use a strain relief (Figure, note 2) on the mains cable at the point of entry. In this case, the mains fuse should be placed after the double-pole on/off switch unless it is a Touchproof® type or similar. Close to each and every fuse must be affixed a label stating the fuse rating and type.

The separate on/off switch (Figure, note 4), which is really a 'disconnect device', should be an approved double-pole type (to switch the phase and neutral conductors of a single-phase mains supply). In case of a three-phase supply, all phases and neutral (where used) must be switched simultaneously. A pluggable mains cable may be considered as a disconnect device. In an

approved switch, the contact gap in the off position is not smaller than 3 mm.

The on/off switch must be fitted by as short a cable as possible to the mains entry point. All components in the primary transformer circuit, including a separate mains fuse and separate mains filtering components, must be placed in the switched section of the primary circuit. Placing them before the on/off switch will leave them at a hazardous voltage level when the equipment is switched off.

If the equipment uses an open-construction power supply which is not separately protected by an earthed metal screen or insulated enclosure or otherwise guarded, all the conductive parts of the enclosure must be protectively earthed using green/yellow wire (green with a narrow yellow stripe). The earth wire must not be daisy-chained from one part of the enclosure to another. Each conductive part must be protectively earthed by direct and separate wiring to the primary earth point which should be as close as possible to the mains connector or mains cable entry. This ensures that removal of the protective earth from a conductive part does not also remove the protective earth from other conductive parts.

Pay particular attention to the metal spindles of switches and potentiometers: if touchable, these must be protectively earthed. Note, however, that such components fitted with metal spindles and/or levers constructed to the relevant British Standard fully meet all insulation requirements.

The temperature of touchable parts must not be so high as to cause injury or to create a fire risk.

Most risks can be eliminated by the use of correct fuses, a sufficiently firm construction, correct choice and use of insulating materials and adequate cooling through heat sinks and by extractor fans.

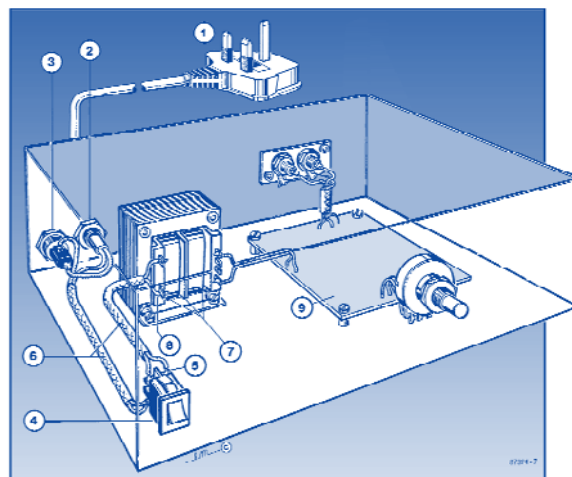
The equipment must be sturdy: repeatedly dropping it on to a hard surface from a height of 50 mm must not cause damage. Greater impacts must not loosen the mains transformer, electrolytic capacitors and other important components.

Do not use dubious or flammable materials that emit poisonous gases.

Shorten screws that come too close to other components.

Keep mains-carrying parts and wires well away from ventilation holes, so that an intruding screwdriver or inward falling metal object cannot touch such parts.

As soon as you open an equipment, there are many potential dangers. Most of these can be eliminated by disconnecting the equipment from the mains before the unit is opened. But, since testing requires that it is plugged in again, it is good practice (and safe) to fit a residual current device (RCD)*, rated at not more than 30 mA to the



1. Use a mains cable with moulded-on plug.
2. Use a strain relief on the mains cable.
3. Affix a label at the outside of the enclosure near the mains entry stating the equipment type, the mains voltage or voltage range, the frequency or frequency range, and the current drain or current drain range.
4. Use an approved double-pole on/off switch, which is effectively the 'disconnect device'.
5. Push wires through eyelets before soldering them in place.
6. Use insulating sleeves for extra protection.
7. The distance between transformer terminals and core and other parts must be ≥ 6 mm.
8. Use the correct type, size and current-carrying capacity of cables and wires – see shaded table below.
9. A printed-circuit board like all other parts should be well secured. All joints and connections should be well made and soldered neatly so that they are mechanically and electrically sound. Never solder mains-carrying wires directly to the board: use solder tags. The use of crimp-on tags is also good practice.
10. Even when a Class II transformer is used, it remains the on/off switch whose function it is to isolate a hazardous voltage (i.e., mains input) from the primary circuit in the equipment. The primary-to-secondary isolation of the transformer does not and can not perform this function.

mains system (sometimes it is possible to fit this inside the mains outlet box or multiple socket).

