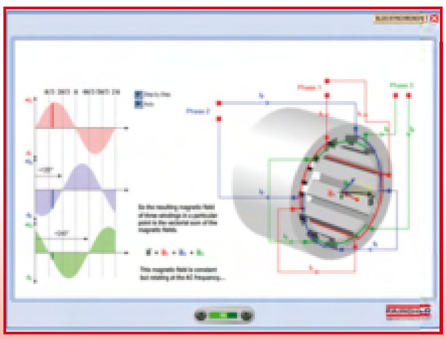
www.elektor-usa.com FEBRUARY 2009 US $ 7.95 - Canada $ 7.95

elektor

electronics worldwide

Transistor curve tracer on USB

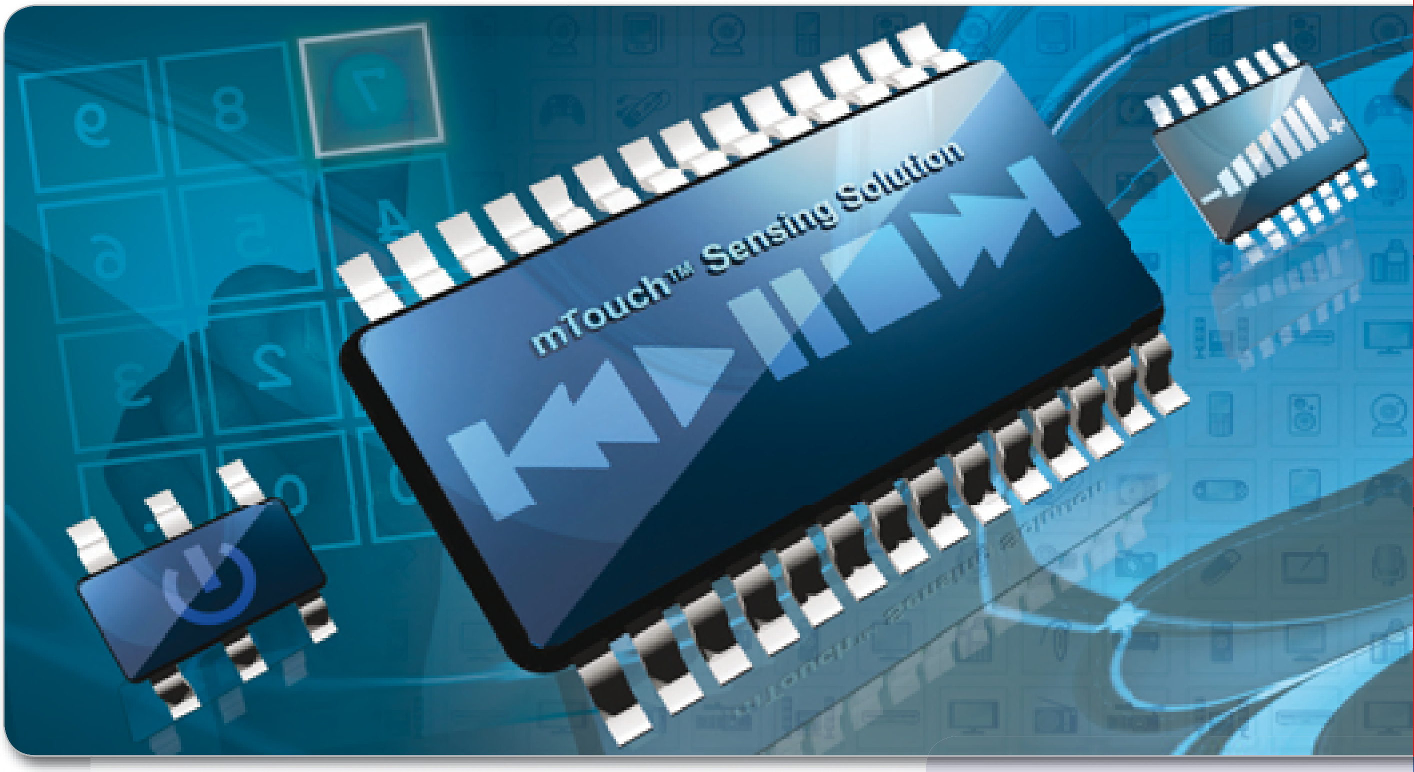
Free CAD design tools online



LED Lighting for model railway cars

FPGA programming in C

Capacitive Touch Sensing In a Flexible, Single-chip Solution



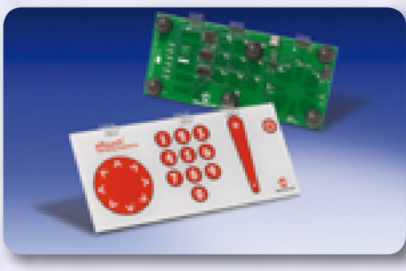
Capacitive touch interfaces provide an excellent way to add low-cost, reliable and stylish buttons into your design. Microchip Technology's mTouch™ Sensing Solution includes comprehensive development kits and a free diagnostic tool to make implementation easy and fast. Our free source code can be seamlessly integrated with your existing firmware on a single PIC® microcontroller – eliminating the need for additional controllers.

THE mTouch SENSING SOLUTION FEATURES:

- FREE license libraries and source code
- A FREE diagnostic tool
- Integration with 8- and 16-bit PIC microcontrollers
- Easy expansion, with support from 6 to 100 pins
- Low-power operation

GET STARTED IN 3 EASY STEPS

1. Visit the mTouch Sensing Solutions design center at www.microchip.com/mTouch
2. Download free libraries and source code
3. For a limited time, save 20% off a variety of touch sensing development tools when you purchase from www.microchipDIRECT.com and use voucher code **EUMTOUCH**.



Intelligent Electronics start with Microchip

microchip DIRECT www.microchipdirect.com

www.microchip.com/mtouch

MICROCHIP

elektor
electronics worldwide

Cooking and grilling with Elektor

Our technical staff in the Elektor lab always do their best to make a serious impression, working all the time with complicated electronic designs and test equipment. However, sometimes their sense of fun gets the upper hand, and then there's no telling what might happen. This time the action revolved around a can of frankfurters left standing in a corner for half a year already, waiting for some sort of special preparation. And when we say special, we mean special.

Needless to say, cooking and grilling with Elektor always goes hand in hand with a certain electric tension (literally) and suitable sensation. The plan was to apply an ac voltage directly to the frankfurter so it could be cooked almost instantaneously. Our lab staff are keen on special effects, so they decided it would be nice to stick a few LEDs in the frankfurter, with the idea that they would light up when the frankfurter was cooked through. This led to a fairly animated discussion of whether they would actually light up. Apparently it's not that easy to apply Ohm's law and Kirchoff's rules to a frankfurter.

Finally, the time came for action. The can of frankfurters was opened, a heavy-duty adjustable, double isolated transformer was brought out of storage, forks were borrowed from the canteen so the current could penetrate deep into the frankfurter, and a good crowd of people gathered around to watch the events first-hand. Interestingly enough, our lab manager, Antoine Authier, took up a position behind a low cabinet that provided strategic cover. The French naturally know how to deal with food, and he probably thought the frankfurter might suddenly catch fire.

…Tension rising … at 40 V, the frankfurter still resolutely maintains its resistance… the first wonder occurs at 70 V: the LEDs light up! …the voltage rises even higher… at 80 V, the first plume of smoke comes from the frankfurter, and it's done!

The LEDs made a positive contribution to the cooking session. After some experimenting, we found that the best results can be expected at 80 V. The only other thing we should mention is that Jan Visser ate the frankfurters and is still in good health (as far as we know). And of course: **never try this at home!**

Wisse Hettinga
International Editor

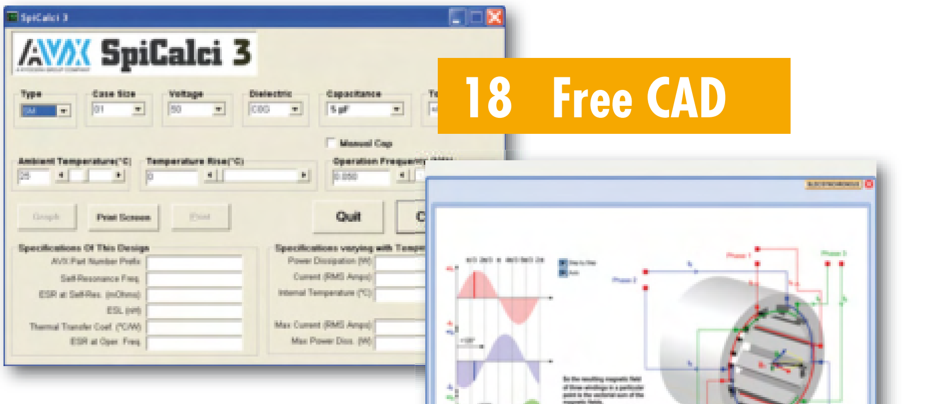
Transistor Curve 24

This practical instrument can be used to measure and record the characteristic curves of NPN and PNP bipolar transistors, N- and P-channel JFETs, and N- and P-channel MOSFETs. The circuit is based on an R8C/13 microcontroller, which transfers the measurement data to a Windows application program via USB.

BASCOM-AVR Reader Offer

details on page 55

18 Free CAD



Several electronic component manufacturers now offer design tools online or for free download. We registered on a dozen or so websites to see just what's being offered and what it's like.

CONTENTS

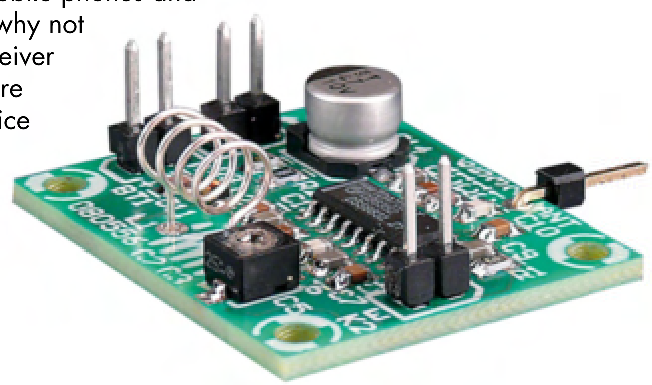
Tracer

56 Model Railway Car Lighting

The lighting controller described in this article is a combination of an SMD LED strip and a PIC12F683 for compatibility with the Märklin system. The LEDs allow easy adaptation to different colours while three different lengths of the board enable the project to fit all common railway car types.

69 Mini VHF FM Receiver

These days tiny FM radios are often integrated in many portable devices such as mobile phones and MP3 players. But why not make a simple receiver ourselves? There are currently several nice ICs available that contain a (nearly) complete receiver.



projects

technology

info & market

infotainment

elektor international media

Elektor International Media provides a multimedia and interactive platform for everyone interested in electronics.
From professionals passionate about their work to enthusiasts with professional ambitions.
From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment.
Analogue and digital; practical and theoretical; software and hardware.

FEBRUARY 2009 ISSN 1757-0875

Elektor 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 US, PO Box 876, Peterborough NH 03458-0876 USA.
Phone: 603-924-9464, Fax: 603-924-9467 www.elektor.com

Elektor USA magazine commences publication with the January 2009 issue.
Elektor is published 11 times a year with a double issue for July & August.

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

International Editor:
Wisse Hettinga (w.hettinga@elektor.com)

Editor: Jan Buiting (editor@elektor.com)

International editorial staff: Harry Baggen, Thijs Beckers, Eduardo Corral, Ernst Krempelsauer, Jens Nickel, Clemens Valens

Design staff: Antoine Authier (Head), Ton Giesberts, Luc Lemmens, Daniel Rodrigues, Jan Visser, Christian Vossen

Graphic design / DTP: Giel Dols, Mart Schroijen

Managing Director / Publisher: Hugo Vanhaecke

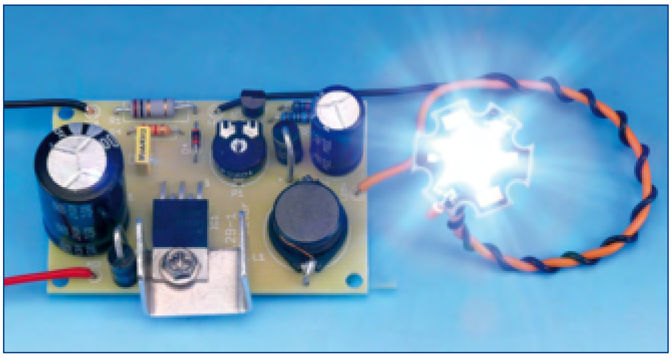
Marketing: Carlo van Nistelrooy, Laurel Humphrey

Customer Services: Sharon LeClair

Subscriptions:
Elektor US, PO Box 876, Peterborough NH 03458-0876 USA.
Phone: 603-924-9464, Fax: 603-924-9467
Internet: www.elektor-usa.com E-mail: custservus@elektor.com

Head Office:
Elektor International Media b.v.
PO Box 11 NL-6114-ZG Susteren The Netherlands
Telephone: (+31) 46 4389444, Fax: (+31) 46 4370161

US Advertising:
Strategic Media Marketing, Peter Wostrel,
1187 Washington St., Gloucester MA 01930 USA.
Phone: 978-281-7708, Fax: 978-281-7706
E-mail: peter@smmarketing.us
Advertising rates and terms available on request.



PLDM LED Driver — DIY and commercial

Dear Editor — I noticed the December 2008 issue's LED bike lamps and clever related step-down current regulator trick: to reduce feedback votage loss — it's been filed!
However with the 60 lux Busch and Muller Cyo available (and legal) with efficient optics, one might as well leave DIY LED bike lamps as extra auxiliary lamps.
I discovered a "£9.74 Nite Ize L.E.D. Upgrade for C & D Cells [LRB-07-PR]" pre-focus bulb with 3-watt LED and electronics: a rectifier and cap would allow easy conversion of lamps: presumably you would need to ensure heatsinking via a metal bulb holder connection to, for example, foil.
It seems to work for 2 to 6 cells so must have a buck-boost converter.
Also, LED domestic bulbs are made by a Phillips division called Lemnis, see www.lemnislighting.com/pharox_led_bulb.html

Alan Bradley (UK)

That's brilliant (pun intended), thanks Alan for letting us know this commercial alternative.

Hi from the USA

The following testimonial about the early days of Elektor was received from our correspondent Ed Dell of Audio Amateur Inc., forwarding an email from Dick Campbell.

"If you are of a certain age or traveled in Europe you will remember the wonderful magazine, "Elektor". I subscribed back when it was translated into English (from the original Dutch) in the UK and shipped to the USA. Then my finances changed and I had to drop the subscription. Later when I tried to find them again the UK publisher had disappeared.
I still have the issues I bought back then and still make reference to them from time to time. Besides the great construction articles they also have very clear explanations of how the circuitry in the articles work. In particular their explanation of how a pseudo random noise generator works is the very best and clearest I have ever found. I have passed that explanation onto one of the best DSP engineers around and he found it useful.
"It is now available in the USA — see www.elektor.com/USA/ for subscription info."
I saw them at the AES in San Francisco and somehow did not stop and subscribe. Thanks for the reminder, I have just sent in my subscription.
"This is a remarkable magazine for DIY or just plain inspirational projects in many areas of electronics."
They also make quality bare circuit boards available at reasonable prices for their projects which makes them very easy to build.

Ray A. Rayburn (USA)

Wiring up (3)

Dear Editor, regarding Phil Pumphrey's letter in Mailbox, October 2008, what he needs is something called a DRO (digital readout). There are many websites on the subject and many homebrew contraptions, including a few ones based on hacked mic even the optical ones. Just ask Google about 'homemade / homebrew dro'.
Please tell Phil that I'll be happy to be of assistance; you may give him my e-mail address.

Jorge L. (Spain)

Thank you Jorge for your kind offer, we'll do the necessary to get you in touch with Phil.

Electrobike — kit supplier

Hi Jan — last November's issue had an interesting article on an electrobike retrofit conversion kit, reportedly sourced via EBay. Unfortunately I was unable to find the kit described. Can you help me, please?

Herbert Pölzer (Germany)

The prototype was bought from Wilkotec's e-bay shop at http://stores/ebay.de.E-bikeTec which was alive and kicking at the time of publishing. Wilkotec also have their own online shop at www.wilkotec.de. Other suppliers of e-bike wheels and kits may be found on ebay.com.

Electrobike — need to know

Dear Elektor — after reading this great article, some questions remain, if I may.
1. did you fit new spokes, reposition the front wheel or convert a children's bike?
2. Any findings on the estimated range?
2. How much should I expect to pay for a battery pack?

Joachim Herbert (Germany)

For the test we had a mountain bike available with 26-inch wheels. However, owing to an error somewhere in the supply chain, the electrobike wheel we received was a 24-inch type (for sure, 26-inch and 28-inch versions are available).
Three series connected 12 V 7.2 Ah batteries represent 30 to 40 minutes worth of bike riding without any pedal assistance. However if the motor is used as a support only, a range of 40-50 kms is not exceptional, depending of course on the terrain and the load. A set of 12-Ah batteries almost doubles the range.
For the experiment we used CSB batteries from the 'EVX' range, which are specially designed for electric wheel powering and

cyclic charging. A 12 V 7.2 Ah battery costs about 45 €, or 60 € for a 12 Ah type.

Jan Visser (Elektor labs)

New Elektor videos on YouTube

Recently added videos on Elektor's YouTube channel include the Messaging Spin Top from the December 2008 issue, and a filmed report of Elektor's Live! event held in the Netherlands on 22 November 2008. An overview of current Elektor videos is found at www.youtube.com/elektorim

where 'im' stands for International Media. Once on the channel you'll be able to watch short films on the Profiler milling machine, Microdrones, the Elektor SMT oven, Ball & Beam, Formant, study trips to China and a few items from the Retronics series. New films will be added to the 'channel' as they become available.

Selecting 'videos' on YouTube and then searching for 'elektor' will return all videos somehow linked to the word 'elektor'. It should be noted that many of these videos are unrelated to our magazine, or to electronics in general.

Vinculum not secure?

Dear Editor — I'd like to thank you for the article 'ATmega meets Vinculum' (*Elektor* November 2008, *Ed.*), which should keep me busy soldering and programming for a few winter evenings. I was surprised however to read the notice on having to remove the USB stick when the Vinculum is switched on or off. As I see it, the Vinculum is not ready for its expected application. After all, it should be perfectly possible to switch an electronic device on or off without suffering data loss. I may have missed something and would be grateful for your advice.

Very likely another error was present at the time the data loss occurred. Whatever the cause, one thing to avoid at all times is supply removal during a write process. Windows even has a special function for this: remove hardware securely.

Burkhard Kainka

Electronic Transformers

Dear Elektor — as a staunch reader since issue # 1 I'd like to provide some information that's supplemental to your article 'Electronic Transformers Revealed and Explained' in the December 2008 issue.

These 'transformers' are actually switch-mode power supplies or, more accurately, 'power auto oscillators' and most of them are based on application note AN528 from STmicroelectronics. Even if bipolar transistors have been replaced by MOSFETs in recent models, the principle remains the same. The application note provides a detailed operation of these supplies, and has good educational value.

Interesting as that may be, if we look at the schematics in detail, it's apparent that these supplies may be modified to act as **dimmers**. The oscillator is triggered by a diac using an RC network. If a potentiometer with value 10 R is inserted in series with R, a dimmer is created with a range close to 0-100%.

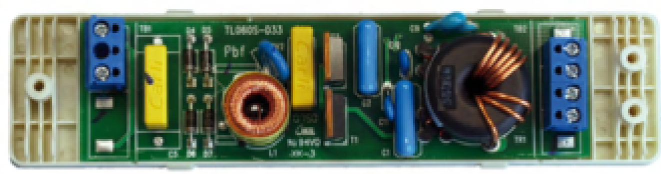
Sure, you have to know a bit about electronics but I would expect Elektor readers to be able to locate the resistor (if necessary, consult the application note), and lift up one of its legs. The resistor value is usually between 100 kΩ and 220 kΩ, depending on the model and the associated capacitor. Do not remove the resistor completely, as it will act as a protection if the pot is set to zero. In the absence of the fixed resistor, a resistance below the original value could well cause the supply to go up in smoke.

An even more exciting modification is to link the potentiometer to a switch. This setup allows a smooth switch-on of the load, which will benefit the life expectancy of the bulbs (effectively preventing the high inrush current when the filament is cold). It is also much more agreeable in the bathroom, particularly in the morning when harsh lights normally greet you there.

Electronics enthusiasts may reap another benefit from these cheap supplies by simply rectifying and smoothing the output voltage, and so make a cheap, powerful, 0 to 15 volts adjustable supply.

Be sure to observe electrical safety, as the potentiometer terminals are connected to the rectified AC mains voltage. A potentiometer with a plastic spindle must be used, preferably in combination with an ABS enclosure.

Alain Caillard (France)



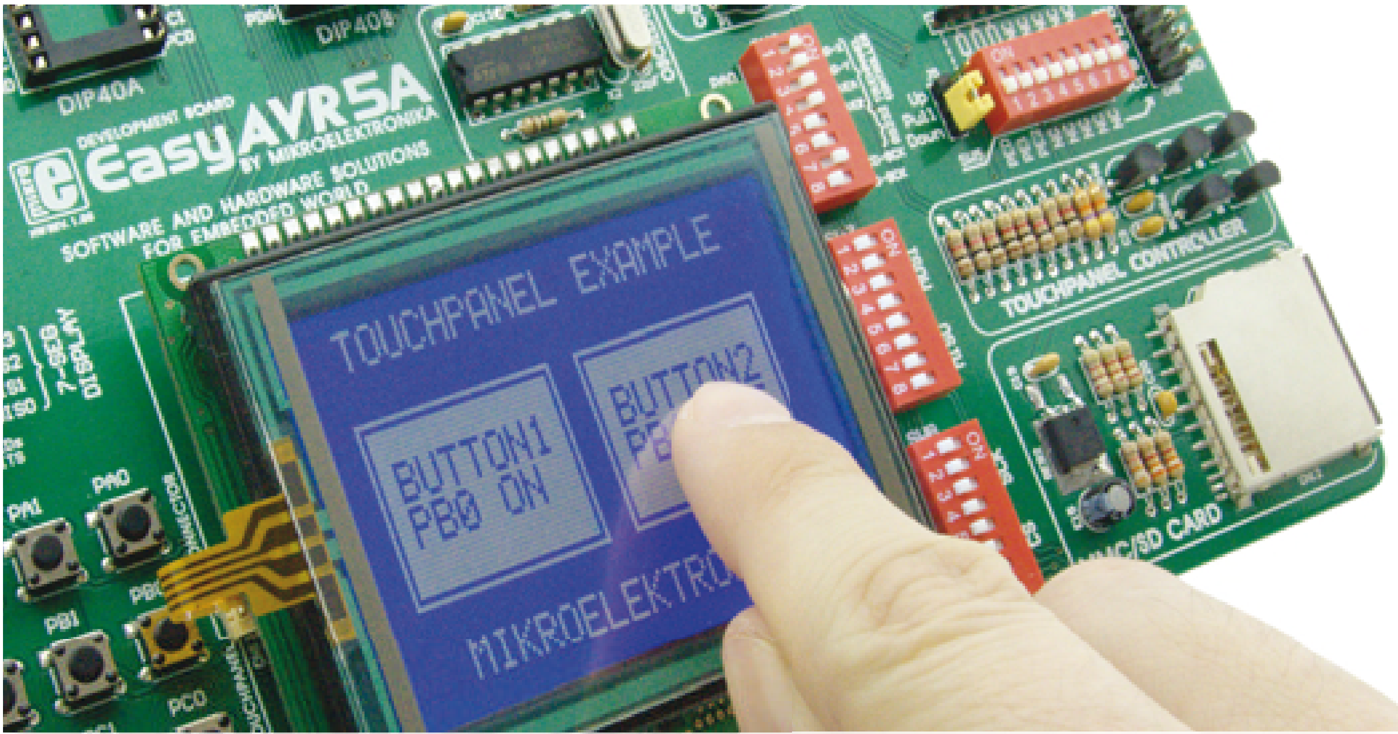
Electronic Transformer for a 12V Halogen Lamp, Fichera, P. and Scollo, R., AN528/0999, STMicroelectronics, 1999. www.st.com/stonline/products/literature/an/3707.pdf

MailBox Terms

- Publication of reader's orrespondence 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.com or
Elektor, The Editor,
PO Box 876,
Peterborough NH 03458-0876,
USA.

OK. Now you need a... TOUCHSCREEN



Do you want your new device to have a simple and intuitive interface? If the answer is YES, then a graphic LCD display with touch panel is the best choice because together they create a Touchscreen (Glcd + Touch Panel = Touchscreen). In that way, with a small number of electronic components you will be able to create an attractive and easy to use device.