* Sometimes called residual current breaker – RCB – or residual current current breaker –RCCB.

do not assume, and hereby disclaim, any liability for any loss or damage, direct or consequential, caused by errors or omissions in these guidelines, whether such errors or omissions result from negligence, accident or any other cause.

These guidelines have been drawn up with great care by the editorial staff of this magazine. However, the publishers

3-core mains cable to BS6500 1990 with three stranded conductors in thick PVC sheath

Max current	3 A	6 A	13 A
conductor size	16/0.2 mm	24/0.2 mm	40/0.2 mm
Nom cond area	0.5 mm ²	0.75 mm ²	1.25 mm ²
overall cable dia.	5.6 mm	6.9 mm	7.5 mm

Insulated hook-up wire to DEF61-12

Max current	1.4 A	3 A	6 A
Max working voltage	1000 V rms	1000 V rms	1000 V rms
PVC sheath thickness	0.3 mm	0.3 mm	0.45 mm
conductor size	7/0.2 mm	16/0.2 mm	24/0.2 mm
Nom cond area	0.22 mm ²	0.5 mm ²	0.95 mm ²
overall wire dia	1.2 mm	1.6 mm	2.05 mm

3-flat-pin mains plug to BS 1363A

Parallel-port Flasher for Elektor Flash Micro board

Tilo Gockel

Most students participating in the microcontroller programming courses taught at our technical university (Karlsruhe, Germany, Ed.) bring their own notebook PCs and run into problems when they discover there's no RS232 port available to use with, for example, Flashbot Roadster. Attempts to flash the *Elektor 89S8252 Flash Micro board* using the AtmelISP utility and several USB-to-RS232 converters only proved that data was terribly slow to get transferred. Ulrich Bangert, the author of the AtmelISP utility explained to us that "apparently, a latency timer has to elapse each time a line needed for ISP is switched to a different state, before the driver change is actually copied to the USB component". A pity, the AtmelISP utility being an extremely useful tool, but by no means hopeless, as most notebook PCs do have a parallel port. Using the program SP89 [1] for Windows (as well as for Linux!) the Elektor Flash Micro board can be programmed via the *parallel* port. The actual use of the program is not as easy as with AtmelISP, but using the tips in this article success is within everyone's reach.

Here's how it works

To be able to use the SP89 Flasher (download it from [1]) a

second program called *userport.exe* is required for Windows 2000 and XP. Fortunately that's also become available from a download link on the SP89 page already mentioned. After unpacking the archive file, the *userport.sys* file has to be copied into the *C:\Windows\system32\drivers* directory, whereupon *userport.exe* can be launched. Then you are able to select the port and address range to be accessed (the relevant information is available via *Control Panels / System / Hardware / Device Manager / Ports (COM and LPT) / Properties / Resources*, and is usually 0378-037F). The entry in the right-hand column of *userport.exe* is the essential one; compare it with the documentation supplied by the

program. Once this has been configured (the program will supply an ok message) the board may be connected up to the PC's LPT port using an easily made cable. The connector pinout is given in the **Table**. All pins of the Flash Micro board are accessible for our purpose, except the reset pin. We recommend fitting a solder pin at the solder side of the board, on the connection for the reset switch. This allows mating sockets of the sleeve type to be used just as with the other connections. Basically the wiring is the same as indicated on [1], although the only components remaining are the three 220-Ω series resistors — the rest is superfluous luxury. It is, however, essential for the cable

to be kept shorter than about 20 cm. A longer cable requires an active alternative, generating the required signal levels with the aid of 74xxx244 bus drivers, see [3]. Although no serious problems were encountered when using the passive version, the active variant is recommended for different reasons. As compared with the serial port, the printer port is quite susceptible to damage by voltage surges and short-circuits. These risks should not be neglected and in case of doubt the active alternative [3] is to be preferred.