By Dusan Mihajlovic
Mikroelektronika Hardware Department

What is a touch panel? A touch panel is a thin, self-adhesive transparent panel placed over the screen of a graphic LCD. It is very sensitive to pressure so that even a soft touch causes some changes on output signal. There are a few types of touch panel. The simplest one is the resistive touch panel which will be discussed here.

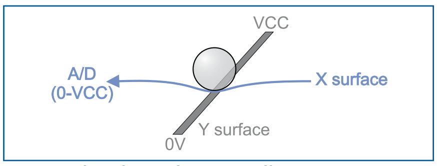
Principle of operation

A resistive touch panel consists of two transparent rigid foils, forming a "sandwich" structure, that have resistive layers on their inner sides. The resistance of these layers usually does not exceed 1Kohm. The opposite sides of the foils have contacts available for use through a flat cable. The process of determining coordinates of the point in which the touch panel is pressed can be broken up into two steps. The first one is the determination of the X coordinate and the second one is the determination of the Y coordinate of the point. In order to determine the X coordinate, it is necessary to connect the left contact on the X surface to ground and the right contact to the power supply. This enables a voltage divider to be obtained by pressing the touch panel. The value of the divider is read on the bottom contact of the Y surface. Voltage can be in the range of 0V to the power supply and depends on the X coordinate. If the point is closer to the left contact of the X surface, the voltage will be closer to 0V. In order to determine the Y coordinate, it is necessary to connect the bottom contact on the Y surface to ground, and the upper contact to power supply.

In this case, the voltage is read on the left contact of the X surface.

Connecting to microcontroller

In order to connect a touch panel to the microcontroller it is necessary to create a circuit for touch panel control. By means of this circuit, the microcontroller connects appropriate contacts of the touch panel to ground and the power supply (as described above) in order to determine the X



Determination of Y coordinate

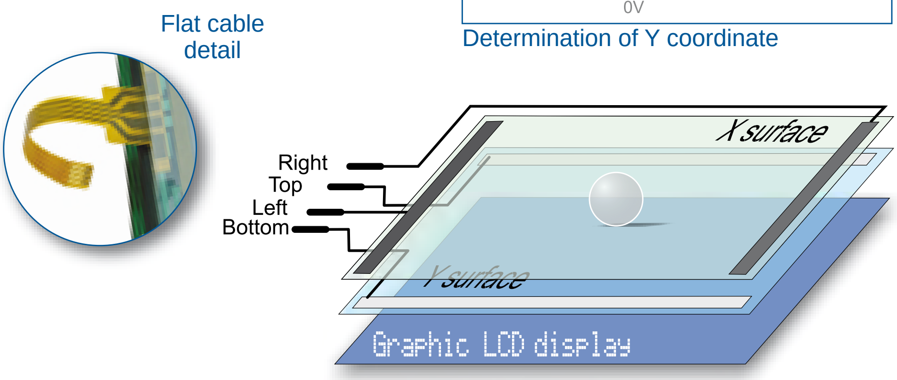
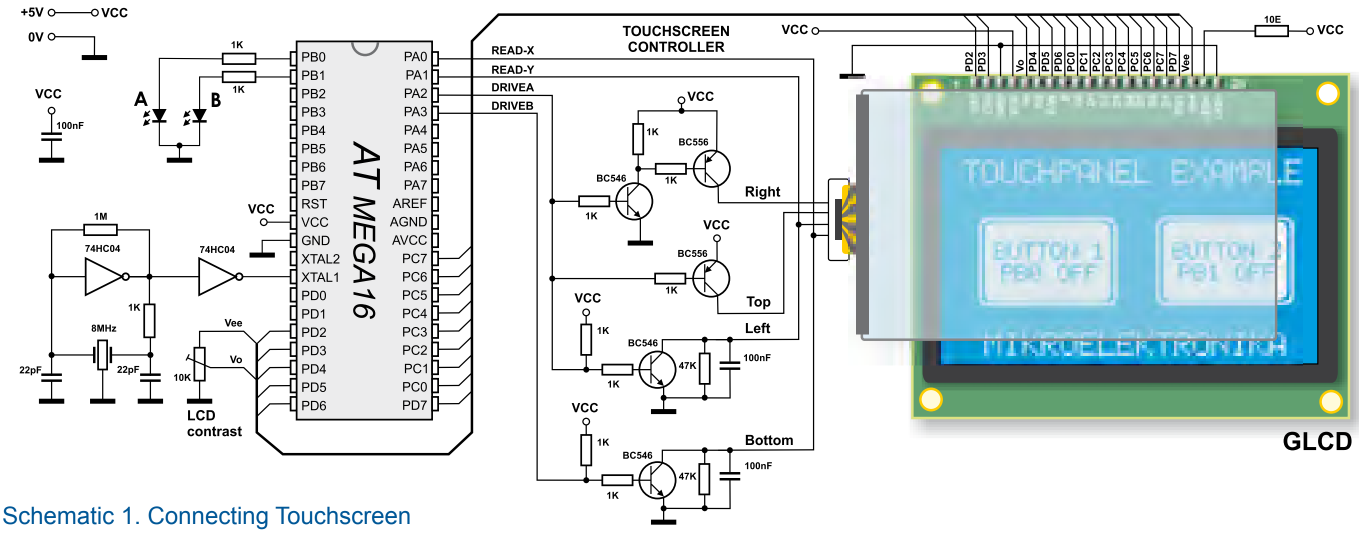


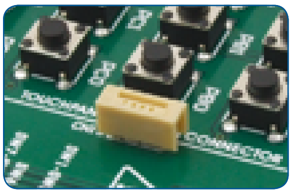
Figure 1. Touch panel internal structure



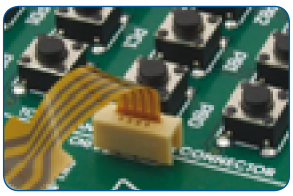
Schematic 1. Connecting Touchscreen

and Y coordinates (Refer to Schematic 1). The bottom contact of the Y surface and left contact of the X surface are connected to the microcontroller's A/D converter. The X and Y coordinates are determined by measuring voltage on these contacts, respectively. The software consists of writing a menu on graphic LCD, turning the circuit for touch panel control on/off (driving touch panel) and reading the values of A/D converter which actually represent the X and Y coordinates of the point.

Once the coordinates are determined, it is possible to decide what we want the microcontroller to do. For the purpose of illustration, let us examine Example 1. It explains how to turn on/off two digital microcontroller pins, connected to LED diodes A and B, using a display and a touch panel.



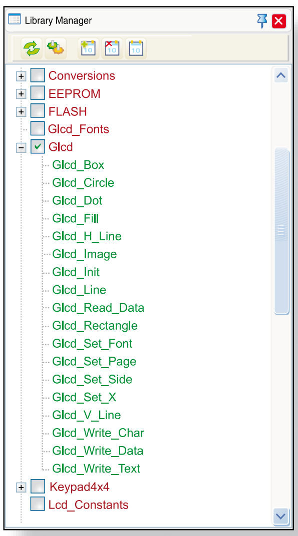
Flat cable on-board connector before...



...and after connecting touch panel.

Considering that the touch panel surface is slightly larger than the surface of the graphic LCD, in case you want greater accuracy when determining the coordinates, it is necessary to perform the software calibration of the touch panel.

Functions used in the program



ADC_Read()	Read analog value
Delay_ms()	Delay
Glcd_box()	Draw filled box*
Glcd_circle()	Draw circle
Glcd_Dot()	Draw dot
Glcd_Fill()	Delete/fill display*
Glcd_H_Line()	Draw horizontal line
Glcd_Image()	Import image
Glcd_Init()	LCD display initialization*
Glcd_Line()	Draw line
Glcd_Read_Data()	Read data from LCD
Glcd_Rectangle()	Draw rectangle*
Glcd_Set_Font()	Select font*
Glcd_Set_Page()	Select page
Glcd_Set_Side()	Select side of display
Glcd_Set_X()	Determine X coordinate
Glcd_V_line()	Draw vertical line
Glcd_Write_Char()	Write character
Glcd_Write_Data()	Write data
Glcd_Write_Text()	Write text*

* Glcd library functions used in the program

mikroC PRO for AVR® library editor with ready to use libraries such as: Ethernet, CAN, SD/MMC etc.

NOTE: Code for this example written for AVR® microcontrollers in C, Basic and Pascal as well as the programs written for PIC® and dsPIC® microcontrollers can be found on our web site **www.mikroe.com/en/article/**

Example 1: Program to demonstrate touchscreen operation

```
// Glcd module connections
char GLCD_DataPort at PORTC;                char GLCD_DataPort_Direction at DDRC;

sbit GLCD_CS1 at PORTD.B2;                  sbit GLCD_CS1_Direction at DDRD.B2;
sbit GLCD_CS2 at PORTD.B3;                  sbit GLCD_CS2_Direction at DDRD.B3;
sbit GLCD_RS  at PORTD.B4;                  sbit GLCD_RS_Direction  at DDRD.B4;
sbit GLCD_RW  at PORTD.B5;                  sbit GLCD_RW_Direction  at DDRD.B5;
sbit GLCD_EN  at PORTD.B6;                  sbit GLCD_EN_Direction  at DDRD.B6;
sbit GLCD_RST at PORTD.B7;                  sbit GLCD_RST_Direction at DDRD.B7;
                                            // End Glcd module connections

sbit DRIVE_A at PORTA.B2;   sbit DRIVE_A_Direction at DDRA.B2;   // Touch Panel module connections
sbit DRIVE_B at PORTA.B3;   sbit DRIVE_B_Direction at DDRA.B3;   // End Touch Panel module connections

long x_coord, y_coord, x_coord128, y_coord64;                    // scaled x-y position

unsigned int GetX() {       //reading X

 DRIVE_A = 1;               // DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
 DRIVE_B = 0;               // DRIVEB = 0 (BOTTOM drive off)
 Delay_ms(5);
 return ADC_Read(0);        // READ-X (BOTTOM)
}

unsigned int GetY() {       //reading Y

 DRIVE_A = 0;               // DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
 DRIVE_B = 1;               // DRIVEB = 1 (BOTTOM drive on)
 Delay_ms(5);
 return ADC_Read(1);        // READ-X (LEFT)
}

void main() {

 DRIVE_A_Direction = 1;                    // Set DRIVE_A pin as output
 DRIVE_B_Direction = 1;                    // Set DRIVE_B pin as output
 PORTB.B0 = 0;
 DDRB.B0 = 1;                              // Set PB0 pin as output (Default value 0)
 PORTB.B1 = 0;
 DDRB.B1 = 1;                              // Set PB1 pin as output (Default value 0)

 Glcd_Init();                              // Initialize GLCD
 Glcd_Fill(0);                             // Clear GLCD
 Glcd_Set_Font(font5x7, 5, 7, 32);         // Choose font,
 Glcd_Fill(0);

 Glcd_Write_Text("TOUCHPANEL EXAMPLE",10,0,1);
 Glcd_Write_Text("MIKROELEKTRONIKA",17,7,1);

 Glcd_Rectangle(8,16,60,48,1);             //Display Buttons on GLCD:
 Glcd_Rectangle(68,16,120,48,1);
 Glcd_Box(10,18,58,46,1);
 Glcd_Box(70,18,118,46,1);
 Glcd_Write_Text("BUTTON1",14,3,0);
 Glcd_Write_Text("PB0 OFF",14,4,0);
 Glcd_Write_Text("BUTTON2",74,3,0);
 Glcd_Write_Text("PB1 OFF",74,4,0);

 while (1) {                               // read X-Y and convert it to 128x64 space

   x_coord = GetX();
   y_coord = GetY();
   x_coord128 = (x_coord * 128) / 1024;
   y_coord64 = 64 -((y_coord *64) / 1024);

                                           //if BUTTON1 is selected

   if ((x_coord128 >= 10) && (x_coord128 <= 58) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
    if(PORTB.B0 == 0) {
     PORTB.B0 = 1;
     Glcd_Write_Text("PB0 ON ",14,4,0);
    }
    else {
     PORTB.B0 = 0;
     Glcd_Write_Text("PB0 OFF",14,4,0);
    }
   }

                                           //if BUTTON2 is selected

  if ((x_coord128 >= 70) && (x_coord128 <= 118) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
    if(PORTB.B1 == 0) {
     PORTB.B1 = 1;
     Glcd_Write_Text("PB1 ON ",74,4,0);
    }
    else {
     PORTB.B1 = 0;
     Glcd_Write_Text("PB1 OFF",74,4,0);
    }
   }
   Delay_ms(100);
 }
}
```

Written in compiler mikroC PRO for AVR

Saturday, February 21, 2009, from 10:00 am to 3.30 pm.

Masterclass High-End Valve Amplifiers

Birmingham City University, Technology Innovation Centre.
Presenter: Menno van der Veen, MSc.

In this Masterclass Menno van der Veen will examine the predictability and perceptibility of the specifications of valve amplifiers. Covered are models that allow the characteristics of valve amplifiers to be explored up to the limits of the audible domain from 20 Hz to 20 kHz. This then leads to the minimum stability requirements that the amplifier has to satisfy. The coupling between output valves and output transformer are also modelled. This gives new insight into a unique type of distortion: Dynamic Damping Factor Distortion (DDFD). Negative feedback is often used in amplifiers. What is the optimum and what are the audible consequences? The correct amplification of micro details is explained, based on new research, and new models about this are presented.

The course fee is £ 160, including handout, certificate and lunch. Elektor subscribers are entitled to a 5% discount. Register now, seating capacity is strictly limited.

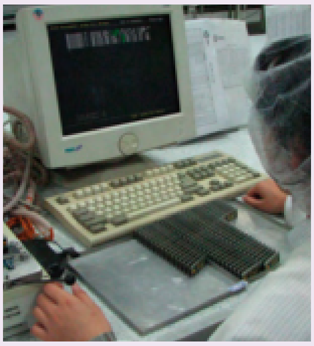
Further information and registration at www.elektor.com/events

April 3–12, 2009.

Study trip: Visit China with Elektor

Tour host: Margriet Debeij (with assistance from local guides and interpreters).

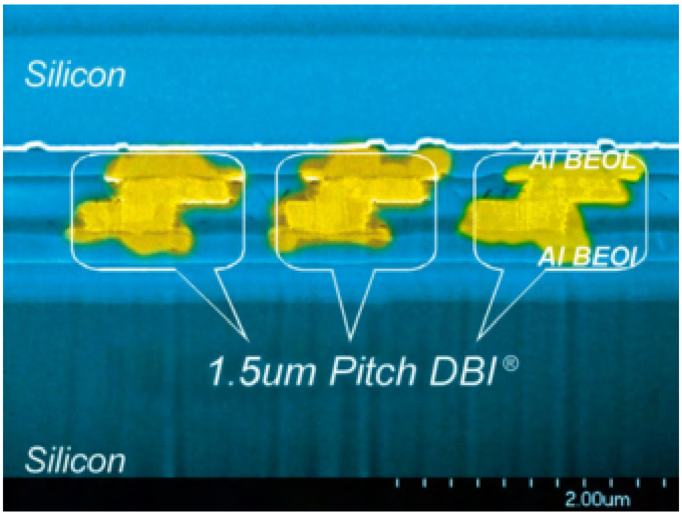
Elektor's third study trip to China is planned for 3-12 April 2009. And you can join us! During this 10-day trip we will visit the China Electronics Fair in Shenzhen, a professional industrial electronics fair with an area of no less than 60,000 m^2. We will also pay at least one visit to the well-known 'electronics high street' in Shanghai. As the name suggests, this street is entirely dedicated to electronics shops, each vying to be the largest. In addition, a variety of interesting company visits are on the itinerary (with a tour of the production department). We are also organising a business conference where you can obtain a wealth of information about doing business (and how not to do business) in China. We put all the do's and don'ts in a tidy list for you. Naturally, there's also time for culture. We will visit the Bund, French Confession and the Shanghai TV tower. There's also a Shanghai sightseeing tour planned.
As with the first and second 'Elektor goes to China' tours, this edition will again be blogged using the Elektor website.



Further information at www.elektor.com/china-trip

Introducing Jeremy, the ultimate 3D IC customer

The future success of the semiconductor industry will depend on how successful it is in implementing 3D IC technology to meet the demands



of a 17-year-old consumer named Jeremy, according to Ziptronix CEO Dan Donabedian. In a presentation at the fifth annual The future success of the semiconductor industry will depend on how successful it is in implementing 3D IC technology to meet the demands of a 17-year-old consumer named Jeremy, according to Ziptronix CEO Dan Donabedian. In a presentation at the fifth annual 3D Architectures for Semiconductor Integration and Packaging conference today in Burlingame, CA, Donabedian described the role his company's revolutionary bonding technology will play in the emerging 3D IC supply chain.

According to Donabedian's presentation, one of the crucial elements for successful implementation of 3D IC technology is a reliable, economical, low temperature oxide bonding process that will enable true 3D integration of semiconductors. "A high throughput bonding process that achieves true metal-to-metal interconnect without requiring high temperature or compression has long been seen as the 'missing link' in the 3D IC supply chain that is now taking shape in the semiconductor industry," he explained.

"Ziptronix technology will be one of the key factors in making 3D IC integration a mainstream semiconductor technology."

Donabedian sees Jeremy (and the next generation of electronics consumers) as ready to buy a host of integrated applications that will be enabled by 3D IC technology, including:

- Mobile phones with high resolution digital cameras and increased functionality
- Ultramobile, low power, lightweight PCs
- Interactive handheld gaming devices with projection capabilities
- Embedded pico-projectors in a variety of portable multimedia devices
- Advanced automotive sensors (lane change/collision warning)
- Sophisticated medical imaging systems

The Ziptronix processes can be implemented throughout the present semiconductor supply chain - by the OEMs/IDMs; by fabless and 'fab-lite' companies; by the major foundries; by semiconductor tool manufacturers and EDA vendors; and by the OSAT houses.

www.ziptronix.com

(080965-VIII)

Elektor USA on newsstand distribution

Following a three-month introduction period and the rapid growth of its subscriber base, Elektor USA is now progressing to newsstand sales, allowing 'folks' to browse a copy before buying. Starting with this February 2009 issue, the magazine is stocked by Borders bookshops in selected cities in the USA. This may be followed by other store chains.
Elektor USA joins the successful English, Dutch, Spanish, French, German, Italian, Portuguese and Brazilian editions centrally produced by Elektor International Media, with websites to match.
American and Canadian readers originally subscribed to the European Elektor can now subscribe on-line using the specially created USA landing page, which contains an offer they will find hard to refuse. The expanded Elektor USA website is expected to come online around 1 February 2009.

www.elektor-usa.com

MontaVista: big endian support for ARM11 family

MontaVista® Software, Inc. announced that MontaVista Linux now provides support for the ARM1176JZ-S™ and ARM1176JZF-S™ processors, including the first ever 'big endian' support option for the ARM11™ family in a Linux environment. By supporting these ARM11 family processors, MontaVista gives consumer device manufacturers a commercial-quality Linux implementation along with tools to reduce time-to-market and development costs for custom SoC designs.

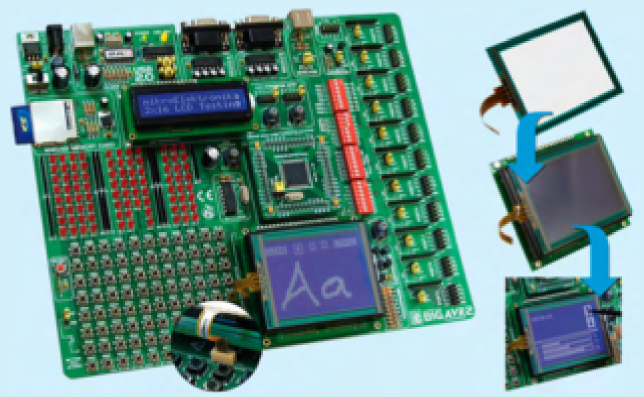
MontaVista provides a commercial quality Linux platform for mobile device manufacturers. Its broad hardware support, dedication to quality, and full support has made it the undisputed leader in the mobile Linux market. Its advanced power management, fast startup, and advanced connectivity provide the features mobile device manufactures require. In addition to powering a majority of today's Linux handsets, MontaVista Linux is the only Linux to demonstrate support of and integration with all major Linux mobile software stacks, is the only mobile Linux certified as being ready for IPv6, provides support for new mobile device processors from Freescale® Semiconductor, Intel®, Texas Instruments and others and was awarded "Best Software Innovation of the Year" for 2007 by EDN.

www.mvista.com

(081040-IX)

BIGAVR®2 Board

MikroElektronika recently introduced a new development tool for AVR® microcontrollers. The new BIGAVR®2 supports 8, 64-pin and 100-pin AVR (TQFP package) AVR® and gives designers an easy to use platform to try a multitude of designs. The BIGAVR®2 comes complete with everything you need to learn, experiment, design and program with AVR®. The BIGAVR®2 includes a TouchPanel controller, so you can easily add additional input to prototype devices.
Like its predecessor, BIGAVR®2 has an ultra fast on-board USB 2.0 programmer. The new board delivers an extensive list of features, including:
- TouchPanel controller
- Buttons for changing states of all pins
- LEDs for displaying states of all pins
- Support for Character LCD
- Support for Graphic LCD
- JTAG connector
- RS232 communication
- Support for PS/2
- 4.096 Vref for ADC
- Selectable USB or external power supply
- MMC/SD Card Slot
- IDC10 connectors for further expansion
- Pull up/pull down selection of each pin



Every feature of the board is supported by examples written in mikroC PRO, mikroPascal and mikroBasic compilers for AVR®.
The BIGAVR®2 Board is available for purchase on the mikroelektronika website and through authorized distributors.

www.mikroe.com

(081040-V)

CapSense(tm) proximity sensing achieves 25 cm detection range

Cypress Semiconductor Corp. announced that its proximity detection solution enables a best-in-class proximity detection range of 25 cm. Proximity sensing enables new usage models where direct touch is not required, instead detecting a finger or another object as it approaches the device. The solution allows for enhanced industrial designs that only expose interfaces or buttons when necessary and provides incremental power savings by activating an interface only when needed.
To simplify use of this proximity sensing functionality, Cypress also announced a free online tutorial that shows how to develop proximity sensing detectors without

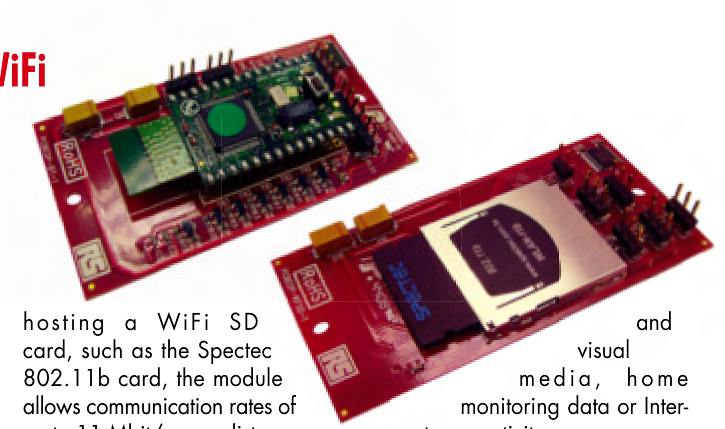
[continued on page 16]

writing or debugging a single line of C or Assembly code using Cypress's PSoC Designer(tm) embedded design tool. The 30-minute interactive video entitled "Building a Proximity Detector Using PSoC Designer" is available at the website below.
The new video includes an introduction to proximity sensing with a demonstration of the detection range of the CapSense solution, and it features Cypress member of technical staff Dave Van Ess presenting the design of a proximity sensing interface using PSoC Designer software via the PSoC FirstTouch(tm) Starter Kit. The USB thumbdrive kit can also be used for touch-sensing, temperature-sensing and light sensing-designs. The CY3270 PSoC FirstTouch Starter kit is available for US$ 29.95. The follow-on CY3271 PSoC FirstTouch Starter Kit with CyFi(tm) low-power RF enables proximity sensing as well, and is available for US$ 69.95 (all prices plus P&P). CapSense Sliders and CapSense Buttons evaluation kits are available from Elektor at a special discounted price.

www.cypress.com/proximitysensing
www.cypress.com/ProximityDemo
www.cypress.com/FirstTouch

(081040-I)

New EDP modules from RS: Bluetooth and WiFi



Following Elektor's review of the RS Components Embedded Development Platform (EDP) in the December 2008 issue, RS Components have released two new modules for the system.

The new **Bluetooth Application Module** (EDP-AM-BT54) allows you to replace cable communication in a variety of applications, with ease. The module combines a PIC18LF6722 microcontroller and FlexiPanel's LinkMatik Bluetooth radio, allowing designs to synchronise with any Bluetooth enabled device within a 100 m range and allow data rates of up to 2.1 Mbit/s. The module is fully FCC / CE / IC certified and operates in the globally unlicensed Industrial, Scientific & Medical (ISM) 2.4 GHz short-range radio frequency bandwidth, therefore reducing the need for licensing and certification of your final design.

The new **WiFi Application Module** (EDP-AM-WFSD**)** provides a solution for designs requiring higher throughput over larger distances than Bluetooth. Through hosting a WiFi SD card, such as the Spectec 802.11b card, the module allows communication rates of up to 11 Mbit/s over distances of up to 150 m. The modules provides a robust and highly secure wireless medium for applications such as the transmission of audio and visual media, home monitoring data or Internet connectivity.

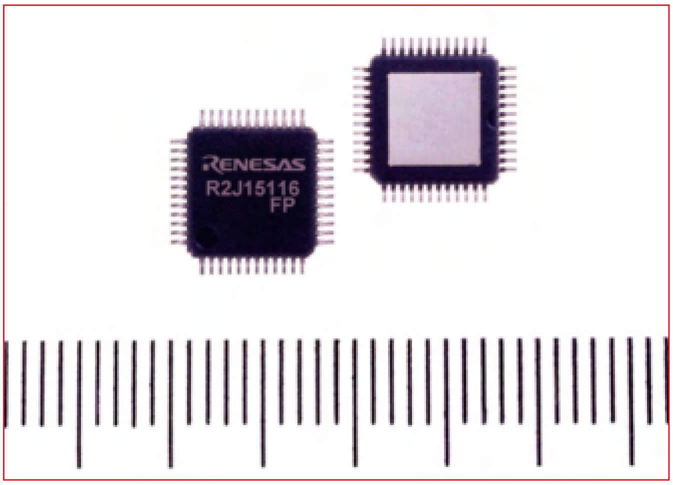
http://rswww.com/edp

(081040-VI)

Compact, high-performance digital amplifier with IIS digital audio signal input

Renesas Technology Europe recently announced the R2J15116FP, a compact, high-performance digital amplifier supporting IIS*1 digital audio signal input with a built-in 24-bit audio DSP. It is intended for use in thin flat-screen TVs employing LCD or plasma display panels.

The device, contained in a compact 7 × 7 mm package, accepts IIS digital audio signal input, performs signal processing using a built-in 24-bit audio DSP, and amplifies the result with a digital amplifier to deliver stereo output to two speakers at up to 15 watts per channel. Typically, thin flat-screen TVs require detailed tuning of their audio systems for different markets and screen sizes. This processing is usually performed in the preceding stage of this digital amplifier by the main DSP, which mainly handles tasks such as surround processing and decoding. This imposes a significant processing load. In the R2J15116FP, in contrast, processing for the volume control and the parametric equalizer*2 for optimizing the speaker characteristics is done by the built-in audio DSP. This reduces the load on the main DSP and makes it possible to generate a more varied and higher quality sound environment for video applications.



The R2J15116FP employs a feedback circuit to stabilize the output level when the power supply voltage fluctuates, an approach with a proven track record in products employing analog input. In addition to the volume control and seven-band parametric equalizer, there is a two-band tone control, a loudness function useful for low- and high-range expansion compensation, a power limiter function that allows setting of a user-defined maximum output level, and a dynamic range control (DRC) function that lowers the signal amplitude without distortion at high volume levels and raises it at low volume levels. Settings for all of these functions can be made flexibly from the MCU via the I^2C bus.

The R2J15116FP uses a compact 48-pin HTQFP package, requires no heat sink, and delivers maximum stereo output of 15 watts per channel.

http://eu.renesas.com/

(081040-II)

Renewable Energy Education set

Horizon announces its new and comprehensive Renewable Energy Education set, the latest and final development in its line of clean energy science sets designed for discovery and inventiveness in classrooms, science fairs, or at home.

Similar to a futuristic chemistry set, this new kit is Horizon's most comprehensive science kit covering all aspects of wind, solar, water and hydrogen elements in a modular form for numerous experiments. The kit explores clean energy from wind or solar power, hydrogen energy storage, and distributed power generation using fuel cells. The kit includes a miniature wind turbine, a solar cell, an electrolyzer, a fuel cell, water-based hydrogen and oxygen storage tanks, a small electric motor and an LED light module. All items are set on modular base with interconnectable wiring and tubing.

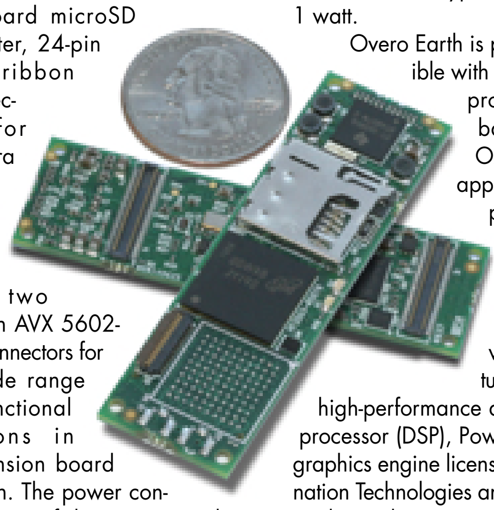
The Renewable Energy Education set introduces fuel cell, solar cell and wind turbine technology to students, aspiring scientists, teachers and engineers investigating clean energy concepts and technologies.

This new set is part of Horizon's extensive line of clean energy science education kits.

www.horizonfuelcell.com

(081040-VII)

Gumstix: open source, Linux compatible, lowest cost ARM platform



At only 17×58×4.2 mm in size, the Overo™ Earth motherboard from Gumstix gives open source innovators access to the industry's highest performance, generally available ARM® based platform in the tiniest, lowest cost Linux computer available based on the Texas Instruments (TI) OMAP 3503 applications processor. With Overo Earth, the open source community can freely innovate in a vast range of application areas using this tiny computer.

Nearly 40 percent smaller than existing Gumstix motherboards, the Overo Earth motherboard runs Linux kernel 2.6.27 or higher. Linux developers can also take advantage of the 256 MB low power DDR RAM, 256 MB NAND flash, on-board microSD adapter, 24-pin flex ribbon connector for camera control signals and two 70-pin AVX 5602-14 connectors for a wide range of functional options in expansion board design. The power consumption of the Overo Earth motherboard is typically less than 1 watt.

Overo Earth is plug compatible with future Overo products to be based on TI's OMAP35x applications processors, such as the superscalar OMAP3530 which features the ARM, high-performance digital signal processor (DSP), PowerVR SGX™ graphics engine licensed by Imagination Technologies and multimediarich accelerators.

Leveraging the laptop-like performance of the OMAP3503 processor, the Overo Earth motherboard takes advantage of the 600 MHz ARM Cortex™-A8 processor and integrated peripherals. This Cortex-A8 processor achieves an additional 4x performance improvement over the 300 MHz ARM9 through its superscalar architecture, which allows implementation of instruction-level parallelism within a single processor.

The Overo Earth motherboard is now available for purchase for US$ 149 (plus P&P).

www.gumstix.com

(081040-III)

Free CAD

online design tools

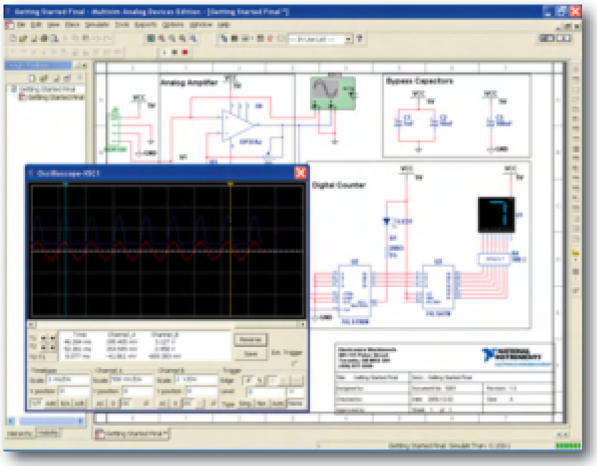
Clemens Valens

Clemens Valens (Elektor France Editorial)

Several electronic component manufacturers now offer design tools online or for free download. We registered – often an obligatory step – on a dozen or so websites to see just what's being offered and what it's like.

Analog Devices — www.analog.com

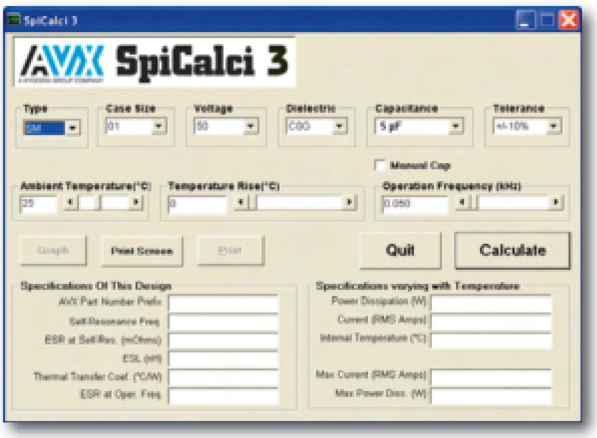
The Analog Devices website offers a whole range of interactive design tools, with names that all start with **ADIsim**. For example, there's **ADIsimPower**, a simulator for power supplies – or DC-DC convertors, to be more precise. Analog Devices also offers a resistance bridge calculator, an aid for dimensioning photodiode amplifiers, and a tool for calculating analogue filters.
There are also tools on free download, like **SRD Design Studio**, a tool for simulating short range radio systems, and **Multisim for Analog Devices**, a monster at 183 MB (v 10.0.0).
After running this program, we downloaded the *Getting Started* project, which is not actually mentioned anywhere, but can be found at the bottom of the directory where the program has been installed. This example shows a single-digit digital counter using the output of an amplifier amplifying a 1 kHz signal as its clock. Clicking the *Run* button at the top right of the screen starts the simulation. Double-clicking on the oscilloscope icon opens a window displaying a two-channel digital oscilloscope. It's all very pretty.
Apart from some models of certain common components,



Multisim only handles Analog Devices components, and it's not possible to modify them or create new ones.

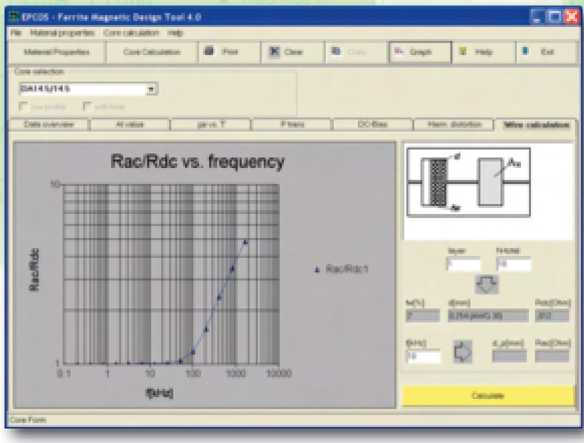
AVX — www.avx.com

AVX offers several tools for selecting its capacitors. We tried **SpiCALCI 3.0**, a tool for dimensioning capacitors for switch-mode power supplies. Unfortunately the program does not offer a help function so a good awareness is required of the various capacitors in the AVX product range. After selecting a capacitor type, working voltage, dielectric, etc. the program displays the capacitor's resonance frequency and dissipation at a user selected temperature. A graph can be produced showing impedance response and ESR.



Epcos www.epcos.com

This passive components manufacturer offers several tools, including **AlCap** for dimensioning aluminium electrolytic capacitors, and the **Ferrite Magnetic Design Tool**, used for evaluating Epcos magnetic materials and for performing certain calculations pertaining to inductances and coils.

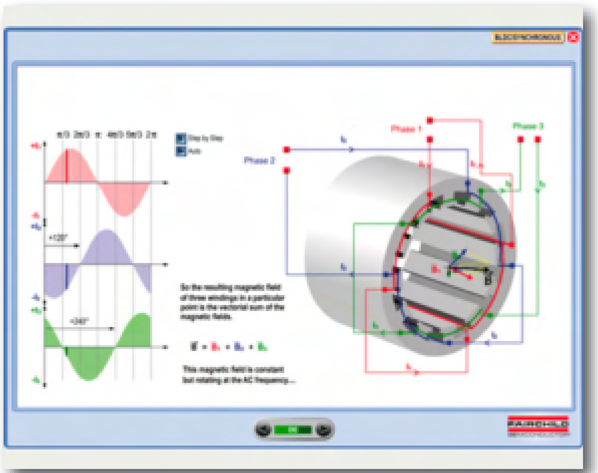


Fairchild www.fairchildsemi.com

FETBench is Fairchild's online design tool, and comprises three parts:

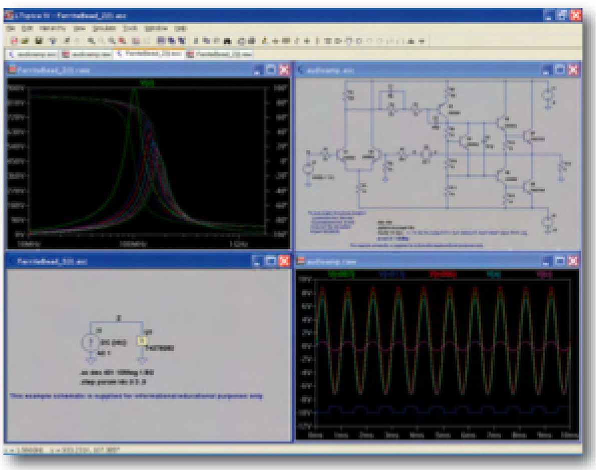
- **Device Analysis** lets you evaluate a component with customized graphs;
- **Application Analysis** is a switch-mode power supply simulator;
- **Thermal Analysis** lets you specify a board and position active components from Fairchild on it. Then the tool shows a coloured diagram with the temperature distribution across the surface of the board.

But there's even better from Fairchild: **SPM Tool**, a very fine piece of software that is not only able to simulate motor drivers, but also contains all the theory of motors in the form of interactive animations. Not to be missed! Before you can download the software, you are asked for a username and password. These are the same ones you used when you signed on.
The same goes for **PFC Toolkit**, a tool for exploring phase correction techniques.



Linear Technology www.linear.com

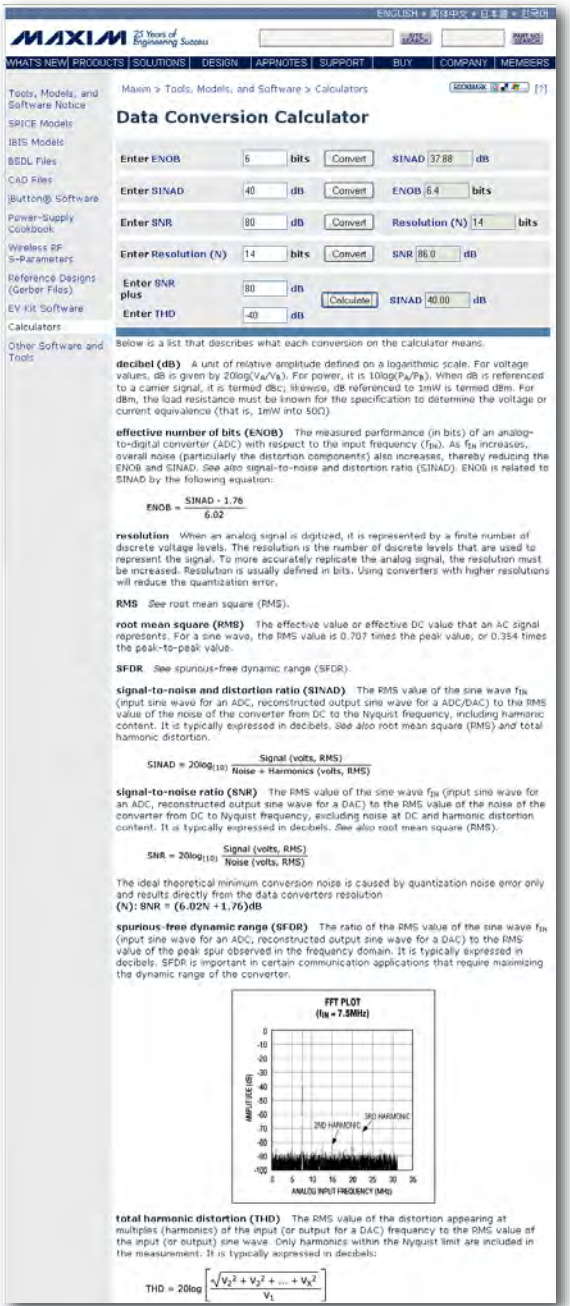
LTSpice IV is the current name of the SPICE simulator from Linear Technology, originally launched a few years ago as **SwitcherCAD**, principally for simulating switch-mode power supplies. Users soon found that the tool is in fact a souped-up SPICE that lends itself perfectly to other simulations. LTSpice is fairly easy and intuitive to use. You can quickly draw a circuit diagram and simulate it equally swiftly. OK, so maybe the graphics aren't the best in the world – but this is a very powerful tool and is regularly updated.
Apart from LTSpice, the site also offers **FilterCAD** and **BodeCAD** and a number of tools for evaluating certain Linear Technology components.



Maxim www.maxim-ic.com

Maxim does not offer any very sophisticated online design tools, but there are twenty or so useful calculators available. Some of them are dedicated to a specific Maxim component, but not all of them, and there are several you may find useful. The interesting thing is that Maxim gives references and detailed explanations that let you delve into the theory applied practically in the calculator.

The **Power-Supply Cookbook** lets you download, for each regulator in the Maxim catalogue, a PDF file containing a circuit diagram and component list for several configurations of the chip. It is also possible to download all the files in one go, in the form of a 9 MB PDF, which does save time. It's a bit like the old databooks, with loads of ideas for circuits.



Microchip www.microchip.com

Mindi is the analogue simulator from Microchip based around (X)SPICE and developed by SIMetrix. Mindi is supplied with lots of component models – we counted around 4,500 – which makes entering the circuit very convenient. After selecting a component, you can modify its parameters, which in most cases comes down to choosing one of the models provided.

Another free design tool from Microchip is **FilterLab2**, which lets you simulate Chebychev, Butterworth, and Bessel type filters. In a quite user-friendly environment, you can choose your filter, and the software plots the characteristic curves. It also offers an implementation of the filter, based on Sallen & Key or MFB (multiple feedback) sections. Several ways of designing the filter are offered. Once you are satisfied with the theoretical performance of the filter, you have immediately available a circuit diagram and a SPICE model.

Murata www.murata.com

Five design tools are available for download from Murata, for capacitors, EMC filters, and thermistors. We tried out the **Murata EMI Filter Selection Simulator** (version 3.6.0). This tool lets you dimension EMC filters for transmission lines. Several configurations are supported and users can set the circuit parameters in detail. Once these parameters have been entered, a click on the *Start Simulation* button is all it takes to have access to the results in the form of diagrams. In short, a fine, easy-to-use tool.

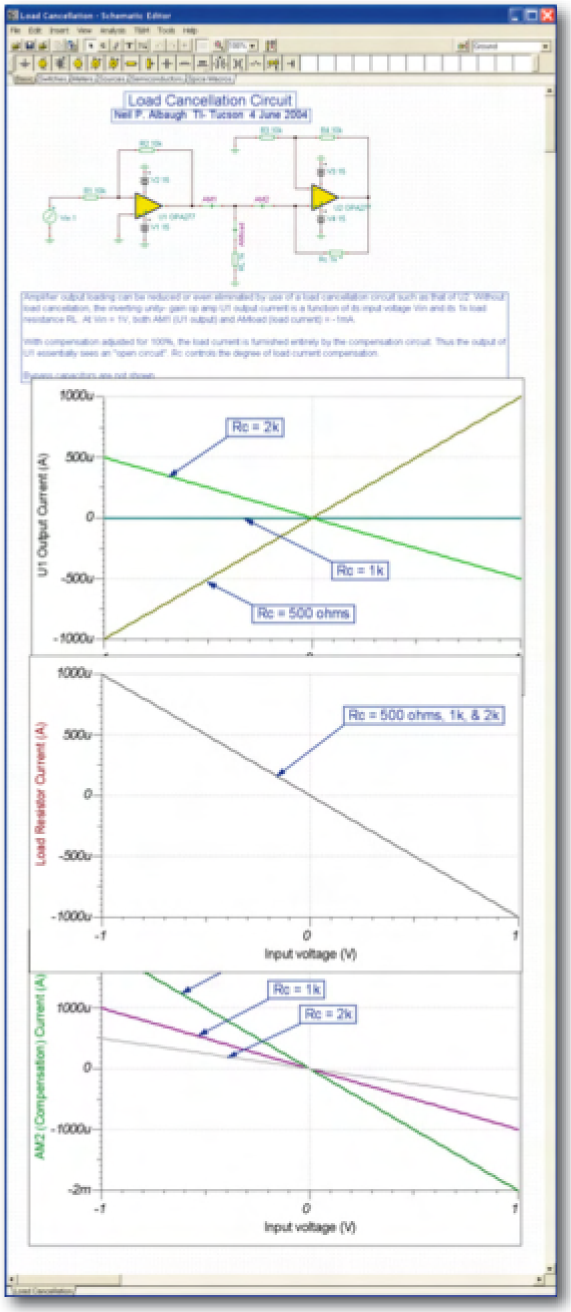
National Semiconductor www.national.com

National Semiconductor has been offering its **WEBENCH** simulator for some years now, originally for simulating power supplies, but now the tool also helps with designing filters, phase-locked loops, LED drivers and even audio amplifiers. In less than ten minutes, for example, we were able to design and simulate an ADC with anti-aliasing filter. Remember to enable pop-ops, otherwise you won't see all the results of the simulations.

NXP

www.nxp.com

SimPort brings together the online simulators from NXP, developed, like Mindi from Microchip, by SIMetrix/SIMPLIS. In fact, there are just two: **Active Datasheet** and **Buck Designer**. Before it is possible to use them, you need to register, just like everywhere else. Nothing so special about that, you may say – but you do need to have good eyesight! If you're not yet registered, you need



to follow the link at the top right of the page which lets you choose between the two simulators. In *Active Datasheet*, you can select a MOSFET from NXP and generate for yourself the curves that are usually found on a data sheet. The difference from a data sheet is that here it's up to you to specify the values for which the curves are generated. *Buck Designer* is a tool for simulating a step-down converter The tool offers you a block diagram and a detailed circuit diagram, and generates diagrams where you can enlarge certain zones and measure certain parameters.

STMicroelectronics www.st.com

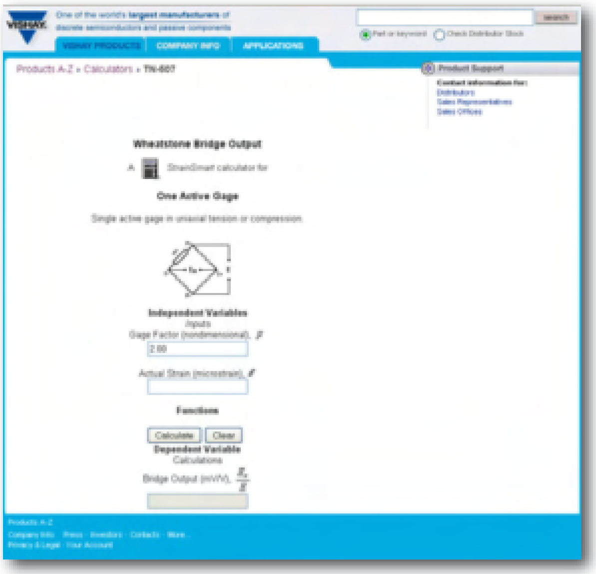
Tools from ST, online and to download. For the online tools, an audio amplifier simulator and a switch-mode PSU simulator, you need to install the Java runtime environment (JRE) first. Then the tools will work… more or less, at least, on my test computer.
On the other hand, **ST Lighting Designer** is a Flash tool for dimensioning fluorescent lighting that is worth going out of your way for. A designer must have spent a lot of time on the graphics interface – you'll either like it or you won't – and the tool works wonderfully well. Once you have realized that it's not possible to have a solution with all the options enabled, the tool outputs the circuit, the components list, and some graphs. You can then fine-tune loads of parameters.

Texas Instruments www.ti.com

TINA-TI is the free simulator for downloading from Texas Instruments and is able to use SPICE models. TINA-TI is one part of TINA, a complete CAD tool with circuit entry, PCB design, and simulator.
SwitcherPro is Texas Instruments' switch-mode PSU simulator. For some months now, this simulator has also been available online. As usual, you have to register before you can use it. Once you are connected to the simulator, it can be run in two ways: from a component or from specifications.