As soon as the connection has been established, the program may be launched as follows (note the spaces)

```
SP89 -i52 -M11 -fwCp test.ihx
```

Apparently, it is necessary after each unsuccessful launch to clear the resource file *_sp89rc* to make sure a new one is created the next time SP89 is used. The relevant file contains plain ASCII — hence can be opened with a word processor — and will quickly tell you if the program was launched successfully. Regarding the LPT mode in the PC's BIOS we should not forget to mention that our tests were successful using the 'ECP+EPP' setting. Further details on the program start options may be found in the documentation file that comes with SP89 [1].

(050229-1)

SP89 - a programmer for AT89 microcontrollers

The present version (0.7) is suitable for the following uCs And offers facilities as shown in the table:

	chip	read/write	read/write	write
	erase	Flash	EEPROM	Lock
AT89S8252	YES	YES	YES	YES
AT89L8252	YES	YES	YES	YES
AT89S53	YES	YES	YES	YES
AT89L53	YES	YES	YES	YES

Have a look at the [documentation](#).

The software is intended for 'in circuit' programming, which means that you don't need special hardware, apart from a cable between a parallel port of your PC and the target board. That board must allow free access to MOSI, MISO, SCK and Reset. It is a good idea to terminate SCK on the target board using a 22pF capacitor and a 100 ohms resistor. A look at the schematic:

The programmer software is available under the [GNU General Public Licence](#); the packages below (for Linux, XP and Dos/VoX) contain the source code, documentation and an executable.

SP89 for Dos and Windows 9X/9X DosBoxes

COMPATIBILITY
The 32 bit protected mode executable found in this version of SP89 relies on MS-DOS and BIOS for file I/O and other basic functions such as keyboard input, screen cursor position, etc. It uses "DPMI" (the DOS Protected Mode Interface) to allow DOS/BIOS calls from protected mode. SP89 is known to run in MS-DOS, in a Windows-3.X DosBox and in a Windows-9X DosBox or DosMode. Windows NT does not provide the access to the parallel port required by SP89.

Connector cable wiring		
3 (DATA1)		RESET pin (see text)
9 (DATA7)	via 220 Ω	P1.5 (MOSI)
11 (STATUS 7, busy)	via 220 Ω	P1.6 (MISO)
2 (DATA 0)	via 220 Ω	P1.7 (SCK)
25 (GND) K7		(GND)

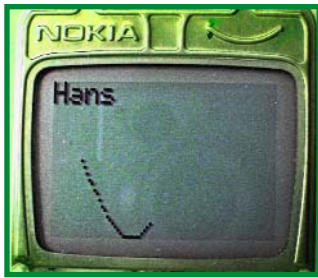
References

[1] SP89 Parallel Port Flasher homepage:
<http://www.xs4all.nl/~sbolt/e-ser89.html>

[2] Active (buffered) Parallel port flasher for the AT89S8252:
http://web51.hw-server.com/isp_prg.html

[3] Programming the AT89S8252 using SPI:
<http://jelmer.vernstok.nl/oss/at89prog/at89prog.pdf>

PICAXE controls Nokia 3310 display



Hans Middelbeek

Spare displays for Nokia 3310 mobile telephones are offered cheaply not only by several surplus goods outlets but also via Ebay. A small program allows such a display to be controlled from a PICAXE micro. The PICAXE is a PIC-based microcontroller marketed by Revolution Education Ltd. The special loader utility programmed into the chip allows it to be programmed with uncommon ease.

The little system shown here is perfect for applications like a small datalogger with graphic output. **Figure 1** lists the connection data for the display while the actual circuit diagram around a PICAXE 18X is given in **Figure 2**. Two signal lines, SCLK (clock) and SDA (data) are needed to transfer data to the display.

Besides these two, we also need DC (data/command), CS (chip select) and RES (reset).