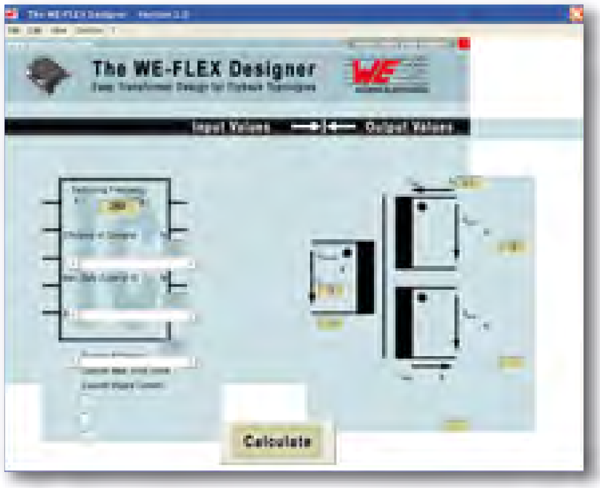
Vishay www.vishay.com

ThermaSim is a thermal simulator for PCBs. It's not terribly intuitive to use, but after a bit of effort you can manage to get the tool to accept your parameters, and it will then send the results by e-mail – rather original, but not very quick. And then when the message does eventually arrive, you find that there was a problem and the simulation has failed. "Please correct and start again", the e-mail reads. Sure, of course…
Apart from ThermaSim, Vishay also offers several basic calculators, like for example the TN-507 calculator for strain gauges.



Würth Elektronik www.we-online.com

Two tools are available from the Würth website: **WE Inductor Selector 1.0** and **WE Flex Designer 2.0**. The latter is used to select a transformer for flyback-type switch-mode power supplies. The tool isn't exactly pretty to look at, at least on the test PC, since all the buttons and boxes were displaced, however a single click on the *Calculate* button is all it takes to generate a transformer reference number.

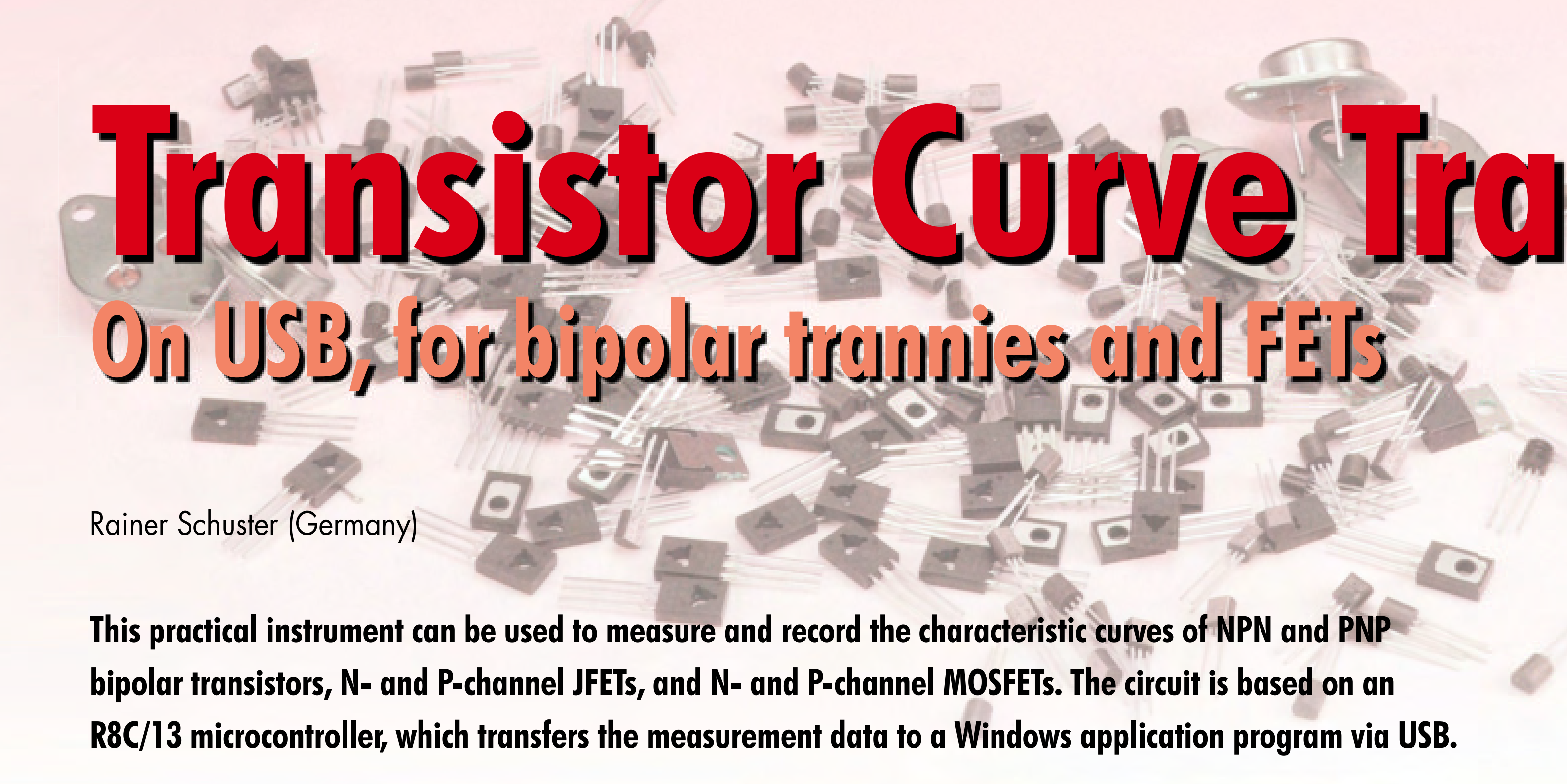


And finally…

It's 2009, and all the component manufacturers are offering free design tools… All of them? Well, not quite… some manufacturers are still holding out against this trend. On the Infineon, Hitachi, Freescale, Toshiba, and NEC websites, for example, we found nothing at all that was worth mentioning here.

Despite the number of design tools available on the Web, we may well be asking ourselves just how useful they are. Sure, some of them are well done, some of them are good to look at — but is there really any need for so many switch-mode PSU simulators?

(080874-I)



Transistor Curve Tra

On USB, for bipolar trannies and FETs

Rainer Schuster (Germany)

This practical instrument can be used to measure and record the characteristic curves of NPN and PNP bipolar transistors, N- and P-channel JFETs, and N- and P-channel MOSFETs. The circuit is based on an R8C/13 microcontroller, which transfers the measurement data to a Windows application program via USB.

Quick Project Specs

- R8C/13 microcontroller control module
- PC link via USB port
- User-friendly PC program with Help function (runs under Windows 2000 and XP)
- Curve data can be exported to and imported from Excel
- Checking transistor matching
- Records characteristic curves of NPN and PNP transistors, N-channel and P-channel MOS-FETs, and N-channel and P-channel JFETs
- Plotted curves can be printed out or embedded in other applications

Curve tracers have a long history in *Elektor*. The first article on this subject appeared in the 1979 Summer Circuits issue, followed by an article in June 1980, one in October 1988, and another one in December 1989. However, none of these circuits had a computer interface, and except for the circuit published in the December 1989 issue, they were limited to NPN transistors.
This situation changed with the project published in the May 1990 issue, which provided a control interface for connection to the Centronics port of an Atari computer. A version for use with a PC followed in September 1993. The circuit was upgraded in April 1998, including control software running under Windows 95.

Now, more than 10 years on it's time for a new curve tracer with features that definitely outclass all the previous versions.

Curve tracer operating principle

The operating principle of curve tracers has not changed much since the first article appeared in *Elektor*. The basic idea is to generate a time-dependent collector current by combining a ramp-shaped collector–emitter voltage with a base current that is increased either in discrete steps or continuously. The collector voltage is plotted on the X axis, and the collector current is plotted on the Y axis. The block diagram in **Figure 1** is based on this operating principle.

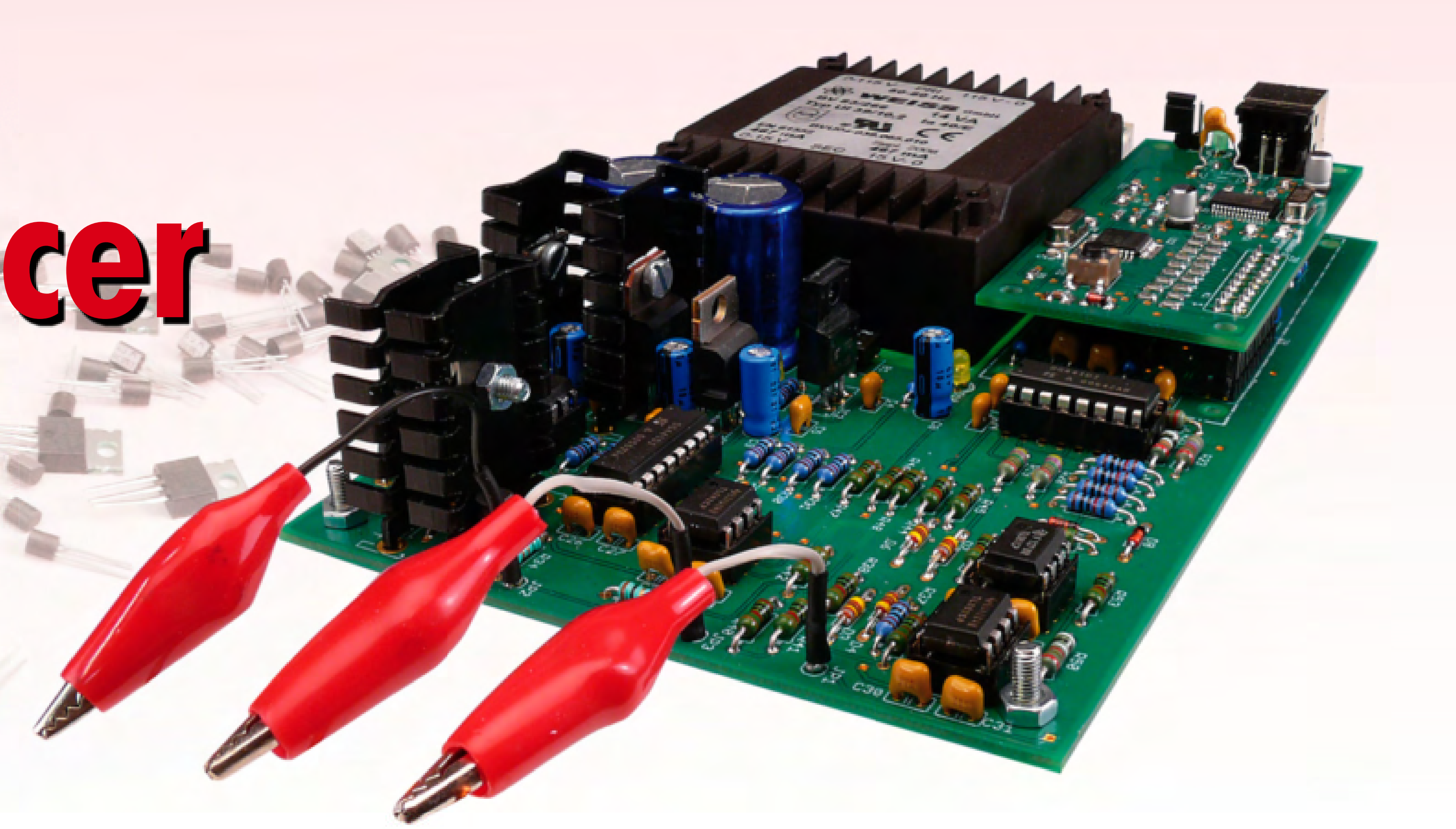
Control module

The curve tracer described here is built around an R8C/13 microcontroller, which is linked to a USB to serial converter (type PL2303). The complete circuit diagram is shown in **Figures 2a** and **2b**. It does not require much comment, since the circuitry of the R8C/13 section was described earlier in the February 2006 issue [1] and the USB controller is wired as described in the March 2006 issue [2]. All of this is fitted on a PCB measuring only 80 × 35 mm. Power is supplied via the USB port. Various I/O port pins, +V and ground are fed out to a 20-way socket header so the microcontroller module can also be used for other tasks.
The microcontroller can be reset whenever desired by means of Reset push-button S1. A set of eighteen 470-Ω resistors limit the output currents of the I/O pins to approximately 10 mA in order to protect the board against the potentially fatal consequences of a connection error.
When jumper JP1 is fitted, program code can be downloaded into the microcontroller via the USB port. One way to do this is to use the Renesas Flash Development Toolkit, which can be downloaded free of charge from the Web address for this project [3]. The previously mentioned Elektor articles and the R8C topic on the Elektor forum [4] provide adequate information about generating the software, downloading the hex file to the microcontroller, and installing the USB driver on the PC.

Circuit description

Generating the base current or gate–source voltage
Timer Y is operated in PWM mode to drive the base or gate of the transistor under test (TUT). See the description of the firmware for more details. The pulse rate is 2 kHz, and the pulse

cer



width can be set over a range of 0 to 100%. A Butterworth low-pass filter built around IC5a converts this signal into a variable DC voltage, which is amplified by a factor of 2 in IC5b to provide a voltage at the output of IC5b with an adjustment range of 0 to +10 V. The linearity of the output voltage can be seen clearly in the oscilloscope images (**Figure 3**).

The output voltage is inverted by IC8a to generate the base current for PNP transistors or the gate voltage for P-channel MOSFETs and N-channel JFETs. Switches S1 and S2 (IN1 and IN2) of analogue switch IC7 connect either a positive voltage or a negative voltage to the base or the gate of the TUT (see the firmware description for the associated truth table).

Switches S3 and S4 (IN3 and IN4) of IC7 are used to select the desired current range (0–10 µA, 0–100 µA, or 0–1 mA). The 0–1 mA range is always used with the 'FET' setting to set the gate voltage.

Generating the collector-emitter or drain–source voltage

Timer Z of the R8C/13 is also operated in PWM mode with a fixed pulse rate of 2 kHz. A Butterworth low-pass filter built around IC6 converts the pulse width into an analogue voltage and amplifies the voltage by a factor of 2 to again provide an output voltage with a range of 0 to +10 V. The voltage is inverted by IC8b to generate the collector–emitter voltage for PNP transistors or the drain–source voltage for P-channel MOSFETs and P-channel JFETs. Switching between positive and negative voltages is provided by analogue switch IC9. The current gain is provided by IC10 in combination with a transistor output stage formed by T1–T4. Resistors R34 and R35 limit the output current to approximately ±0.5 A.

Measuring the characteristics

The configured base current is not measured, but instead calculated by the software from the configured base–emitter voltage and the selected base current range.

In order to measure the configured collector–emitter voltage, it is first rectified by IC11 because it can be either positive or negative. After being divided by a factor of 2 by R42/R43, the rectified voltage is fed to the microcontroller A/D converter input AD0. The

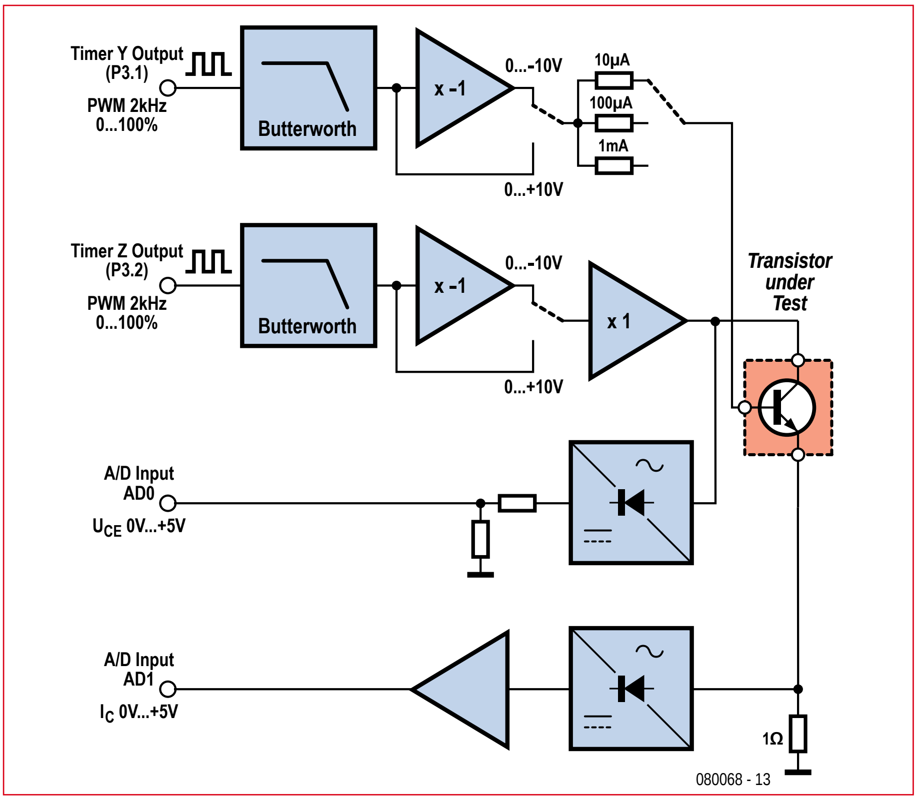


Figure 1. Curve tracer block diagram.

measurement range of 0 to ±10 V for the collector–emitter voltage thus corresponds to a range of 0–5 V at the A/D converter input. Diode D5 prevents the A/D input voltage from rising above 5 V in the event of a fault and protects the input from negative voltages.

The collector or drain current flows through R49 and generates a positive voltage at the output of IC12b. The output voltage ratio is 1 mV/mA. IC13 amplifies the voltage by a factor of 10 to yield a ratio of 10 mV/mA at the A/D converter input (AD1). The voltage and the A/D input is 4 V at the maximum collector or drain current of 400 mA. Here again, diode D8 prevents the A/D input voltage from rising above 5 V in the event of a fault and protects the input from negative voltages.

Table 1: Pin assignments of K1.

Pin	Signal	Pin	Signal
1	P1.7	11	P3.0
2	GND	12	P3.1
3	P1.3	13	P0.7
4	P1.6	14	P0.6
5	P1.1	15	P0.4
6	P1.2	16	P0.5
7	P4.5	17	P0.2
8	P1.0	18	P0.3
9	P3.2	19	+5V
10	P3.3	20	P0.1

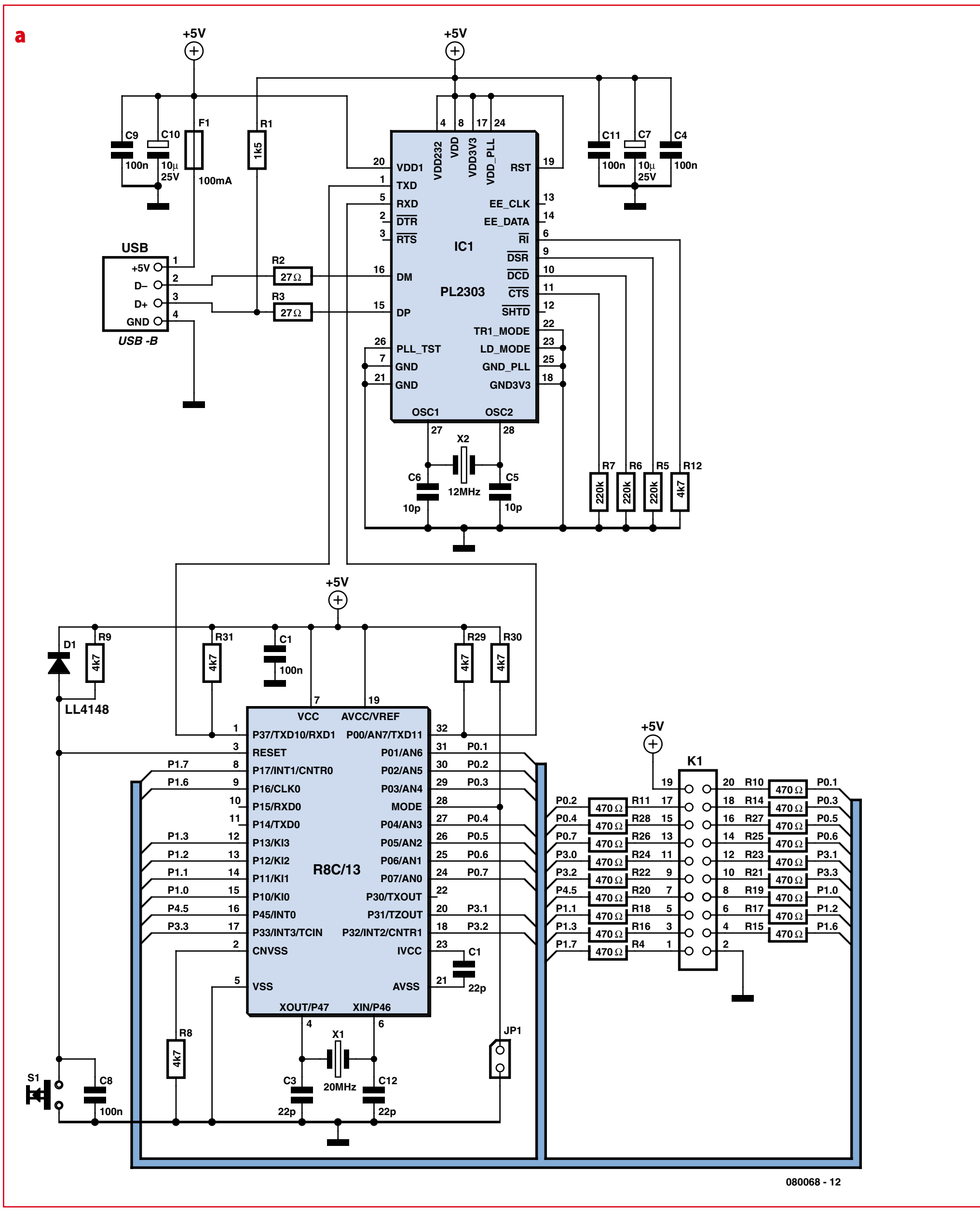


Figure 2. Schematic diagram of the microcontroller module (a) and control circuitry (b).

Power supply

The curve tracer is powered by an on-board PCB-mount transformer with two secondary windings, each rated at 15 V / 0.5 A. The rectified DC voltages of approximately ±20 V are fed to IC2 and IC3, which produce regulated voltages of ±15 V to power all of the analogue circuitry with the exception of the output stages. LEDs D10 and D11 indicate the presence of the supply voltages.

The ±12 V supply voltages for the output stages are provided by IC1 and IC4, which require heat sinks. Here LEDs D9 and D12 indicate the presence of the supply voltages.

R8C firmware

The microcontroller firmware was written in C using the Renesas High-performance Embedded Workshop. The details of the development environment fall outside the scope of this article. It has been described in the R8C articles in previous issues of *Elektor* [1][2][4]. HEW can also be downloaded from the project Web address [3].

Program flow

Initialisation

Timers Y and Z are first initialised as PWM timers with a pulse rate of 2 kHz. Timer X is used to execute wait loops. The UART1 interface is initialised to 9600 baud, 8 data bits, 1 stop bit, and no parity. The interrupt for this interface is enabled with a priority of 6. The following default values are then set:

- Maximum collector or drain current: 100 mA
- Transistor type: NPN
- Base current range: 10 μA
- Number of curves: 5

Configuring the base current range

The base current range is set using I/O pins P1.2 and P1.3 as follows:

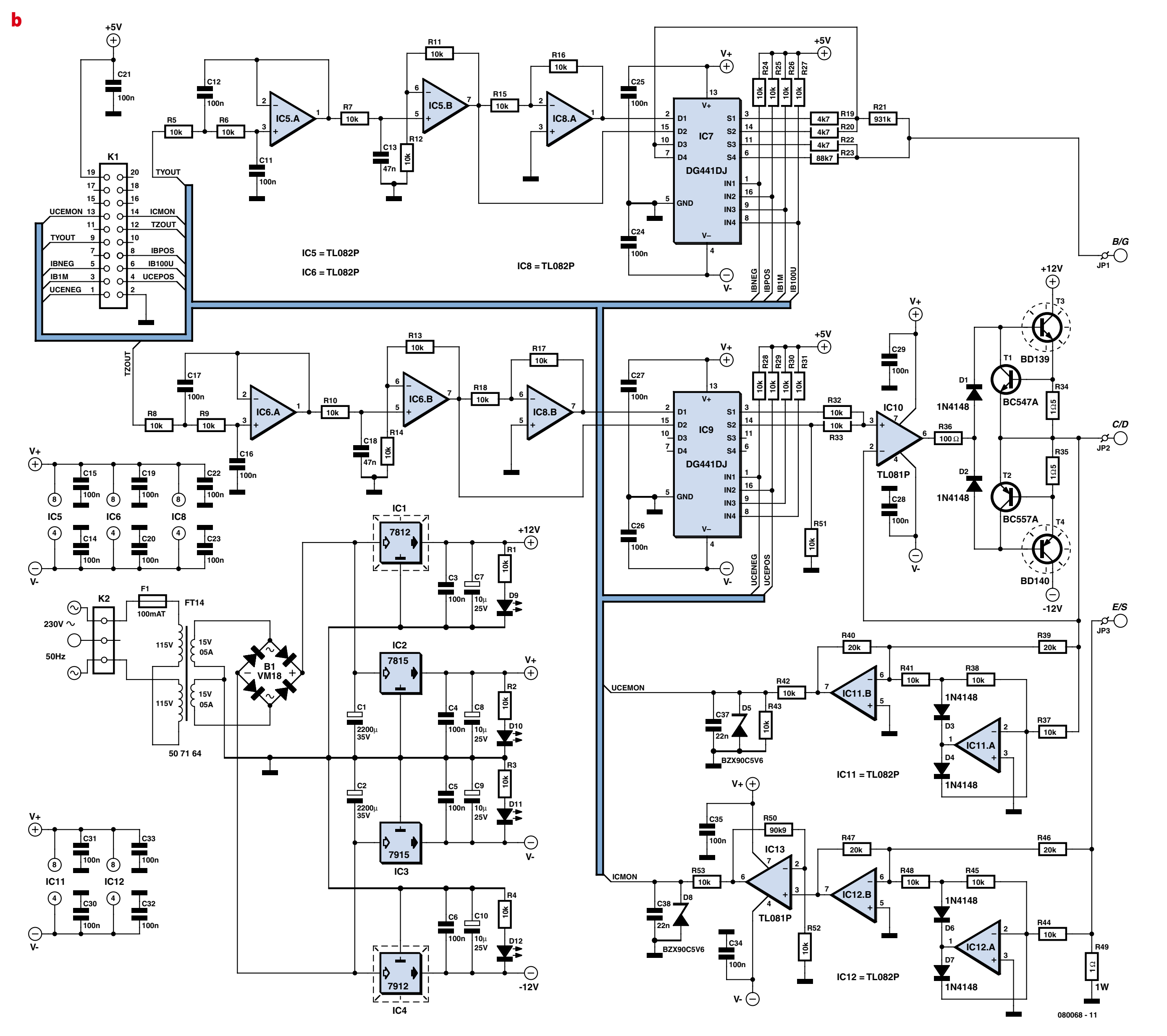
P1.2	P1.3	Base current range
1	1	10 μA
0	1	100 μA
1	0	1 mA

Configuring the transistor type

The polarity of the base current or gate voltage and the collector–emitter or drain–source voltage are set using I/O pins P1.0, P1.1, P1.6 and P1.7 (see **Table 2**).

Main program loop

The main program loop constantly polls the UART1 serial interface to see



whether any characters have been received. This interface is connected to the PC via the USB to serial adapter (PL2303). The protocol described below has been implemented for data transmission.

Each command consists of a command code (a letter in the range A to Z) optionally followed by one or more parameters, which are always written as ASCII character strings separated by commas. The defined command string terminators are CR (0x13) and LF (0x10). This protocol applies to transmissions in both directions (from the PC to the curve tracer module and from the curve tracer module to the PC), although the curve tracer module does not send commands to the PC on its own initiative. It can only send replies in response to requests from the PC. The commands are summarised in **Table 3**.

Table 2: Transistor type setting.

P1.0	P1.1	P1.6	P1.7	Base or gate polarity	Collector or drain polarity	Transistor type
0	1	0	1	+	+	NPN transistor or N-channel MOSFET
0	1	1	0	+	–	P-channel JFET
1	0	0	1	–	+	N-channel JFET
1	0	1	0	–	–	PNP transistor or P-channel MOSFET

Initiation of curve measurement

The curve measurement process starts with the configuration of the output parameters.

For bipolar transistors, the base current range is set to 10 μA and the base–emitter voltage (U_{BE}) is set to 0. For MOSFETs, the base current range is set to 1 mA and the gate–source

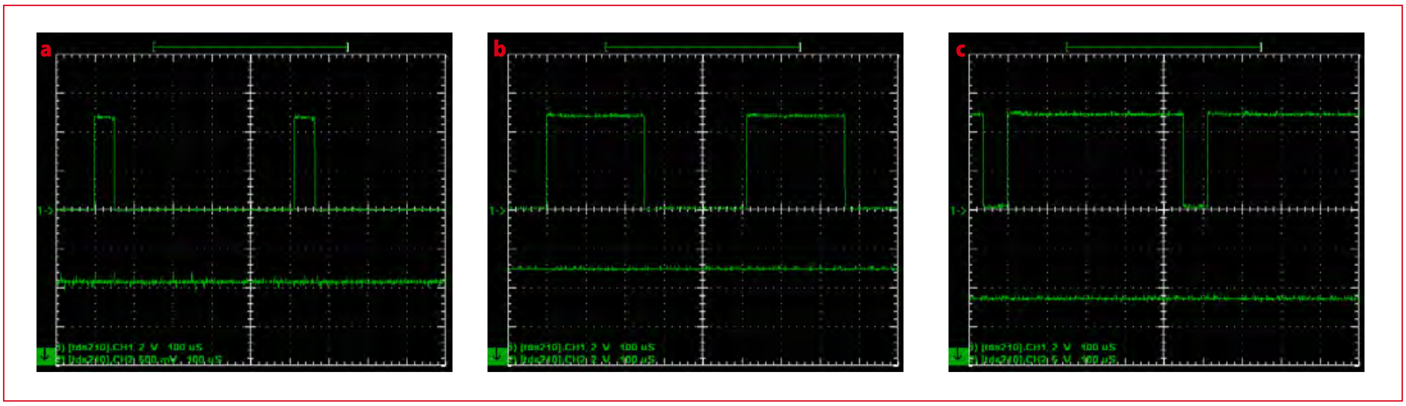


Figure 3. Pulse width and IC5b output voltage at pulse width ratios of 10% (a), 50% (b), and 90% (c).

voltage (U_{GS}) is set to 0. For JFETs, U_{GS} is set to the maximum level, which is +10 V for P-channel JFETs or –10 V for N-channel JFETs. The collector-emitter voltage (U_{CE}) or drain–source voltage (U_{DS}) is set to the maximum value (+10 V or –10 V), and curve tracing is started.

The base current (I_B) or gate-source voltage (U_{GS}) is increased until the maximum collector or drain current is reached. With bipolar transistors, the base current range is also incremented if the maximum collector current has not been reached. With JFETs, U_{GS} is reduced until the maximum drain current is reached.

If the maximum collector or drain current is not reached with the maximum value of I_B or U_{GS}, curve tracing is terminated and the error message 'M1<CR><LF>' is sent to the PC. The message 'M2<CR><LF>' is sent if the maximum collector or drain current is reached with a very low value of I_B or U_{GS}, which can occur with a shorted transistor. This message is also sent if too many curves are set with the "K" command. After the base current or gate–source voltage corresponding to the maximum collector or drain current has been calculated, the delta (increment value) is calculated from the base current and the number of curves to be measured (delta = $I_{B(max)}$ divided by number of curves). The previously mentioned error message is also sent if the increment is zero.

The actual measuring process begins if no error has been detected. For each value of I_B or U_{GS}, the collector-emitter or drain–source voltage is swept from 0 to 10 V while the associated collector or drain current is measured. The results are sent to the PC in the following format:

P<curve_no.>,<IB>,<Range><CR><LF>
N<UCE>,<IC><CR><LF>

curve_no.: the number of the curve presently being measured;
IB: xbase current or gate–source voltage (0–10 V);
Range: base current range; 0 = 10 µA, 1 = 100 µA, 2 = 1 mA;

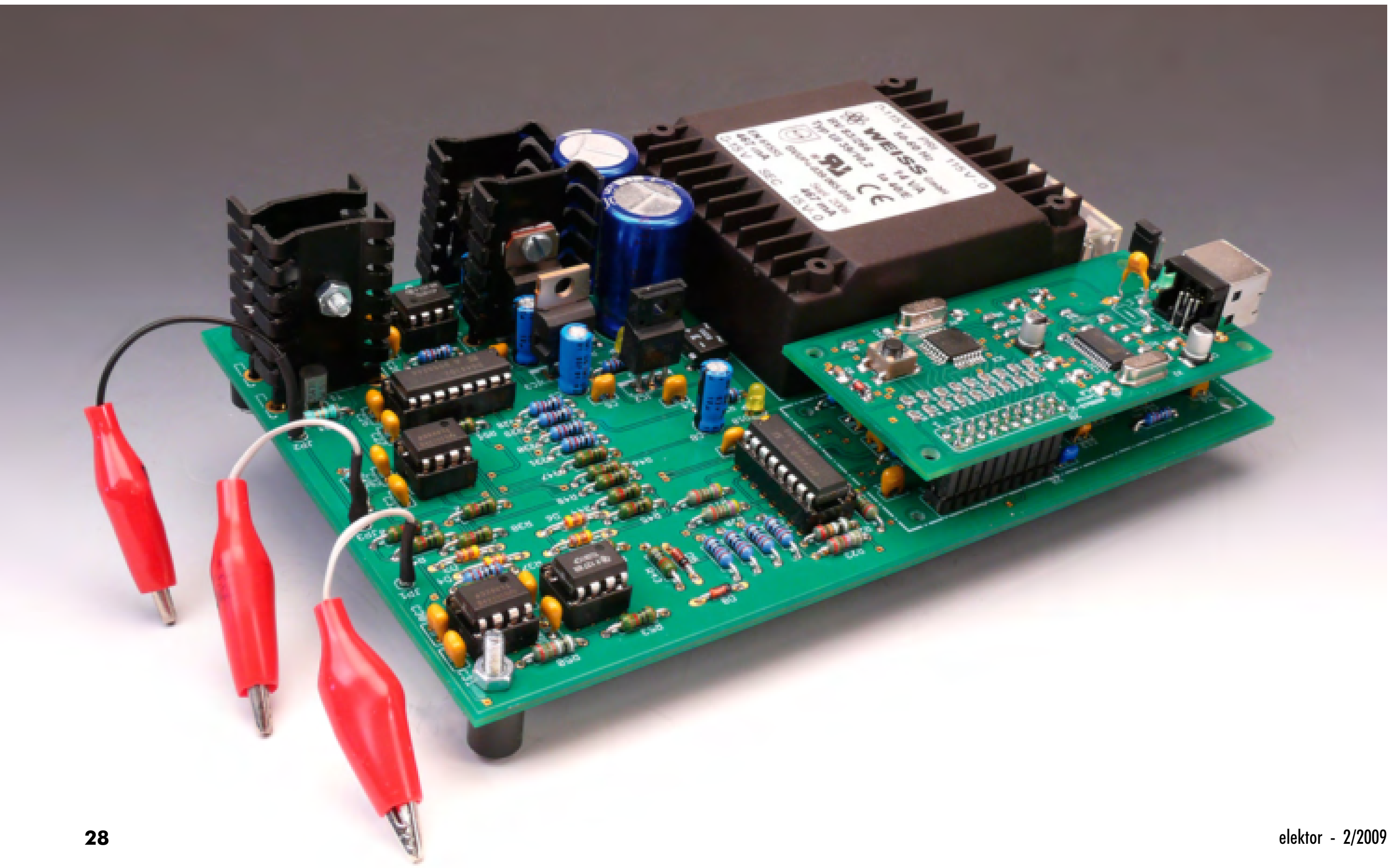


Table 3: Commands.

Command letter	Parameter	Meaning	Response from curve tracer module
A	-	Query I_D of curve tracer module	Note 1)
B	0 - 255	Pulse width for I_B or U_{GS}; range 0–10 V	–
C	0 - 255	Pulse width for U_{CE} or U_{DS}; range 0–10 V	–
D	0 = 10 µA 1 = 100 µA 2 = 1 mA	Base current range	–
E	0 = Positive 1 = Negative	Polarity of I_B or U_{GS}	–
F	0 = Positive 1 = Negative	Polarity of U_{CE} or U_{DS}	–
G	–	Query current value of U_{CE} or U_{DS}	G0 - 1024 Voltage [V] = parameter value divided by 100
H	–	Query current value of I_C or I_D	H0 - 1024 Current [mA] = parameter value divided by 2
I	0 - 1024	Maximum collector or drain current Current [mA] = parameter value divided by 2	–
J	1 = NPN 2 = PNP 3 = N-MOSFET 4 = P-MOSFET 5 = N-JFET 6 = P-JFET	Transistor type setting	–
K	0 -10	Number of curves (see text)	–
L	–	Start curve measurement	See text

1) The curve tracer module responds by sending its ID code: 'A + Curve_Tracer VX.X + <CR><LF>' (X.X = current version number). After this, the curve tracer module sends the current settings of the following configuration parameters:
- Maximum collector current (default = 100 mA)
- Transistor type (default = NPN)
- Number of curves (default = 5)

UCE: collector–emitter voltage; range 0–1024 (0–10.24 V);
IC: collector or drain current; range 0–1024 (0–512 mA).

The base current is calculated as follows:

Range = 10 µA: base current = IB × 0.025 [µA];
Range = 100 µA: base current = (IB – 14) ÷ 2.3 [µA] if IB ≥ 14; otherwise base current = 0;
Range = 1 mA: base current = (IB – 16) ÷ 2.6 × 10 [µA] if IB ≥ 16; otherwise base current = 0.

After all the values have been transferred, the curve tracer module sends 'O<CR><LF>' to the PC, which can then process the data and display the curves.

Downloading the firmware

The Renesas High-performance Embedded Workshop generates a Motorola (!) hex file (Curve_Tracer.mot), which can be downloaded via the USB interface with the aid of the Flash Development Toolkit [3].
To enable downloading, fit jumper JP1 on the microcontroller board and press the Reset button. After the firmware has been downloaded, remove the jumper and press the Reset button again.

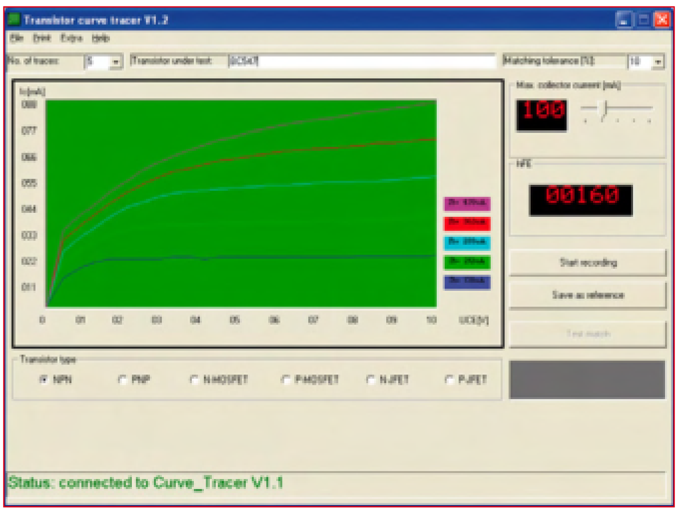


Figure 4. Example characteristic curve chart.

Application program

The application program provides the user interface for the curve tracer module and is written in Visual Basic 6.0. All program components were packaged using the VB6 Packaging Wizard. They can be installed in any desired folder by running the Setup.exe program. The application program can run under Windows 2000, XP, and Vista. All DLL files and ActiveX components are registered during installation, except for the Prolific driver for the USB to serial converter. This driver must be installed by running PL-2303 Driver Installer.exe. The Curve_Tracer.exe program can be started after the curve tracer module is connected to a USB port of the PC.

Functions

The window shown in **Figure 4** appears after the program starts up and establishes a connection to the

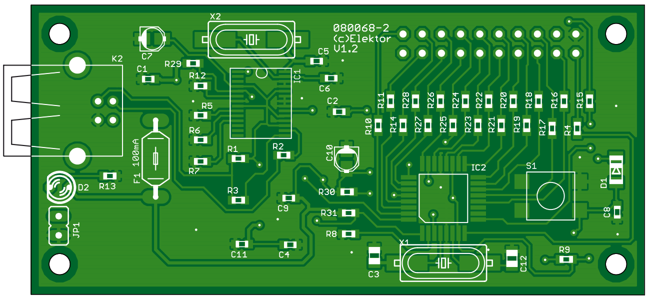


Figure 5. Layout of the microcontroller PCB.

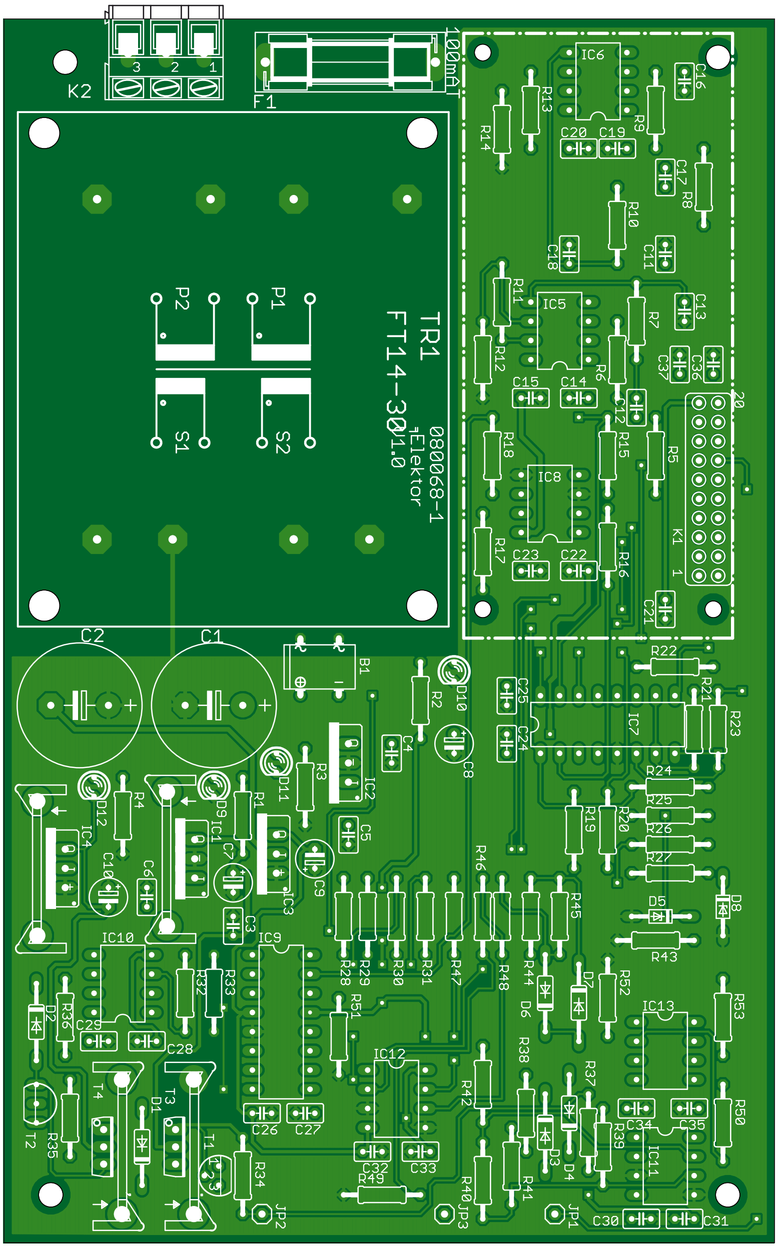


Figure 6. Layout of the main PCB.

COMPONENT LIST
Controller board # 080068-2

Resistors
R1 = 1kΩ5, SMD 0603
R2,R3 = 27Ω, SMD 0603
R4,R10,R11,R14-R28 = 470Ω, SMD 0603
R5,R6,R7 = 220kΩ, SMD 0603
R8,R9,R12,R29,R30,R31 = 4kΩ7, SMD 0603
R13 = 1kΩ, SMD 0603

Capacitors
C1,C2,C4,C8,C9,C11,C13 = 100nF, SMD 0603
C3,C12 = 22pF, SMD 0603
C5,C6 = 10pF, SMD 0603
C7,C10 =10µF 25V, SMD case A

Semiconductors
D1 = LL4148
D2 = LED, 3mm, low-current
IC1 = PL2303X

curve tracer module. The following functions are available in this window:

- Recording the characteristic curves of NPN and PNP transistors, N-channel and P-channel MOSFETs, and N-channel and P-channel JFETs.
- Exporting curve data to Excel and importing curve data from Excel.
- Checking transistor matching.
- Printing curve charts or embedding them in other applications.

The program has an extensive Help function, so the operation of the program is not described in detail here.

Construction

As can be seen from the schematic diagrams, the hardware is divided between two PCBs. The small microcontroller board (**Figure 5**) holds the R8C/13 and the PL2303, along with the associated components (**Figure 2a**). The main board (**Figure 6**) contains the rest of the hardware (**Figure 2b**), including the transformer.

All of the components on the microcontroller board are SMDs, so a reasonable amount of soldering experience is necessary for assembly. The main board is more roomy, and all of the components have leads or pins. The two 12-V voltage regulators, as well as power transistors T3 and T4, are fitted with heat sinks. Here you must take care that the heat sinks of the two transistors do not

IC2 = R5F21134FP (R8C/13)

Miscellaneous
K2 = USB-B socket
K1 = 20-way (2x10) DIL pinheader
S1 = pushbutton, 6mm, SMD
X1 = 20 MHz quartz crystal, SMD
X2 = 12 MHz quartz crystal, SMD
JP1 = 2-way SIL pinheader with jumper
F1 = 100 mA Polyfuse

COMPONENT LIST
Main board # 080068-1

Resistors
R1-R18,R24-R33,R37,R38,R41-R45,R48,R51,R52,R53 = 10kΩ
R19,R20,R22 = 4kΩ7
R21 = 931kΩ
R23 = 88kΩ7
R34,R35 = 1Ω5
R36 = 100Ω
R39,R40,R46,R47 = 20kΩ
R49 = 1Ω 1W
R50 = 90kΩ9

Capacitors
C1,C2 = 2200µF 35V radial
C3-C6,C11,C12,C14-C17,C19-C35 = 100nF, lead pitch 2.5mm
C7-C10 = 10µF 25V radial
C13,C18 = 47nF, lead pitch 2.5mm
C36,C37 = 22nF, lead pitch 2.5mm

Semiconductors
D1-D4,D6,D7 = 1N4148
D5,D8 = 5.6 V 500 mW zener diode
D9-D12 = LED, low-current, 3 mm
B1 = VM18, bridge rectifier 1 A / 100 V, DIP4
T1,T2 = BC547A
T3 = BD139
T4 = BD140
IC1 = 7812
IC2 = 7815
IC3 = 7915
IC4 = 7912
IC5,IC6,IC8,IC11,IC12 = TL082P
IC7,IC9 = DG441DJ
IC10, IC13 = TL081P

Miscellaneous
K1 = 20-way DIL socket
K2 = 3-way PCB terminal block, lead pitch 2.5mm
F1 = 100 mAT (slow) glass fuse, with holder
TR1 = mains transformer, secondary 2x 15V / 14VA
4 heatsinks type FK218/SA32 (21 K/W)

Microcontroller and PC Software: free download **080068-11.zip** at [3].
PCBs **080068-1** and **080068-2**, available from the Elektor Shop.
Controller board also with components fitted, Elektor Shop # **080068-91**.

touch. It's a good idea to use a piece of insulating material here.

The microcontroller board plugs into a 20-pin DIL socket on the main board. The assembled board set can be fitted in a suitable enclosure, such as the Bopla type E450 FVL. Various types of transistor sockets can be fitted to the enclosure to provide connections for the transistors, although a set of three short leads terminated in alligator clips is also a workable solution.

(080068-1)

Internet Links

[1] www.elektor.com/050179-2

[2] www.elektor.com/050179-3

[3] www.elektor.com/080068

[4] www.elektor.com/forum, go to R8C topic.

Microcontrollers fo

… Arduino for the enlightened

Clemens Valens (Elektor France Editorial)

Apparently Arduino is an Italian name – but when you search on the Internet, you mainly find dozens of references relating to electronics and programming. What's more, these references are often in relation to art. Electronics and art – now there's an interesting subject that's worth delving into! So just what exactly is Arduino?

At first sight, Arduino [1] is a small microcontroller board with a USB port (**Figure 1**) that comes in several models. There are even 'daisy'-shaped boards (Lilypads) intended for wearable applications, i.e. to be incorporated into garments. The Arduino board is programmed in a language very similar to C using Open Source tools available for Windows, Mac, and Linux. The hardware is also Open and anyone can make their own Arduino – the circuit diagrams and PCB photo masks are available free over the Internet. Arduinos are used a great deal by artists who need electronics in their creations.

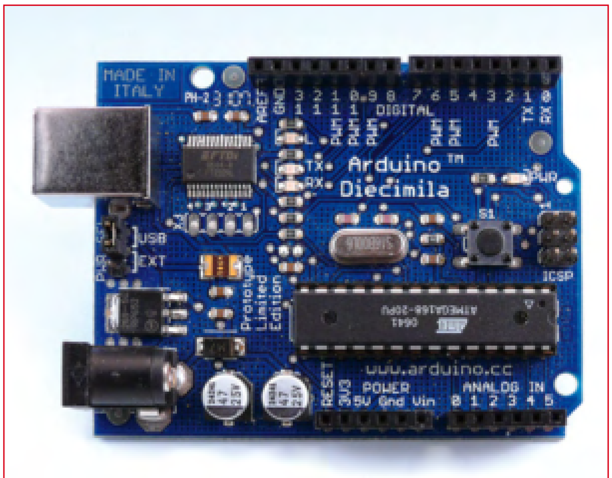


Figure 1. A Diecimila Arduino board. The new Duemilanove board is almost identical. These boards are cheap and easy to find.