We need to start by telling the display if the information to be sent is data or a command — a high level of DC indicates data, a low level, a command. Next, we pull CS low to select the display.

We can then change the value of bit 7 (most significant bit) on the SDA line and change SCK from low to high. The display awaits this action before it reads in the level of SDA.

Next, the whole byte is shifted to the left by one bit position (i.e., multiplication by two) and the above actions are repeated until all eight bits of the command or data byte have been conveyed.

As shown in the table, only six commands are in principle necessary to initialise the display. Once these have been processed, the display is ready to receive further data. The transmission of data in order to display pixels follows the same procedure, eight pixels being packed into one byte.

The program written by the author for the PICAXE arranges the initialisation of the display, writes a word on the top line and then displays a part of a sine wave. The program may be

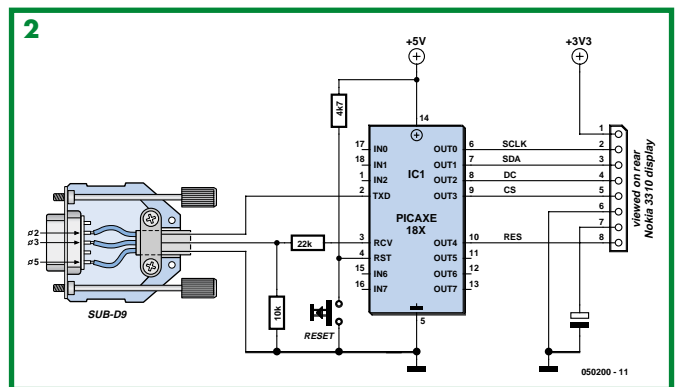
1 Connection data

Pin 1	LCDRSTX - REst (2,8 V)
Pin 2	VOOUT - DC/DC - DC/DC voltage converter output (0V)
Pin 3	Gnd (0V)
Pin 4	LCDCSX - Chip select input (2,8V)
Pin 5	LCDCDX - Control/display data flag input (2,8V)
Pin 6	SDA - Serial data input (0V)
Pin 7	SCLK - Serial clock input (0V)
Pin 8	VBB - Supply voltage (2,8V)

downloaded free of charge from our website, the file number is **050200-11.zip**. The included

subroutines speak for themselves by and large.

(050200-1)



Formerly a tobacco tin

Günther Kuhnle

Many electronics enthusiasts are at war with mechanical engineering, and in keeping with the slogan 'if you can't beat them, join them' we present a quick and easy method of housing small circuits in an all-metal enclosure. Although the pictures speak for themselves, some comment may be useful.

The dimensions of the author's favourite (empty!) tobacco tins are 110 _ 80 _ 26 mm (l _ w _ h). The box itself conveniently holds the components or PCB while the lid not only serves as a mounting panel for controls and connectors, but also as an effective screen. If necessary the project schematic may be secured to the inside of the lid, and protected with a coating of clear lacquer.

Clearances in the lid should be marked with a felt pen only, to prevent scratching the paint on the surface. A piece of wood, slightly higher than the inside of the box, is used to prevent the drill bit from wandering around. To make sure the holes become round, wood drills should be used instead of spiral ones for metal. Wood drills will not take damage from the thin sheet metal, provided you pre-drill each hole with a 1.5-mm bit.

To ensure a constant supply of empty tobacco tins, the author encourages a friend to keep smoking his pipe, having convinced him that by doing so, many will profit from his personal choice of tobacco. So far, he has remained faithful to his favourite brand.

(050206-1)



Tolerances

Karel Walraven

How accurate do you need to be?

Prices in the supermarket are often rounded to 5 pence, but the price of cars is usually rounded to hundreds of pounds. There is apparently a relationship between the amount to pay and the accuracy.

This idea can be misleading, because we're used to thinking in absolute amounts. When a salesman says that a product is only 5% more expensive and that it doesn't amount to much, I always immediately suggest that he gives me 5% of his salary. It is not much, is it? Unfortunately I have never succeeded in getting even a single cent this way. When you use an ordinary transistor such as the BC547B, the gain can easily have a tolerance of 100%. Many electrolytic capacitors have tolerances from -20 to +50%. With resistors, a tolerance of 5% or 1% is quite normal, but most capacitors have a tolerance of at least 10%. If you look at it that way, it is a small wonder that circuits actually work at all! Or does it not work that way?