When you look at it a bit more closely, an Arduino is not exactly a microcontroller board. In fact, an Arduino is quite simply an 8-bit microcontroller from Atmel – an ATmega8 for the earliest Arduinos and now more likely to be an ATmega168. This microcontroller is loaded with a 'bootloader' program that lets you load an application into the controller via a serial port, without overwriting the bootloader. Since modern computers no longer have serial ports, a USB port is often used. All this becomes an Arduino when you decide to dedicate certain pins of the controller to certain functions, since this allows the Arduino development environment to be used for writing and compiling application before loading them onto the controller.

The applications, called 'sketches', are written in a language that closely resembles C. Hardly surprising – it *is* C, but with some additional functions. All of the functions presented as the language for Arduino form a Hardware Abstraction Layer that lets you program the controller without needing to delve into the innards of the processor. The language has everything you need for most applications. Broadly speaking, there are functions for digital and analogue inputs/outputs, a few basic mathematical functions, time management functions (delays) and a few function for serial port communication – asynchronous (UART) and synchronous (SPI).

The digital I/O functions let you manipulate the logic levels of the pins, to read and write them. There is also a special function that makes it possible to measure the duration of a pulse. Using the analogue I/O functions, it is possible to read voltages and generate PWM signals. Lots of applications don't require anything more than this, and this is exactly where Arduino's strength lies. There's no need to go ferreting around in the registers and the controller data sheet to make a PWM output or a counter work — the 'dirty work' has already been done.

If these functions aren't enough, it's perfectly possible to program on a lower level and, just as in standard C, you can also add libraries with their own functions. But do watch out – if you go off into the darker depths of the Arduino programming language, you risk losing compatibility with the rest of Arduino community.

The Arduino community? Already, Arduino is a microcontroller, as well as a development environment and a programming language – now it's a community too? Yes! In fact, Arduino is more of a philosophy, the aim of which is to popularize technology to make it accessible to artists. Arduino is a logical sequel to Processing [2] and Wiring [3] projects. Processing is a multimedia programming language and Wiring is a development environment for artistic electronics But now we're starting to get away from our original

r Dummies ...

point; refer to the box about the origins of Arduino if you want to find out more.

Elektorino

Simple, free programming is something we're interested in. What's more, the electronics involved seem to be simple – so what could be more logical than to produce our own Arduino-compatible system? Well, that's just what we're going to do!

Our starting point is the basic Arduino Serial board. The office computer I use all the time still has a serial port, but for the unlucky owners of a computer that doesn't, we're going to use the USB-TTL cable [4]. In any event, we're going to be needing a TTL interface of some kind, as our own Arduino will only have a TTL serial port.

Our processor is going to be an ATmega168, which we'll be running at 16 MHz, to avoid getting caught out. For even though the controller is perfectly capable of operating at up to 20 MHz, the standard bootloader assumes a speed of 16 MHz. This can of course be modified if you are prepared to go delving into the bootloader – but for the moment we just want an Arduino board that works.

To finish off our Arduino, all we need do is add an LED, a reset push-button, a few resistors and capacitors, and two connectors: one for the serial port and the other for programming the bootloader. We need the latter to be able to program our Arduino for the first time, when the controller is still blank. Later, when the application is finished and the bootloader is no longer needed, this connector can be used for programming the controller directly from the application, which saves memory.

The LED has several functions. Given that the LED is present on several types of Arduino, lots of sketches use it. So does the bootloader, which flashes it at start-up.

Here's the circuit diagram of our finished Arduino (**Figure 2**). Thanks to the simplicity of the circuit, we can build it on prototyping board.

Unfortunately, we are not allowed to call our fine project Arduino – only boards approved by the Arduino community have the right to that name. This is why a second movement Freeduino [5] was created, which allows free use of the name Freeduino for home-made Arduino boards. But after all, it's only a name, so let's call ours Elektorino.

Implementation

Before you can load a sketch into the Elektorino, you need to load the bootloader. This is where things get more complicated, as there are two official Arduino bootloaders, the only apparent difference between them being the way the sketch is run after loading. This is achieved by way of a controller reset, which the Arduino environment can handle if the board has been equipped for it, and if it has the right

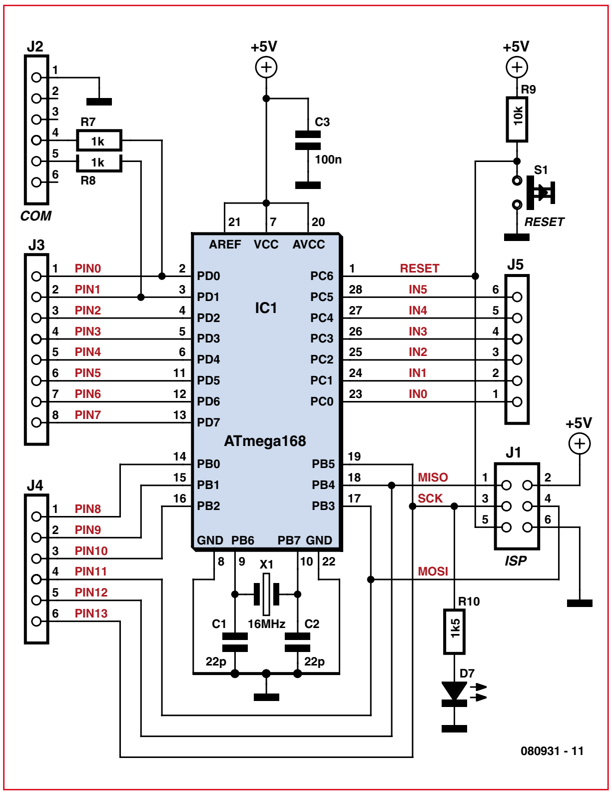


Figure 2.
The circuit of the Elektorino. Not at all complicated. The connectors for the pins are not necessary, they are mainly used as a reference.

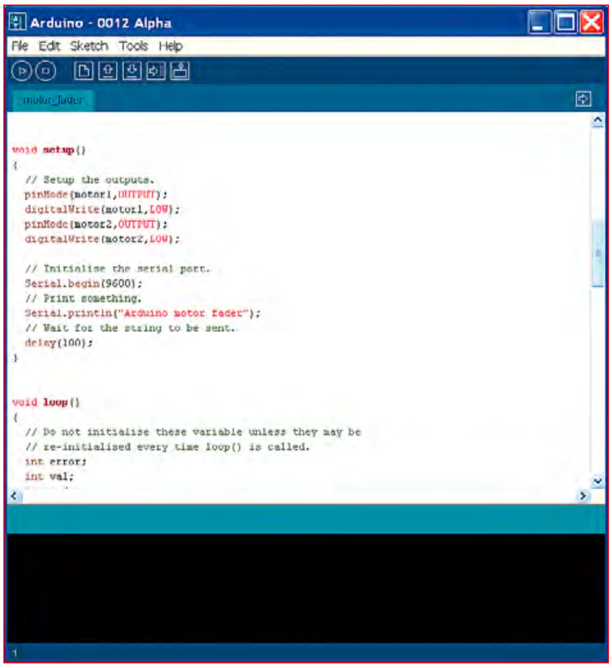


Figure 3. The Arduino development environment.

bootloader. We haven't made any provision for this, and so we need the basic bootloader for the so-called NG (New Generation) boards.

But there is also another option: a third bootloader called ADABOOT [6]. This bootloader, an improved version of the official bootloaders, handles the reset and run delays differently. Initially, I worked for a bit with the NG bootloader, before replacing it with ADABOOT. Both worked perfectly well, but in the end I adopted ADABOOT, because it flashes the LED while the sketch is loading and because it is more convenient.
See the box to find out how to load the bootloader into the controller.

Hello world!

Once you have succeeded in loading the bootloader, it's time to see if Elektorino manages to communicate with the Arduino environment and if it is possible to load a sketch. So let's install Arduino. I've only done it under Windows XP, and that was extremely easy. All I had to do was download a large zipped file and unzip it somewhere onto the hard drive.

After running the Arduino environment (arduino.exe), you find yourself with a window like the one shown in **Figure 3**. Go into the *Tools* menu, then *Board*, and select the Arduino board being used. Of course, our one isn't listed, but an NG board using an ATmega168 will do.
You also have to select the serial port to be used for programming the microcontroller. Go into the *Tools* menu, then *Serial port* and select the correct port. If you want to use a USB serial port, check first that the drivers are present.
The Arduino environment comes with a small sketch, *Blink*, for checking that the board is working, and we can use this, since we have fitted the LED. The procedure is simple:

- Load the sketch; it's in *File > Sketchbook > Examples > Digital > Blink*.
- Compile the sketch by clicking the *Verify/Compile* button; this only takes a few seconds and (usually) ends with a success message.

- Load the sketch into Elektorino; first press the Reset button briefly, then click *Upload* to set the program loading. If all is well and if you are using ADABOOT, after a short delay you'll see the LED start to flash randomly — this is normal, it shows the transfer is taking place. After around five seconds (depending on the size of the sketch), the program is loaded and the controller is rebooted (the exact way in which the program is run depends on the bootloader). If the LED now flashes at a frequency of 1 Hz, everything has gone alright. Elektorino is working! If nothing happens, try resetting Elektorino.

Loading the bootloader...

…is not as hard as all that, as long as you have all the information you need. To save you hours on the Net, we've summed it up for you here in a few lines.

First of all, you need a programmer. There are several possibilities, for example, the one published in the 2008 double issue [7], or another 'SK200 compatible' programmer, easy to build using the circuit available on the PonyProg website [8]. On the Arduino website [1] yet another parallel port programmer is mentioned which is very simple and can be used directly from the Arduino environment. I tried it out, and managed to scramble one controller with it… so I went back to an SK200 one I already had.

Next, choose your bootloader. I recommend ADABOOT [6], but the NG version available on the Arduino website works perfectly well too.

Loading the bootloader into the controller can be done, for example, using AVRDUDE [9], supplied with the Arduino environment. AVRDUDE is a typical UNIX tool – it's basically a FreeBSD tool – with lots of incomprehensible options. Because it's very easy to make a mistake, here are the commands that work well (copy the bootloader into the directory that contains the AVRDUDE executable):

```
avrdude -p m168 -c pony-stk200 -V -e -U lock:w:0x3F:m -U hfuse:w:0xDF:m -U lfuse:w:0xFF:m -U efuse:w:0x0:m
avrdude -p m168 -c pony-stk200 -V -D -U flash:w:ATmegaBOOT_168_ng.hex
avrdude -p m168 -c pony-stk200 -V -U lock:w:0x0F:m
```

If you use another programmer, replace pony-stk200 with the appropriate value. Also check the name of your bootloader.

There are three commands that, broadly speaking, unlock the memory, load the program, set the fuses, and finally lock the memory. Refer to the AVRDUDE instructions if you want to know exactly what is going on (sensitive souls are advised to refrain!). Locking the memory is used to avoid overwriting the bootloader accidentally when loading a sketch.

A good website about the AVR is called Lady Ada [10].

A real application

It's all very well to have an Arduino development environment that works wonderfully well, but without a real application, it's not very interesting. I already had ten motorized slider pots, and it was high time to put them to good use. Why not with Elektorino? Elektorino has analogue inputs and PWM outputs — everything we need to drive a motor. So I'm going to suggest a driver for motorized faders. Note that this circuit can be used with any ATmega168-based Arduino board.

The fader in question (**Figure 4**) consists of a slider pot, a small motor, and an assembly of a few rollers, springs, and a piece of cord that enables the motor to move the slider in both directions. This assembly allows the motor to freewheel when the slider is unable to move – at each end of its travel, for example. Apart from the *10K B* marked on the fader, I didn't have any technical data on it, but a few experiments showed that the motor turned at a suitable speed when powered from 12 V. In this case, its consumption was around 200 mA.

The *B* marked on the fader might lead us to think it's a log model (as is often the case), but after checking, my faders turned out to be linear ones.

As a motor driver, I chose a cleverly-modified double H bridge with just two control lines and three states: anti-clockwise, clockwise, and braking – just what we need (**Figure 5**). Usually, two controls allow four states, but in this circuit, states 00 and 11 are the same. A 5 V regulator has been slipped into the circuit so as to be able to power the whole controller assembly and motor from 12 V. The transistor are all NPN types, and those forming the bridge must be capable of carrying 200 mA happily. In my prototype, I used BD139s.

The potentiometer is wired as a simple potential divider. By measuring the voltage on the wiper, we know where it is (just so long as it's a linear pot).

The motor driver controls must be connected to digital outputs capable of supplying a PWM signal. An Arduino based around an ATmega168 has six, an ATmega8 only three. The pot wiper itself can be connected to any of the analogue inputs – in our case, in0.

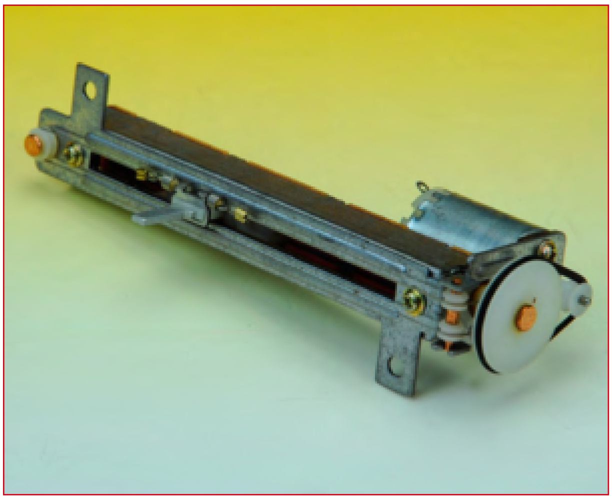


Figure 4.
A motorized fader of unknown make.

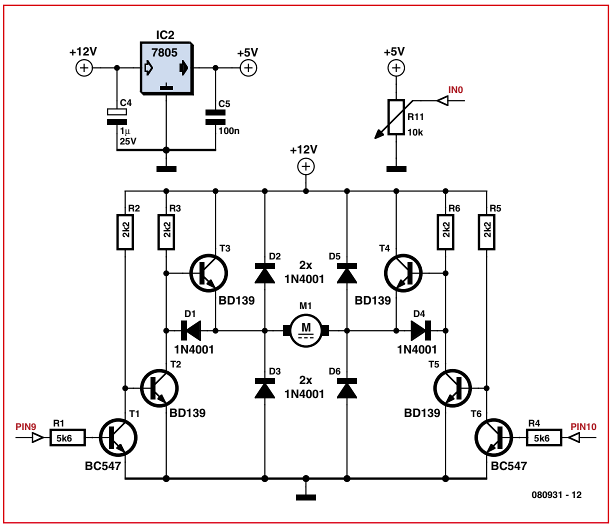


Figure 5.
The modified double H bridge and its three states. The labels refer to the pin designations, not the terminals on the controller.

The sketch

Now that we have connected a motor driver to Elektorino (**Figure 6**), it's time to deal with the software. You'll see, the final sketch will be amazingly simple, thanks to the power of the Arduino.

A basic sketch consists of two functions: *setup()* and *loop()*, which are called by the layer of a lower level. In *setup*, called once at runtime, we put everything that relates to initializing the system – for example, the inputs/outputs and the serial port speed.

99.9% of embedded programs probably spend their whole lives in a loop. This is why in Arduino this loop is already implemented in the form of the *loop* function. This *loop* function is called periodically and may be regarded as Arduino's *main*. It's important to realize that, even though it looks like a special function, *loop* is just like any other function in C. So its local variables are reset each time it is called and variables that are required to keep their values between different occasions *loop* is called must be declared globally (or as *static*, for those familiar with C.)

The *setup* in our sketch doesn't contain anything very much. The Arduino pins are inputs by default, so only the two outputs need to be initialized. We're going to be using the serial port to drive our circuit, and for this it needs to be initialized. Thanks to the simplification offered by Arduino, all we have to do is enter the communication speed – in our case, 9,600 baud.

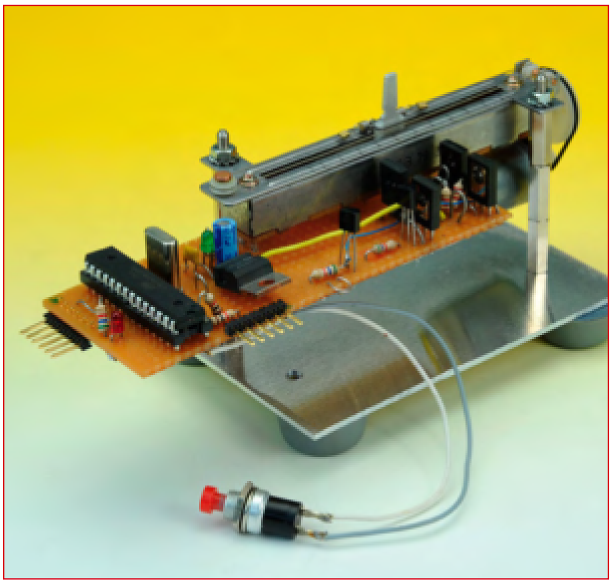


Figure 6.
The Elektorino prototype: the ATmega168 is on the left and the double H bridge to drive the motor on the right.

Processing, Wiring, and Arduino

Processing [2] is a language and an Open Source programming environment for programming images, animations, and interactions. The project, an initiative from Ben Fry and Casey Reas, is based on ideas developed by the Aesthetics and Computation Group of the MIT Media Lab. Processing was created in order to teach the fundamentals of programming in a visual context and to serve as a sketchbook or professional software production tool. Processing runs under GNU/Linux, Mac OS X, and Windows.

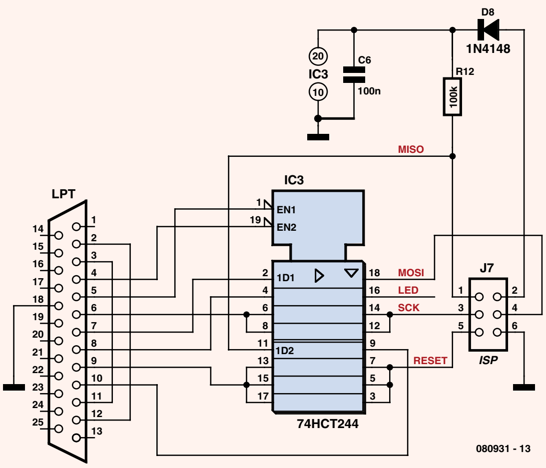
Several books have already been written on Processing.

Just like Arduino, **Wiring** [3] is a programming environment with microcontroller board for exploring electronic arts, teaching programming, and quick prototyping. Wiring, programmed in Processing, is an initiative by Hernando Barragán and was designed at the Interaction Design Institute Ivrea (IDII) in Italy.

Arduino [1] is a fast, Open Source electronic prototyping platform. Arduino is aimed at artists, stylists, enthusiasts, and anyone interested in creating objects or interactive environments. Created by Massimo Banzi, Gianluca Martino, David Cuartielles, and David Mellis, Arduino uses a programming language based on 'Processing'. Arduino may be regarded as a simplification of 'Wiring'.

Moving the pot wiper is done in *loop*. The principle is very simple: if the voltage measured on the input pin is different from the voltage required, the slider must be moved in the direction which will reduce this difference. In real life, it's a bit more complicated than that. To start with, there's the problem of direction, but more significant still is the problem of inertia. Once the slider is moving, it takes a little time for it to come to a complete halt. So it's easy to overshoot the required position if braking occurs too late. In the event of an overshoot, the slider has to be brought back, with the same risk of overshooting again, and so on. The system may even begin to oscillate.

The STK200 compatible programmer used by the author. There are also simpler programmers – it's a matter of personal preference.



To avoid these problems, we have used a Proportional Differential (*P-D*) regulator. In this type of regulator, the system reaches its final value without overshoot by continuously adjusting the correction signal according to the difference remaining to be corrected. So at the start of an adjustment, when the error is greatest, a strong correction signal is applied. Then, once the difference starts to reduce, the correction signal reduces too and the system slows down.

The correction signal consists of two parts: a signal proportional to the error (*P*) and a signal proportional to the error reduction (*D*). With a properly adjusted system of this sort, the slider can be moved quickly without overshooting the target value.

In the sketch (**Listing 1**) we can see the *P-D* regulator in the *loop* function. First, we measure the voltage at the wiper. The target value is subtracted from the measured value to obtain the error to be corrected. From this value, we calculate the two components *P* and *D* of the correction signal. The *P* component is the error multiplied by the constant K_p; the *D* component is obtained by multiplying the difference between the current value and the previous error by the constant K_d. The values for these constants were determined by experimentation, and you can modify them to see how the affect the adjustment. It's highly instructive.

The two components P and *D* are combined and the result is adapted to the range of usable values. The pot slider doesn't move for values below 50, and the maximum value for the PWM signal is 255.

Then we look to see if the error is small enough for us to be able to stop the motor. This comparison has to be performed for both slider directions. We leave a small margin for error, since perfection is perhaps a little over-ambitious.

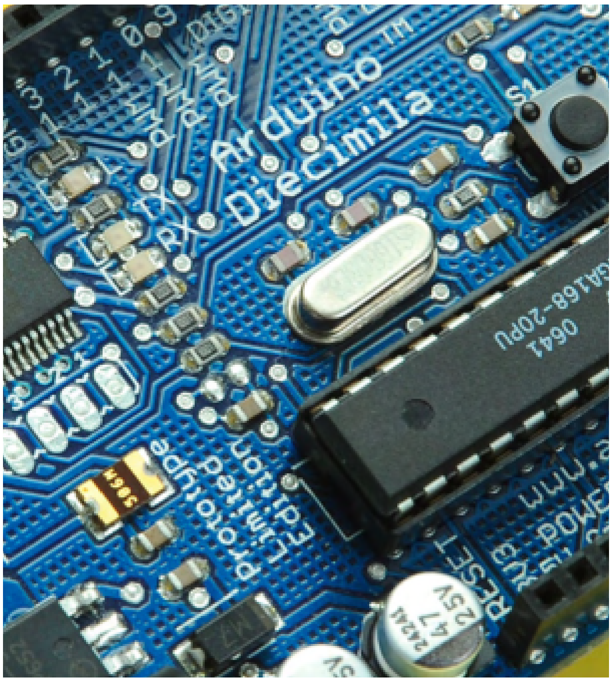
When the error is small enough, we prevent further corrections so as to free up the slider; we make the assumption that the system is never going to overshoot the target value.

AVR ISP via USB

If you use the FT232R chip as a USB interface and if you have access to all its pins, it is possible to use this chip to load the bootloader into the microcontroller without even needing a special ISP programmer! The FT232R chip has a bus called a CBUS with a bit function that lets you manipulate the associated pins individually. A certain Mr Suz from Japan has written a small piece of software that exploits this possibility and which can be downloaded free. Its avrdude-serjtag tool only works under Windows and its website is unfortunately in Japanese (suz-avr.sblo.jp/article/4438871.html). However, on his site one of his compatriots kindly explains in detail in English how to program an Arduino using this tool. See reference [11] for details.

In this way, it's possible to move the slider manually, without the system's trying to move it back into place. (Who's the strongest?)
Once the slider has been released, the system starts to output the slider position periodically (10 Hz) via the serial port. The serial port input is also scanned and as soon as four characters have been received, they are transferred as a target value for the slider and the PD regulator is re-activated to move it to its new position. No format checking is performed for the value received, the system requires an ASCII four-digit value between 0000 and 1023. To minimize errors, the target value obtained is limited between 3 and 1020, which minimizes problems of continuous activation at the ends of the travel.
The serial port is not used while the motor is operating, as this might produce interference, resulting in inaccurate positions or even oscillation. I've not taken the trouble to find out why: I'll leave that for you to do!

(080931-I)



References and Resources

[1] http://arduino.cc

[2] www.processing.org

[3] http://wiring.org.co

[4] www.elektor.fr/usb-ttl

[5] www.freeduino.org

[6] http://nearspacevermont.org/TheShoppe/freeduino/ADA-BOOT.shtml

[7] SimpleProg – ISP for AVR, Elektor, July/August 2008

[8] www.lancos.com/prog.html

[9] www.bsdhome.com/avrdude

[10] www.ladyada.net/learn/avr/index.html

[11] www.geocities.jp/arduino_diecimila/bootloader/index_en.html

Getting Started with Arduino, Banzi, Massimo, O'Reilly, 2008

Making Things Talk, Igoe, Tom, O'Reilly, 2007

The Duemilanove Arduino board is available from several sources including FunGizmos (US), LittleBird (Australia), SKPang (UK), Tinker (Italy), Make Magazine (Makershed.com).

Listing 1 – Easy-peasy!

```
void loop()
{
  int error;
  int val;
  int spd;
  float spd_p, spd_d;

  // read wiper voltage.
  val = analogRead(slider);

  // Calculate error.
  error = val - target;

  // Calculate proportional component P.
  // Two directions - so use absolute value.
  spd_p = abs(error)*Kp;

  // Calculate differential component D.
  spd_d = (last_error-error)*Kd;
  last_error = error;

  // Now mix P and D.
  spd = int(spd_p+spd_d);

  // Do not exceed limits.
  spd = constrain(spd,0,255);
  // Compensate friction.
  if (spd<50) spd += 50;

  if (error<-1 && stop==0)
  {
    // To maximum value ("left").
    digitalWrite(motor2,LOW);
    analogWrite(motor1,spd);
  }
  else if (error>1 && stop==0)
  {
    // To minimum value ("right").
    digitalWrite(motor1,LOW);
    analogWrite(motor2,spd);
  }
  else
  {
    // Shut down motor
    digitalWrite(motor1,LOW);
    digitalWrite(motor2,LOW);
    stop = 1;

    // Transmit cursor position.
    Serial.println(val);
    delay(100);

    // 4 characters form a new target value.
    if (Serial.available()>=4)
    {
      target = Serial.read()
   - '0'; // Thousand.
      target = Serial.read() - '0'
   + target*10; // Hundred.
      target = Serial.read() -
   '0' + target*10; // Ten.
      target = Serial.read() -
   '0' + target*10; // One.
      constrain(target,1,1022);
      // Start motor
      stop = 0;
    }
  }
}
```

From 17x4 to Chann

Marc Neujahr (Germany)

Direct reception of satellite broadcasts is still not a reality — we need to fit a dish and receiver before viewing can begin. However, with the help of a little hardware it is now possible to supply programmes to over 100 subscribers using just one dish. Another novel system allows eight receivers to view different programmes over a single coax cable!

It was back in the mid 1990's that domestic reception of satellite TV programmes first became economically viable. The satellite signal path is line-of-sight and highly directional so it does not suffer so much from the obstacles to propagation that sometimes afflict terrestrial TV broadcasts. The picture and sound quality were a revelation at the time but some of the limitations of this type of reception also became apparent: How could you record one programme while watching another? How could a second TV be added to the system? How could programmes be distributed to many users in apartments without producing an unsightly forest of dishes?

The technology is a little more mature now and many systems have since been developed which directly address these problems.

A typical domestic installation consists of a dish pointed at the satellite of interest with an LNB (see *Glossary* at the end of the article) fitted at the dish focus sending a frequency down-converted band of signals to a receiver (tuner). Adjacent channels are transmitted with alternate polarisation to help reduce interference. This set up effectively produces a horizontal and vertical receive band of frequencies, the basic LNB cannot receive both bands simultaneously so switching between vertically or horizontally polarised signals is achieved by changing the supply voltage from the receiver to the LNB from 14 to 18-V.

When the satellite bandwidth (10.7 to 11.7-GHz) was increased with the introduction of the high band (11.7 to 12.7-GHz) the number of satellite receive bands went up to four: (horizontal Low-Band, vertical Low-Band, horizontal High-Band, vertical High-Band). The high band is used almost exclusively for digital transmissions and is often referred to as the digital band but this is not strictly correct. The receiver superimposes a 22-kHz signal on the LNB coax to make it switch from low to high band reception. Using the basic LNB only one of the bands can be down-converted (to prevent unacceptable losses in the coax) and sent to the tuner at any one time so a single coax cable is required between each LNB and receiver. Band selection takes place inside the LNB or in more complex systems in a Multiswitch unit.

Switched LNBs and Multiswitch units

One solution to the problem of using two TVs in the system is to fit a second complete system so that both receiv-

el Routing

Satellite TV distribution

Illustration: courtesy GTN GmbH

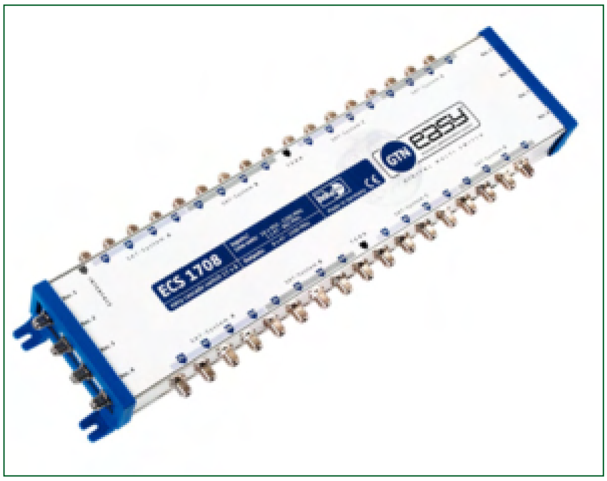


Figure 1. The ECS1708 from GTN GmbH Is a cascadeable Multiswitch capable of distributing 16 satellite bands (plus a terrestrial TV input) to eight subscribers.

ers work independently. More recent advances in design have produced LNBs with multiple outputs, these are supplied with single, twin (known as a 'twin output'), quad or octal outputs supplying signals for one, two four or eight set-top tuners. Each tuner can independently select its own band from the LNB by putting the appropriate control voltage on its coax. This system requires a single coax for each tuner so up to eight cables are run from the LNB to wall mounted satellite signal outlets. The term 'uni' when applied to an LNB indicates that it can receive all four bands transmitted by the satellite.

This type of installation gives several users independent control over programmes transmitted by one satellite but when more than eight subscribers want to select programmes from more than one satellite and also incorporate terrestrial/cable TV feeds it is necessary to install a Multiswitch unit. This is used with a special Quattro LNB which does not need to be switched between bands and simply outputs the four satellite bands simultaneously to the Multiswitch via four lengths of coax. The Quattro LNB is not to be confused with the quad LNB mentioned above. Conventional 'uni' LNBs can also be used here and in this case the Multiswitch fixes the receive band for each LNB by generating the necessary tone and/or voltage level. The Multiswitch unit now has access to the complete satellite signal spectrum so it can switch any selected channel through to any of the 16 user outputs. Sometimes when a Multiswitch installation cannot receive all programmes the problem can be traced back to a fault in the control signals to the LNBs.

Looking for something bigger?

Conventional Multiswitch installations are suitable for up to 16 subscribers and some of the larger units provide inputs for up to 16 bands which allow four Quattro LNBs to be fitted to the same dish, each focussed on different satellites. This type of installation provides a large number of transponders from a single dish and should help to reduce the urban 'dish clutter' often seen in inner cities.

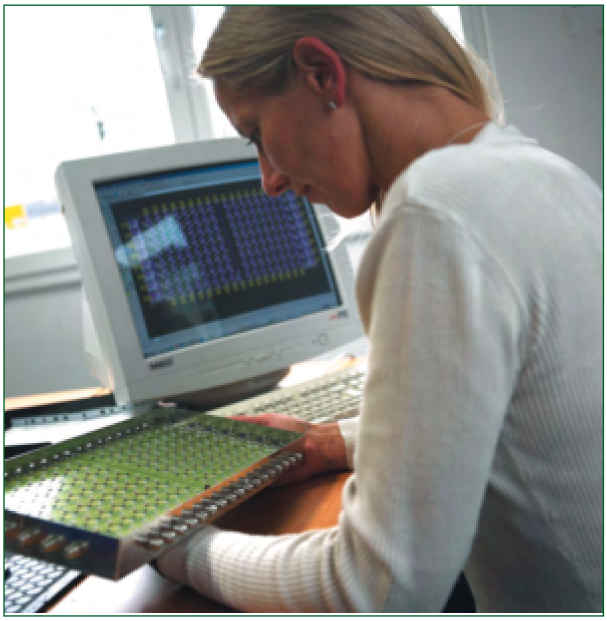


Figure 2. Visual inspection of an SMD populated Multiswitch PCB. (photo: GTN GmbH).

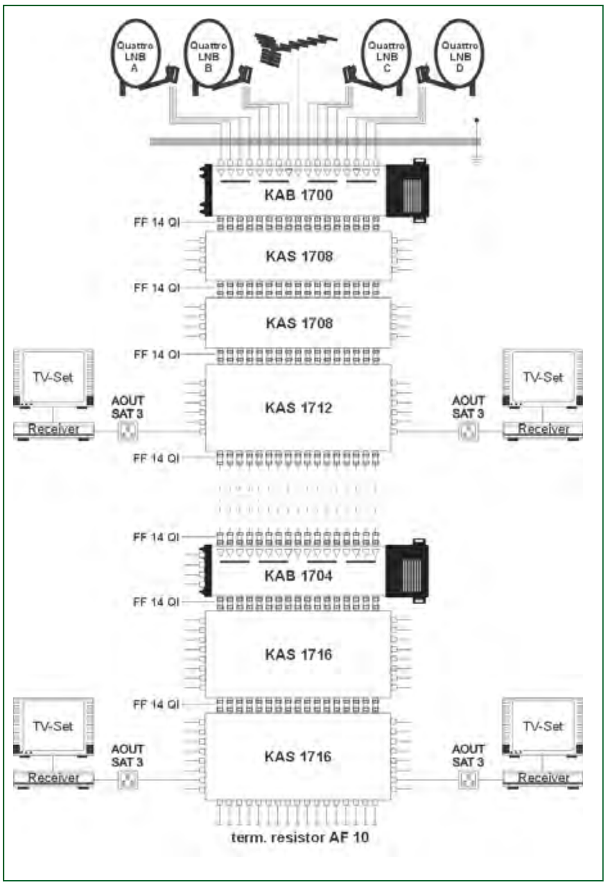


Figure 3. Cascaded Multiswitches distribute the signals from four satellites (16 satellite bands plus 1 terrestrial) to more than 100 subscribers on the left and right of the diagram while input bands are bussed through to the next cascaded Multiswitch unit.

More recent Multiswitch units provide an additional input for a terrestrial UHF TV signal so that this signal can also be switched to the subscriber and sent over the same coax feed. The system can be expanded beyond 16 subscribers by using cascadeable Multiswitch units **Figure-1**). This installation allows up to 100 subscribers to be serviced by just one dish and would be suitable for a small block of apartments or subscribers in a neighbourhood. One advantage of this type of installation is that it is quite simple to expand a relatively simple one-satellite system up to a four-satellite system just by swapping the Multiswitch/LNB units, the existing subscriber coax cabling need not be disturbed or expanded.
As can be seen in **Figure-3** all the received signal bands

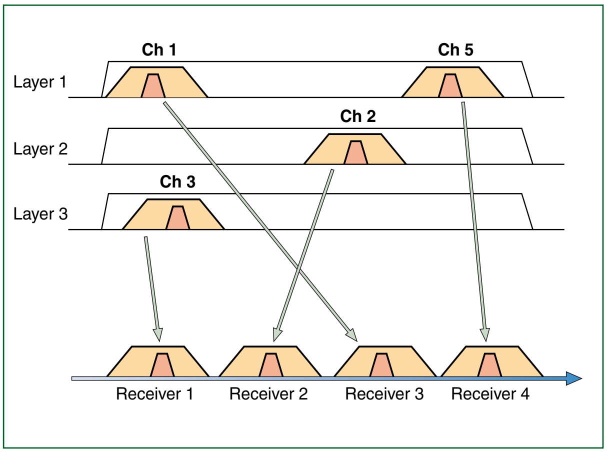


Figure 4. The Unicable system employs channel routing: The Unicable LNB takes the programme of interest from the satellite signal and converts it to a specific satellite IF which will only be detected by the requesting receiver. Programmes for up to eight receivers can be routed onto the same coax. (illustration: STMicroelectronics)

Web links

ST7LNB LNB microcontroller:
http://mcu.st.com/mcu/inchtml.php?fdir=pages&fnam=st7lnb

Channel Router Chip:
www.st.com/stonline/products/literature/bd/10465.htm

DiSEqC:
www.eutelsat.com/satellites/pdf/Diseqc/Reference%20docs/bus_spec.pdf

are bussed to the inputs of all the cascaded Multiswitch units, signal buffering and amplification may be necessary to compensate for signal attenuation and correct terminating resistors should be used at the end of the bus.

DiSEqC switching

Using switched quad LNBs together with a DiSEqC switch it is possible to produce a relatively small system suitable for four tuners which can receive programmes from four satellites.
DiSEqC commands are generated when a receiver requests a programme to route the receiver input through to the correct LNB via the DiSEqC switch, selection of the correct band is performed with 14/18-V voltage level shifting and the 22-kHz tone switching which is routed through the switch to the correct LNB. These switches are available with two, four and more recently eight inputs. Satellite equipment suppliers offer so called '17 by 4' (17 4) –Multiswitch units which actually comprise of four individual 4-to-1 DiSEqC switches fitted in the unit together with a feed from a terrestrial TV signal. Power for the switch is taken directly from the receiver and routed through to the selected LNB via the coax cable so there is no requirement for an external power supply. This method of power distribution can sometimes be a source of interference to other subscribers when the supply loading is switched during channel selection.

The single cable solution

With the advent of digital satellite signal there have been a number of 'single cable' solutions to the problem of distributing the broadcasts especially for satellite receiver owners interested in one language only. Germany, for example, has one of the highest uptakes of satellite TV equipment in Europe (approx. 40% of households) and there are many satellite programmes available in this language.
One such system uses a special LNB which selects specific parts of the received bands carrying programmes of interest to German speakers and packs them all together on a single coax.
A potential disadvantage of this approach is if for any reason one of the TV stations changes their transponder it may no longer be possible to receive that programme without fitting a new LNB!

Channel Routers

The IC manufacturer STMicroelectronics has produced its own solution to the single-cable system and not

surprisingly it involves microcontrollers and integrated circuit wizardry. The system is known as 'Unicable' and employs an 8-bit microcontroller type ST7LNB integrated into an LNB. The single coax output supplies programme feeds for up to eight receivers, each with a fully independent program signal.
In this set up each satellite receiver or recorder has a fixed receive frequency in the SAT IF band and information about each receivers programme channel selection is sent to the controller in the LNB where the requested channel is selected and down-converted to the unique IF band for that particular receiver (**Figure-4**) programme signals for up to eight receivers are then sent along the coax. The LNB is in fact a highly integrated Quattro LNB together with a mini channel-router and microcontroller. The system uses the existing DiSEqC control protocol together with some application specific additional commands and can easily be expanded to cater for signals from more satellites.

This system can currently supply eight independent signals over a single daisy-chained coax which should be sufficient for an average home installation. Single receivers can be replaced by twin receivers without the need for any additional cabling. The price of a Unicable LNB is currently around £100-150 (145-210 euros) which is a little more expensive than a small Multiswitch together with a suitable LNB but the cabling costs are much lower.

(060228-1)

Glossary

LNB: Low Noise Block (down converter). Satellite signals are focussed into this unit by the dish. It converts the received 10.7 to 12.7-GHz signal into a 950 to 2150-MHz signal and sends it to the receiver over one or more coax cables.

Sat IF: Satellite receiver intermediate frequency. The frequency range from 950 to 2150 MHz sent from the LNB to the satellite tuner.

Twin Receiver: Receiver containing two independent tuners allowing the simultaneous reception of two programmes (one for the TV and the other for a video recorder).

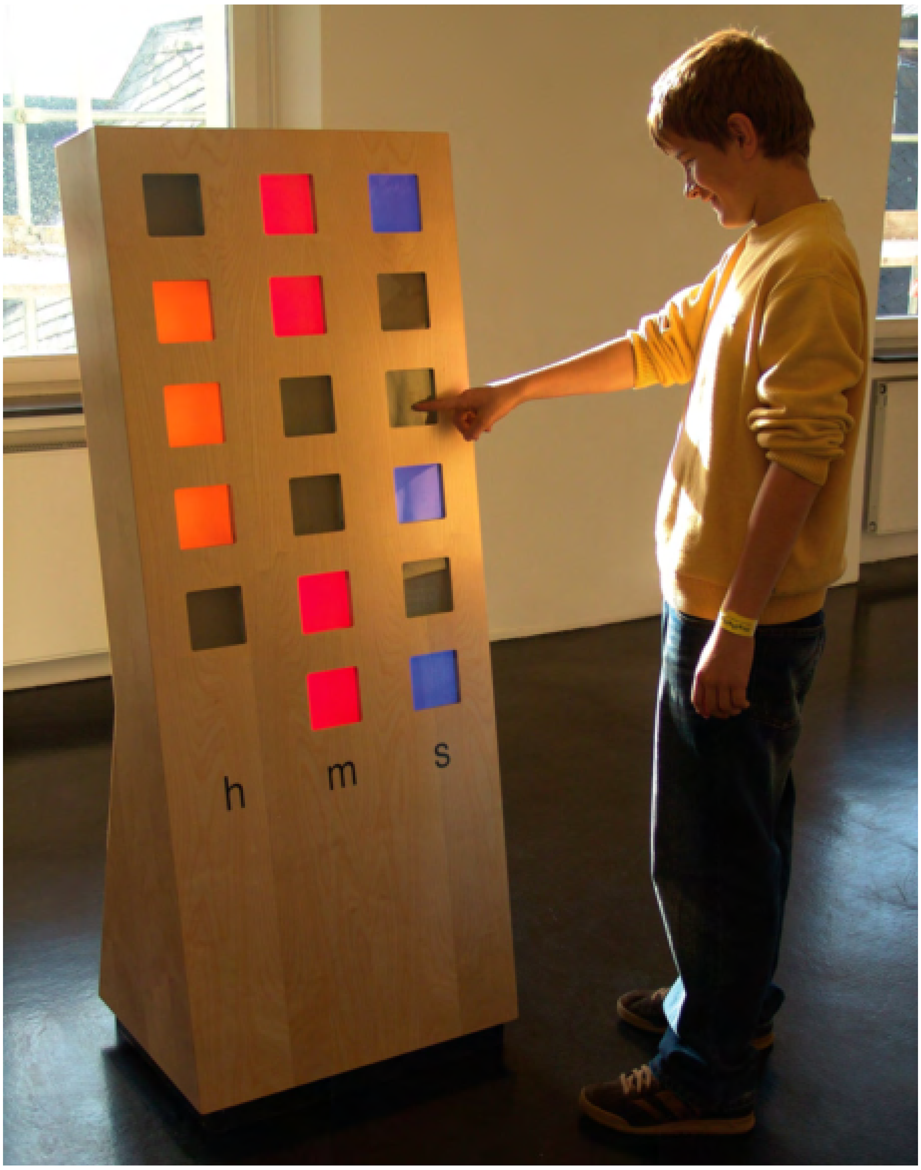
DiSEqC: Digital Satellite Equipment Control. A standard produced by Philips and Eutelsat in 1994 defining a control method between the LNB or Multiswitch and the receiver. (see web links and the DiSEqC moniter article in this magazine).

Transponder: Receive channel from a satellite. Analogue satellite TV used one channel per transponder but several compressed digital TV signals can now be packed into one transponder using a single wideband carrier.

Tri-State Time

Markus Bindhammer (Germany)

Inspired by a binary clock he saw at a mathematics museum the author of this design went one step further and came up with a more mathematically challenging timepiece...



This unusual large format digital clock in the photo shows the time in binary and is exhibited at the German Museum of Mathematics 'Mathematikum' in the town of Giessen, Germany [1]. The museum is the brainchild of Prof. Beutelsbacher and has been described as the first 'hands-on' mathematical museum in the world. The clock inspired Marco Freitag to design the PIC16C54 based binary clock featured earlier in this magazine [1]. After a visit to the museum the author was motivated to experiment with this alternative format for time representation which requires a little more concentration to read compared to the average clock.

The Trit

We are familiar with the role of the 'bit' in binary systems so it is probably not surprising to discover that in trinary systems we use the trit which is a contraction of the words 'trinary digit'. The basis of the trinary system (a.k.a. ternary or base three) is 3. The positional power of any number expressed in trinary (starting from the least significant position) is given by:

$3^0 = 1; 3^1 = 3; 3^2 = 9; 3^3 = 27;$ etc.

The only numbers allowed to represent a value are 0, 1 and 2. To indicate that a value is written in trinary it is appended with a subscripted 3, for example

$1210_3 = 48_d$
$(1 \times 27 + 2 \times 9 + 1 \times 3 + 0 \times 1)$.

In order to display any value in trinary it is necessary to use a device which can have three visible states. Bi-colour LEDs have been chosen here to represent '0' (both LEDs off), '1' (green LED on) and '2' (red LED on).
The table (**Figure 1**) shows the decimal values of a clock reading together with its representation in trinary and the corresponding LED colours used for the display. It can be seen that for the hours (0 to 23) we need three bi-colour LEDs and the minutes (0 to 59) require four. It is also necessary to use a corresponding number of counters to count the seconds and minutes. The counters go from 0 to 2 while the third output resets the counter to zero and acts as a carry-out to increment the next in the chain. The AND gate IC11.B detects when the hour count reaches 24 and resets the hours while IC11.D does the same to the minute counters when they reach 60 and also increment the hour counter.

Counting the time

Reading the time on this clock can be a little challenging but in contrast the circuit diagram is quite simple to follow. In principle it shouldn't be too difficult to read the time providing you can count up to three…
All of the timing and counting for the clock is not hidden away somewhere inside a microcontroller memory but instead is done the old fashioned way by wiring the hardware counters and

Quartz clock with a trinary display

logic shown on the circuit diagram in **Figure 2**. A division ratio 2^{15} (32768) is necessary to provide a 1 Hz signal from the 32.768 kHz watch crystal X1. The 14 stage binary counter IC10 can manage 2^{14} which gives a 2 Hz clock output from pin 3. Resistor R39 has a relatively high value and this helps reduce loading on the crystal. A small watch crystal like this can dissipate a maximum power of around 1 μW (nominally 0.1 μW).

The D-type flip-flop IC12A is configured as a divide-by-two to produce a 1 Hz output while IC8 and IC9 provide a divide-by-10 and divide-by-six function to generate minute pulses. IC12.B drives the second pulses to LED D4 which alternates between red and green. The use of 5-stage Johnson counters (which can count up to 10) for IC1 to IC7 may seem a bit like overkill since they never need to count above three but these devices are less problematic to interface than some of the

trinary	decimal	trinary	decimal	trinary	decimal
0 0 0 0	0	0 2 1 0	21	1 1 2 0	42
0 0 0 1	1	0 2 1 1	22	1 1 2 1	43
0 0 0 2	2	0 2 1 2	23	1 1 2 2	44
0 0 1 0	3	0 2 2 0	24	1 2 0 0	45
0 0 1 1	4	0 2 2 1	25	1 2 0 1	46
0 0 1 2	5	0 2 2 2	26	1 2 0 2	47
0 0 2 0	6	1 0 0 0	27	1 2 1 0	48
0 0 2 1	7	1 0 0 1	28	1 2 1 1	49
0 0 2 2	8	1 0 0 2	29	1 2 1 2	50
0 1 0 0	9	1 0 1 0	30	1 2 2 0	51
0 1 0 1	10	1 0 1 1	31	1 2 2 1	52
0 1 0 2	11	1 0 1 2	32	1 2 2 2	53
0 1 1 0	12	1 0 2 0	33	2 0 0 0	54
0 1 1 1	13	1 0 2 1	34	2 0 0 1	55
0 1 1 2	14	1 0 2 2	35	2 0 0 2	56
0 1 2 0	15	1 1 0 0	36	2 0 1 0	57
0 1 2 1	16	1 1 0 1	37	2 0 1 1	58
0 1 2 2	17	1 1 0 2	38	2 0 1 2	59
0 2 0 0	18	1 1 1 0	39	2 0 2 0	60
0 2 0 1	19	1 1 1 1	40		
0 2 0 2	20	1 1 1 2	41		

Figure 1. Time represented in trinary together with the decimal equivalent. The colours indicate the bi-colour LED display.