The designer of the electronic circuit has to keep tolerances very much in mind. He has to design it in such a way that the proper operation of the circuit is guaranteed under all circumstances. That is normally done by designing in a suitable margin, for example, by assuming the lowest gain for all the transistors. This results in a design that has too much gain, which is subsequently corrected with negative feedback. This is a kind of trade-off, where gain is sacrificed in order to realise a reproducible design. Incidentally, negative feedback is also necessary for a whole host of other reasons that

are not relevant right now. You could say that the negative feedback circuit determine the most important characteristics (gain, in this example) of the design. The components that are part of the negative feedback circuit have a much higher demand placed on them and may not have large tolerances. Fortunately, negative feedback in its simplest form consists of not more than two resistors and resistors are stable and easily manufactured in just about any tolerance.

We usually use 5%-resistors (that is, resistors with a 5% tolerance) in our designs, and occasionally 1% or even less. We do that, because you will not hear whether the volume of an amplifier is 5% higher or lower. If you were to measure it (with an instrument that has an accuracy that is better than 5%) then you could easily measure a difference between amplifiers (left and right channel). This is also the reason why in your digital voltmeter with an accuracy of 1% or better, 1% or even 0.1% resistors are used in the critical locations.

About a century ago, some designers had not quite grasped the idea of tolerances. Resistors with values of 1000 Ω , 2000 Ω , 3000 Ω , 4000 Ω , and so on in

equal steps, were manufactured. Obviously, this range cannot be maintained, because with low values, the steps are much too large, and with high values much too small. The solution to this dilemma is a better understanding of the concept of tolerances as follows: A 1000- Ω resistor with a $\pm 10\%$ tolerance can have a value from 900 to 1100 Ω . The region from 900 to 1100 Ω is therefore covered. The next value has to be selected in such a way that it neatly continues the range. The value of 1200 Ω is suitable, because the actual value will be between 1080 and 1320 Ω . Then follows 1500 Ω with a value between 1350 and 1650 Ω . This is how the E12 range has come about. For the picky among us: to calculate the unrounded E12-values you need to take the twelfth root of 10, which results in a factor of 1.212. So the E12-series should actually have the following values: 1.0 – 1.212 – 1.468 – 1.778 – 2.154 – 2.610 – 3.162 – 3.831 – 4.642 – 5.623 – 6.813 – 8.254. With these values, the tolerance regions do not overlap.

In actual daily use we're not that consistent, because with a 5% tolerance we should really be using the E24-series. We normally don't do that, we use E12 values and allow gaps between the values.

(050288-1)

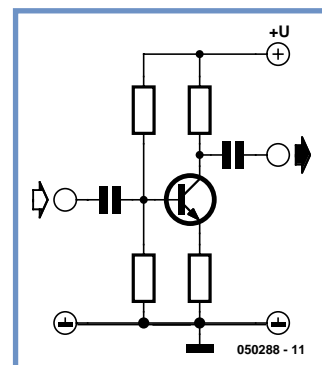


Figure 1. Negative feedback in its simplest form using R_i . We sacrifice gain and in return we obtain a fixed amplification, a stable DC adjustment and high input resistance.

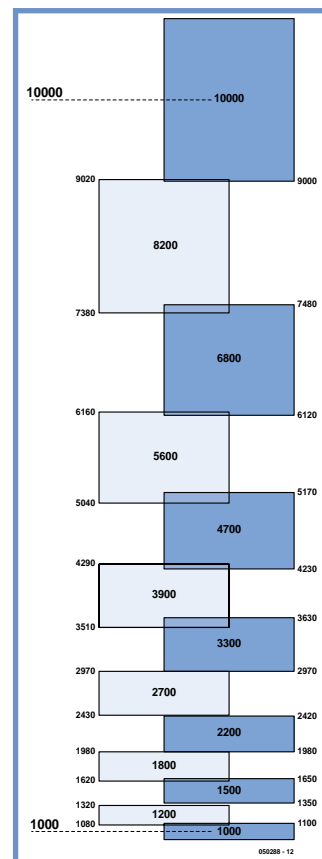


Figure 2. Tolerance regions for the E12 series. Because the values are rounded, the regions do not join exactly but overlap a little or leave a small gap.

Close encounters with a fossil

Friedrich W. Ruppert

When I saw this radio in its beautifully polished wooden case on the pavement beside a garbage bin ready for collection I could not resist saving it. Next, it spent about 20 years in my loft because there was no time 'for such a thing'. Today, I tend to think differently and our children should not miss the opportunity to view the 'magic eye' (for relative newcomers, that's the valve tuning indicator type zM xy, a poor man's scalar cathode ray tube). The results of my first attempt to power up the radio were as expected: soft light from the dial illumination and the valves, the magic eye came on, but no sound. A few turns of the tuning knob did however cause the eye to close and open, so something inside was actually receiving FM broadcasts! Conclusion: a dead power output stage.

The radio was opened and hey presto there was a copy of the schematic secured to the screened bottom plate. Date: June 1959; type: Philips 1002. The documentation is of a standard and quality that's long since disappeared. Second surprise: a 'totem-pole'-ish output stage virtually to the SRPP principle *Elektronics* extensively discussed in a couple of famous articles. Here, pentodes were used rather than triodes (and note the unusual screen grid configuration) The cathode of the upper EL86 is at a perilously high voltage w.r.t. its filament grounded at one side, which apparently was no cause for concern in the 1960s and 70s. Valve handbooks will tell you that the EL86 is the low-voltage version of the EL84. The '86 was developed by Valvo for transformerless power output stages designed for use with 800-ohm loudspeakers only. Faultfinding in the radio proved easy in thanks to the perfect servicing sheet. The actual cause of the utter silence was a broken fil-

ament in one of the ECC83 valves. This caused the plate voltage to rise to a level so high it was endangering the lives of the valves and the electrolytic smoothing capacitors (three, packed into a common case, ready to explode). An easy repair, as it turned out.

However, in order to compensate for the still too high voltages as a result of the mains voltage having risen from 220 V to 230 V in my country, I decided to insert a large 50-ohm dropper resistor (on a heatsink) into the transformer primary circuit.

Once resurrected and restored to its former glory the radio is now a joy to listen to. Nothing on modern equipment compares to the 'feel' of the tuning knob, which, with 400 grams of inertia, allows the entire VHF FM band to be tuned through within half a second — and no waiting for PLLs or similar newfangled circuitry to lock.

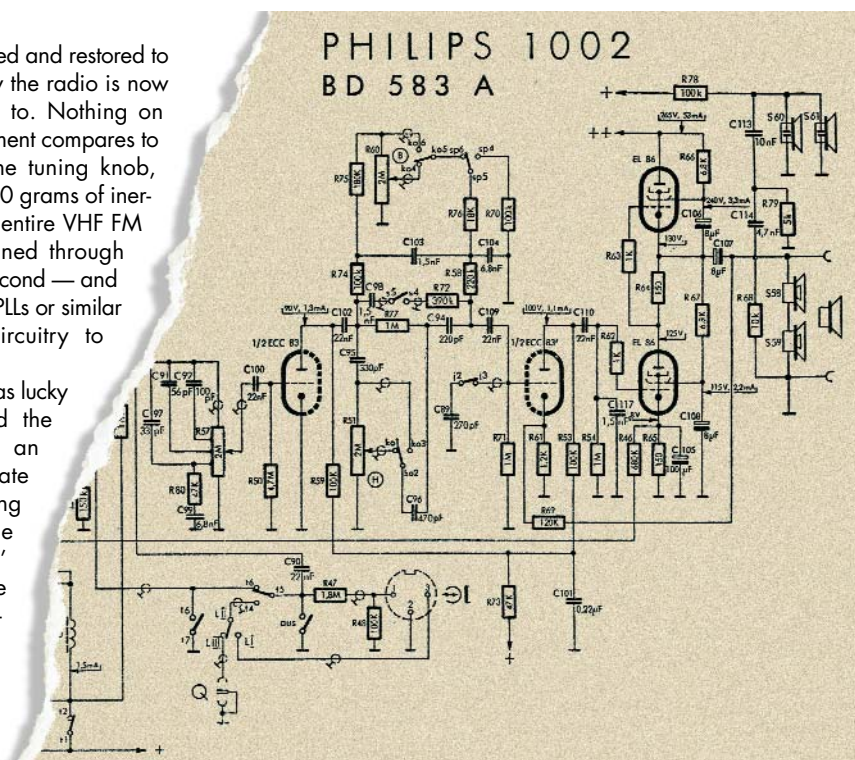
I do realise I was lucky to have found the radio in such an exemplary state — no crackling noises when the heavy 'denture' knobs are pushed, no sizzling or other worrying noises from leaking electrolytics, and

remarkably little noise.

The almost grotesque treble/bass control comfort, including the various music/speech settings and last but not least the 12 watts of audio power from inside a 45-litre wooden case ensure a retro-*deja-vu* (deja-heard?) sensation second to none. The radio is great not only for a classic concert on a rainy Sunday afternoon; it will also fill a busy mid-size office or workshop with 'music at work' (alas no longer allowed these days). Another of its strengths as compared to modern transistor radios is discovered

after sunset, when you can go 'wave-hunting' with just a short length of wire for an antenna. Surprisingly, a lot of stations with their wonderful foreign names printed on the dial are still on the air, broadcasting programmes that somehow seem free from modern society's stress, hustle and bustle. Or is it just the warm sound and the faint smell of 'electrified' dust from the back of the radio? A feather in the cap of those who designed and built this superb product almost 50 years ago.

(055076-1)



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.

NOP changes spectrum

This month we look at an 8051-ish microcontroller (an AT89S8252) generating a simple rectangular signal under interrupt control using the program listed here. The micro is clocked at 12 MHz, which corresponds to a machine cycle of 1 μ s. One interrupt signal is generated by Timer 0 every eight cycles. The interrupt routine (INTT0) toggles port pin bit P2.0 which consequently supplies a rectangular signal at a rate of 62.5 kHz. The main program (START routine) does all the necessary in respect of initialisation and then enters a loop (LOOP) consisting of a few NOP (no-operation) instructions and a jump.

So far, everything's fine: a proper

```

; VERSION1.A51 : AT89S8252 , 12 MHz Xtal
; clean 62.5 kHz Signal on P2.0
ORG 0
LJMP START

INTT0  ORG 000BH ; Timer0 Interrupt
      CPL P2.0 ; Bit P2.0 toggle
      RETI ; done

START  MOV TMOD,#022H ; Mode = Timer
      MOV TH0,#256-8 ; every 8 Ticks one Interrupt
      SETB TCON.4 ; start TIMERO
      SETB IE.1 ; TIMERO Interrupt on
      SETB IE.7 ; Interrupts on

LOOP   NOP
      NOP
      NOP
      NOP
      ; insert additional NOPs here !!
      SJMP LOOP
      END
    
```

Source code listing.

signal is generated of which the clean spectrum (fundamental frequency) is shown in **Figure 1**. A frequency counter will confirm the correct frequency of 62.5 kHz.

Solution to the September

(p. 78; mysterious action from a TTL gate)

To be able to understand why the LED lights when switch S is pressed, you need to be aware of the electronic composition of a TTL NAND gate. **Figure 3** shows the internal structure of a 74LS00 gate (source: Texas Instruments).

The actual logic combination takes place on the two Schottky diodes D1 and D2. Resistor R4 supplies the base current for the first transistor, in case neither logic input is at Low (logic 0). If switch S is opened, current I0 is zero while a current I2 flows from R4 to R2. Both inputs of gate N2 also supply

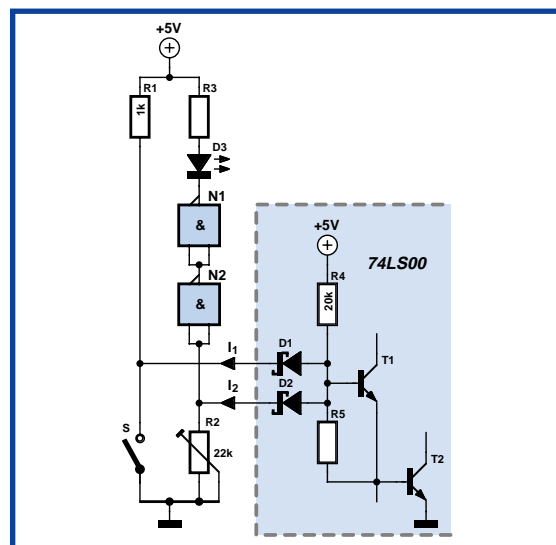


Figure 3. Internal structure of a 74LS00 gate.

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 11-05'.

The closing date is
30 November 2005

(solution published in the December 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.



As of the September 2004 issue Quizz'away is a regular feature in Elektor Electronics. The problems to solve are supplied by Professor Martin Ohsmann of Aachen Technical University.

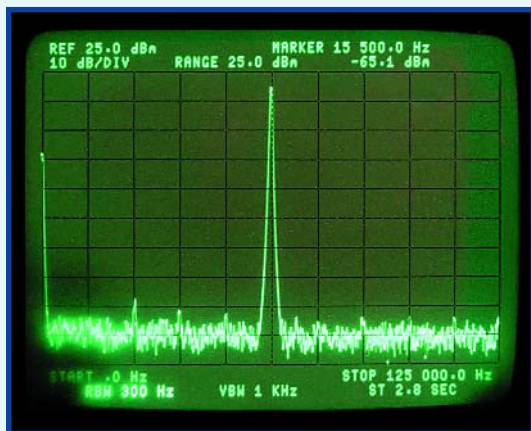


Figure 1. Spectrum of rectangular wave.

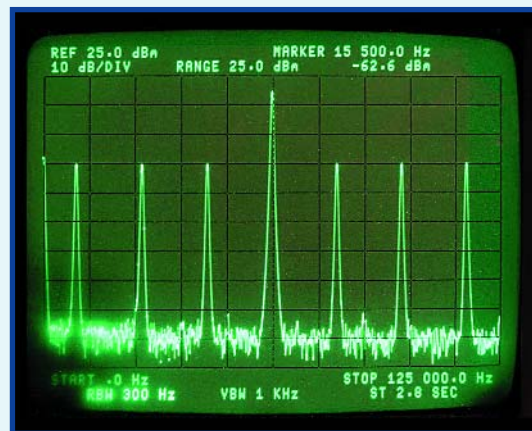


Figure 2. Spectrum with one additional NOP.

We now do a simple modification to the program. Instead of four, we use five NOPs in the loop — we simply insert a NOP at the indicated location. The result of this change is a dramatic change in the output spectrum, as proved by **Figure 2**. Remember, no changes were made to the main program! The frequency counter will again faithfully read 62.5 kHz. So how can a change in the main program result in

a different behaviour of the interrupt routine response?

This month's question is:
Why does a NOP in the main program modify the spectrum?

Hint:

An explanation of the exact composition of the spectrum is not necessary. The experiment was done using an AT89S8252, but we can safely assume that any 8051 compatible microcontroller will behave likewise. You should, however, stick to the exact program listing shown here.

2005 problem

a current. The value of resistor R2 is such that the voltage drop across it is sufficient to be taken for a Logic High level by gate N2. As a result, the LED remains off.

When S is actuated, current I2 drops to zero because current I1 starts to flow. The voltage across R2 drops, N2 recognises a logic Low and the LED lights.

Basically the same applies with chips from the standard 7400 series, albeit that a double-emitter transistor is the functional equivalent of diodes D1 and D2. The example shows how inputs can cause unforeseen effects if they are not properly connected up or driven.

Quizz'away and win!

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

**Elektor
Electrosmog Tester**

ready assembled and tested,
including case

with a value of
about **£ 60**.

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.