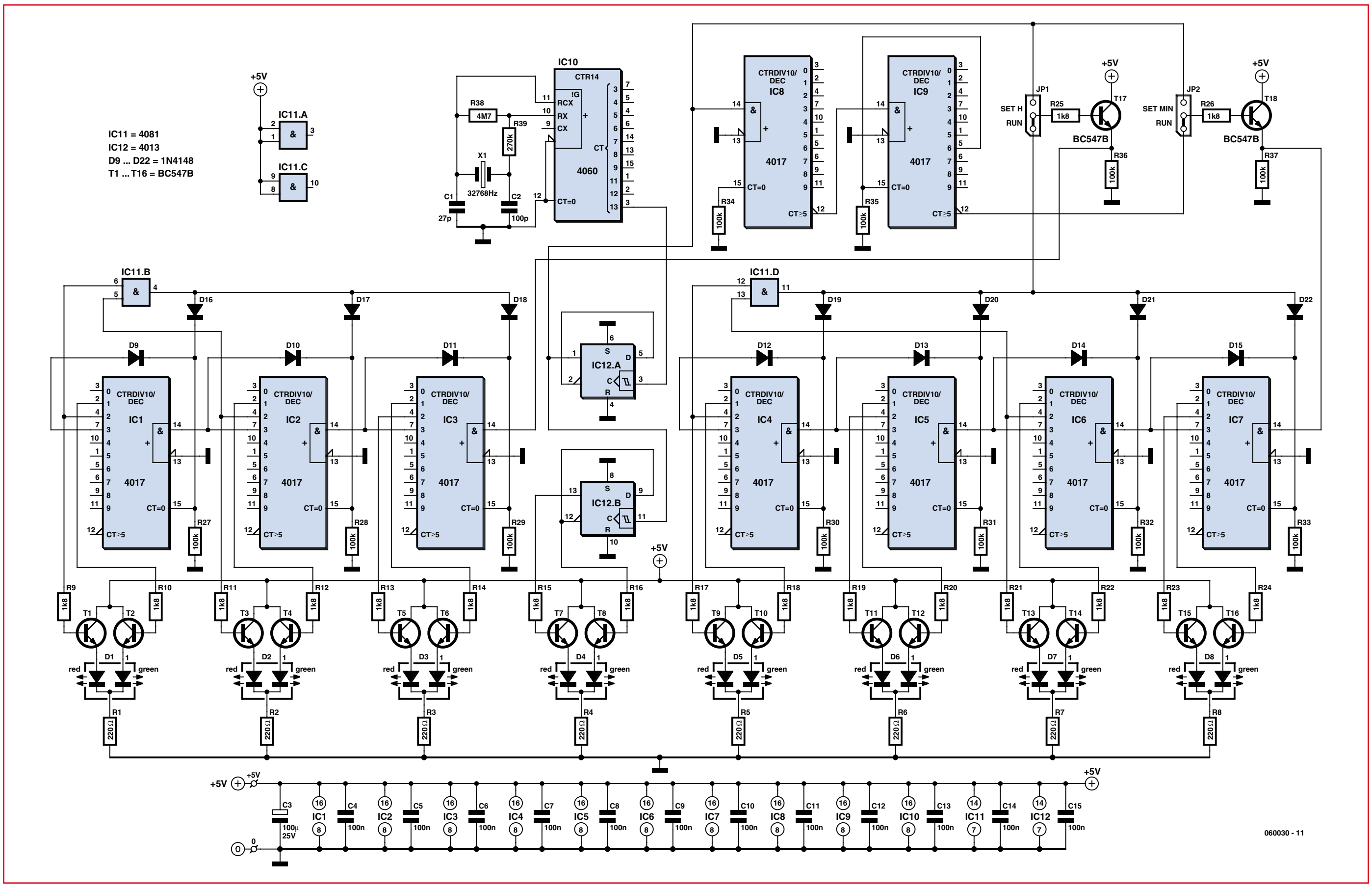


Figure 2. The circuit diagram shows a classic hardwired hardware approach.

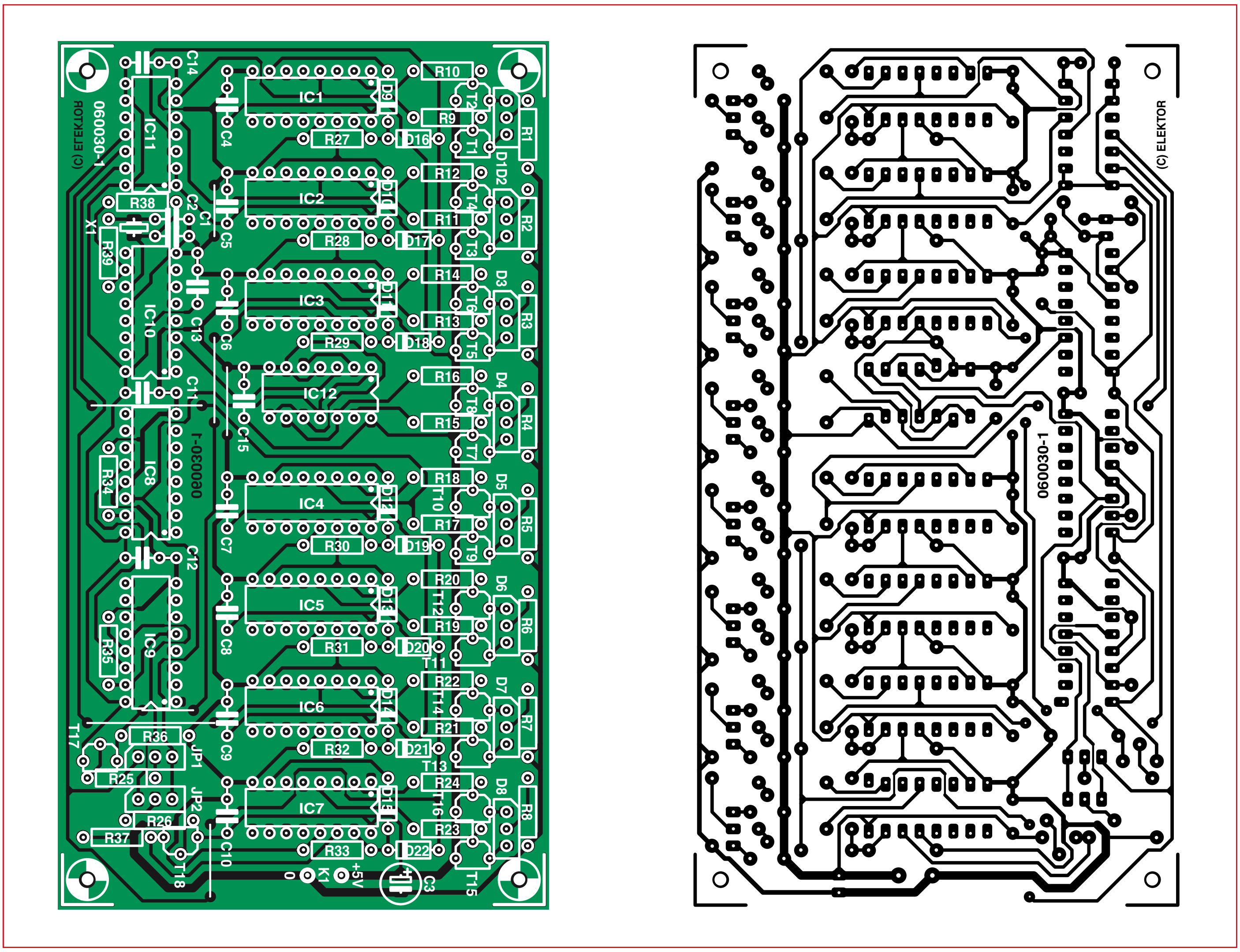


Figure 3. The finished PCB layout.

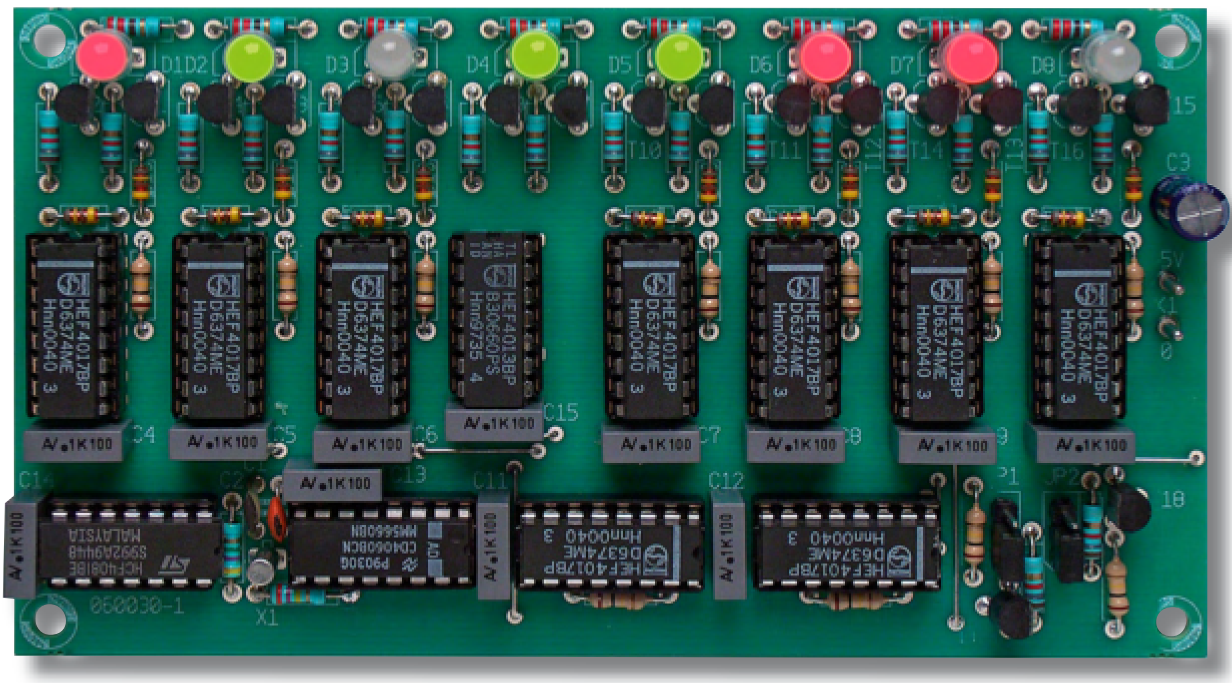


Figure 4. The prototype clock. Can you work out at what time the picture was taken?

other CMOS digital chips and produce a simpler circuit. With reference to the table LEDs D1 to D3 display the hours, D5 to D8 the minutes and D4 pulses at second intervals.

Setting the time

The PCB layout shown in **Figure 3** is rather reminiscent of a digital clock design from around 30 years ago when TTL technology was king. Thanks to CMOS circuitry and the LED outputs used in this design it consumes far less energy. The operating current depends on the number of LEDs illuminated which in turn is governed by the time of day. The prototype board shown in **Figure 4** draws between 22 and 88 mA. This current level would give a relatively short battery life but should be no problem for a low power 5 V, 100 mA mains adapter. The five volt supply does not need to be too precise; the circuit will function happily at any voltage between 4.5 and 6 V. To set the clock it is necessary to pull out the jumpers JP1 and JP2 from their RUN positions to the SET H (set hour) position and then SET MIN (set minute). The hours and the minutes will increment at second intervals until the correct value is achieved whereupon the jumpers are returned to their original position. The jumpers can be replaced by slide or toggle switches to make time setting easier.

With the component values given the accuracy of the crystal in the prototype was measured at +43 ppm which is within the quoted crystal tolerance. The crystal ran below its nominal value but those of you who have the means to measure the frequency accurately (or who have enough patience) can improve the accuracy by adjusting (trimming) the values of C1 and C2 slightly to change the capacitive loading on the crystal. Increasing the capacitance will slow the frequency.

(060030e)

Literature and Links:

[1] The German Museum of Mathematics: www.mathematikum.de

[2] Marco Freitag: 'Binary Clock', Elektor Electronics July/August 2006

COMPONENTS LIST

Resistors
R1-R8 = 220 Ω
R9-R26 = 1kΩ8
R27-R37 = 100kΩ
R38 = 4MΩ7
R39 = 270kΩ

Capacitors
C1 = 27pF
C2 = 100pF
C3 = 100µF 25V radial
C4-C15 = 100nF

Semiconductors
D1-D8 = bicolour LED, red/green, common cathode (e.g. Conrad Electronics # 185000)
D9- D22 = 1N4148
T1-T18 = BC547B
IC1-IC9 = 4017
IC10 = 4060
IC11 = 4081
IC12 = 4013

Miscellaneous
JP1,JP2 = 3-way SIL pinheader with jumper, or miniature changeover switch
X1 = 32.768kHz quartz crystal
7 wire links
PCB, ref. 060030-1 from www.thepcbshop.com

An Eye for Distance

Optical triangulation with the ATM18 board

Udo Jürsz and Wolfgang Rudolph (Germany)

People don't come with built-in rulers, but if we need to know how far away an object is, we can estimate the distance (and we do it all the time). However, how can a robot determine the distance to an object and do so with sufficient accuracy?
In this article, we examine the various methods that can be used and describe a distance measuring system that uses an infrared sensor and the ATM18.

Our ability to estimate distances accurately depends on many factors, such as how well we can see the remote object and whether we know the size of the object and other objects in its vicinity. In any case, our estimates are approximations and rarely exact. However, a moderately accurate estimate is usually sufficient for finding our way around in our surroundings. Things are different with a robot, for example when it has to adjust its speed and acceleration while approaching an object located 80 to 150 centimetres away.

We started by taking a closer look at various methods for determining the distance to an object. The following three methods are apparently the most important:

1. Propagation time and relative phase measurements using radio signals
2. Optical measurements (including laser measurements)
3. Measurements using ultrasonic signals

With regard to the last of these methods, we would like to make a small digression here to the animal world. As you know, bats use various sonar techniques with fixed frequencies and varying frequencies that yield a constant reflected frequency from stationary objects. The results are calculated so fast that these small aerobatic artists can navigate through narrow caves in full darkness with incredible virtuosity, and they can locate and capture insects in full flight. Although artificial ultrasonic measuring devices employ methods that are similar to those found in the animal world, our technology falls far short of the capabilities of natural sonar systems.

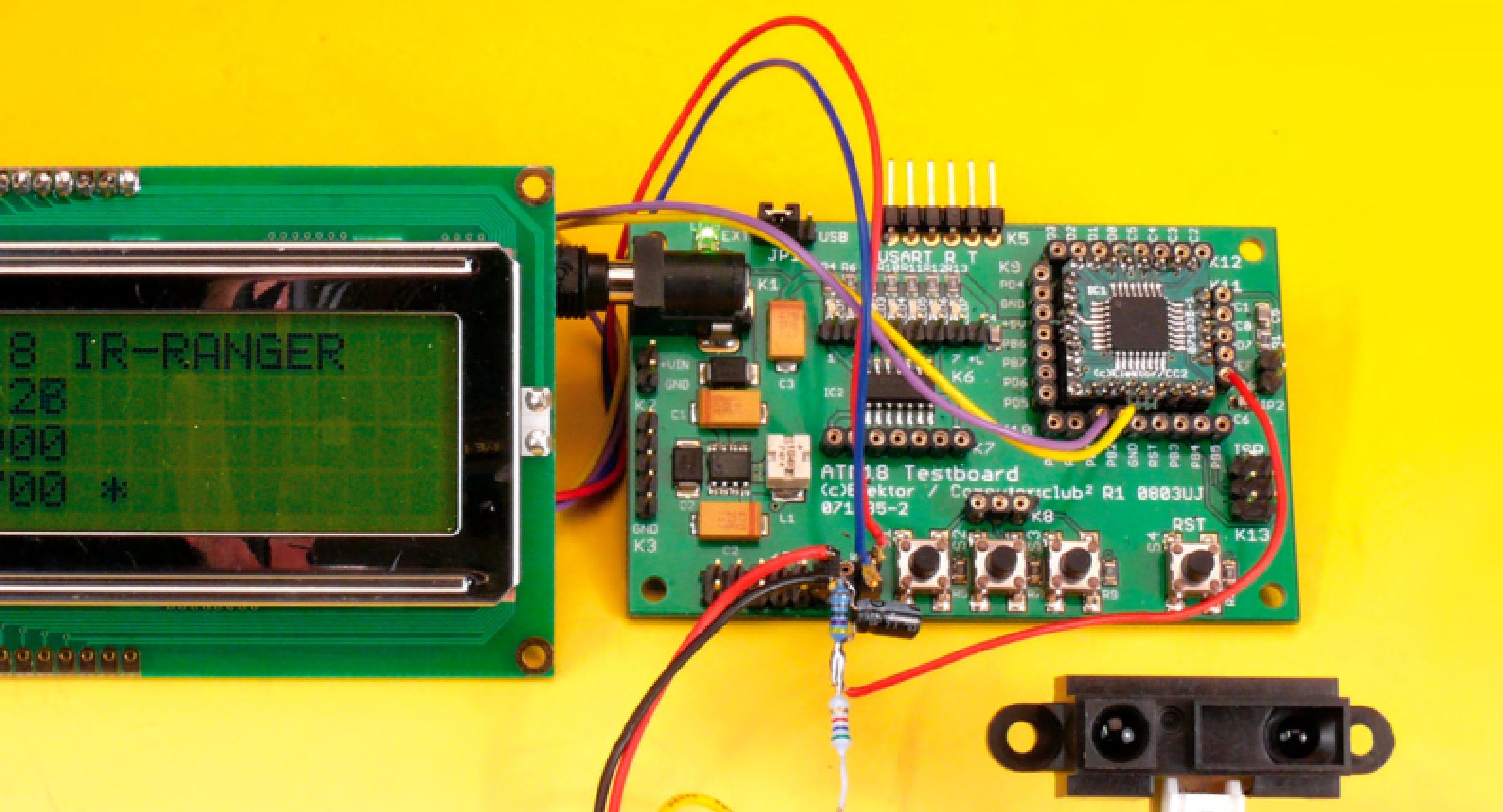
Every method has its advantages and disadvantages. Ultrasound is very sensitive to reflections and the physical properties of the atmosphere. Measurements based on the propagation time of radio signals require lightning-fast signal processing circuitry.

Optical triangulation [1] is a commonly used method for measuring distances with light.

Angle measurement

Optical triangulation is based on measuring the angle between emitted and reflected light beams instead of the propagation time of a light signal. Professional equipment uses laser diodes for this purpose in order to obtain high accuracy, but a normal LED can be used for relatively short distances if high accuracy is not necessary.

The operating principle of optical triangulation is shown in **Figure 1**. The LED at the left end of the sensor acts as the emitter. A precision lens forms the light emitted by the LED into a narrow beam that is reflected from the target object. A portion of the reflected light enters the lens of the receiver section of the sensor (at the right in the figure). The angle of the reflected light beam depends on the distance between the sensor and the target object. A 'position-sensitive detector' (PSD), or in other words a linear-array CCD IC, is located behind the receiver lens. The receiver lens focuses the reflected light beam into a spot that illuminates as few of the light-sensitive cells of the CCD array as possible, so that its position can be determined. If the distance to the target object changes, the angle α of the received light beam also changes, and a different part of the light sensor is illuminated (**Figure 2**). This clearly illustrates the operating principle: when the distance changes, the spot of light on the PSD (which



results from the reflected light beam) moves to a different position. The integrated signal processing circuit of the sensor can thus generate a signal voltage that depends on the angle α and thus on the distance. Unfortunately, the relationship between the signal value and the distance is not linear, since it is based on a trigonometric function.

The essential requirements for using this method are that the distance between the emitter diode and the receiver array of the sensor is known, as well as the angle α. The signal processing circuit obtains the latter value indirectly from the position of the light spot on the PSD. Using this information, the sensor's integrated signal processing circuit generates a signal that is available at the sensor output. Another consideration is that this method is only suitable for short distances (up to a few metres) because the sensitivity depends on the distance between the emitter and receiver sections, which are both contained in a small package.

If you want to determine the distance from the voltage generated by the sensor, you have to do some calculations. The following trigonometric formulas can be used to determine the distance $x - x_0$ from the measured distance $x' - x_0'$:

$$\tan\delta = \frac{x' - x'_0}{f} \rightarrow \tan\alpha = \frac{x_0}{D}$$

$$x = D \cdot \tan(\alpha + \delta) = D \cdot \frac{\tan\alpha + \tan\delta}{1 - \tan\alpha \cdot \tan\delta}$$

$$x = D \cdot \frac{\frac{x_0}{D} + \frac{x' - x'_0}{f}}{1 - \frac{x_0}{D} \cdot \frac{x' - x'_0}{f}}$$

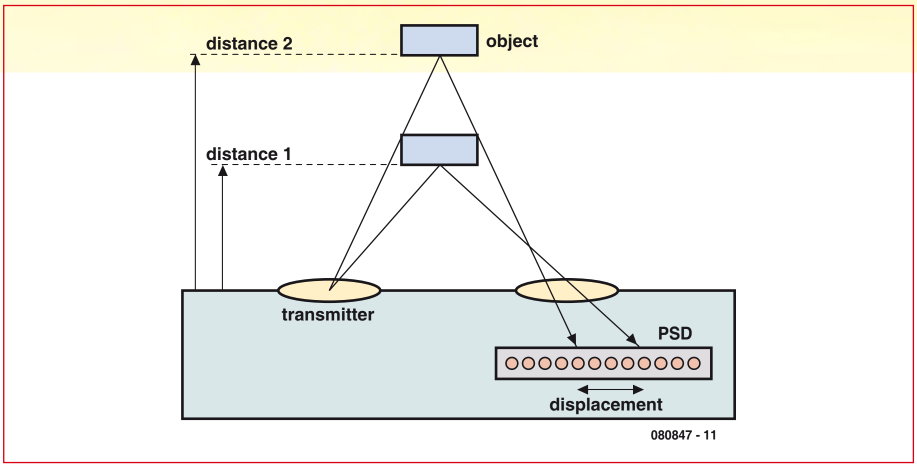


Figure 1. Distance measurement using optical triangulation.

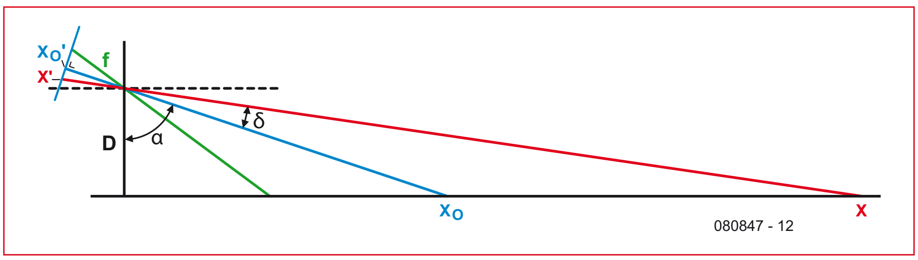


Figure 2. The distance $x - x_0$ can be determined from the measured distance $x' - x_0'$ by using trigonometry.

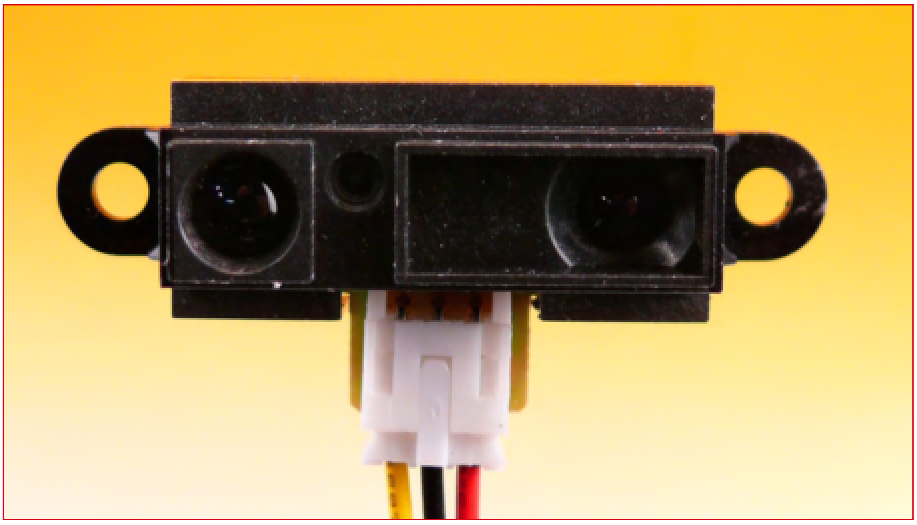


Figure 3. Sharp infrared distance sensor.

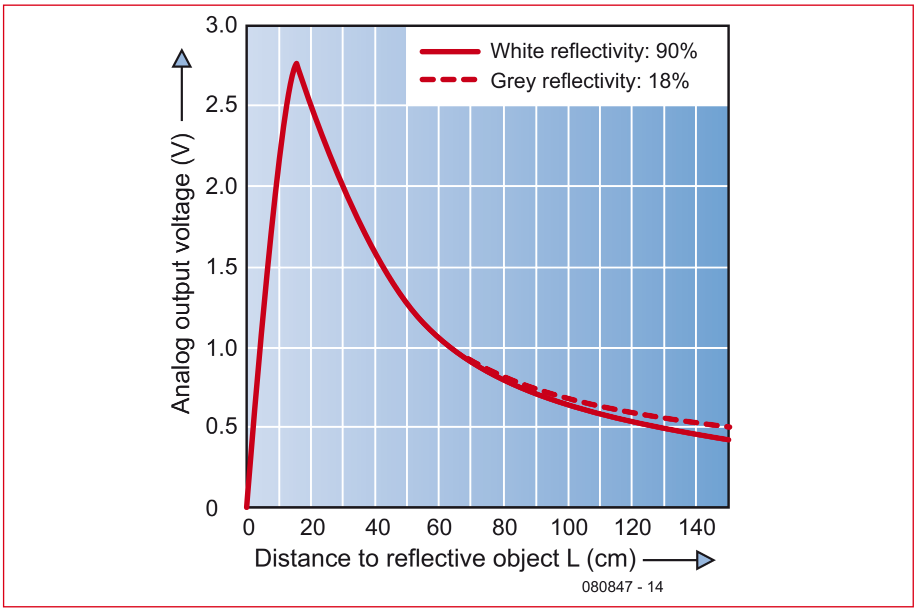


Figure 4. The relationship between output voltage and distance is non-linear.

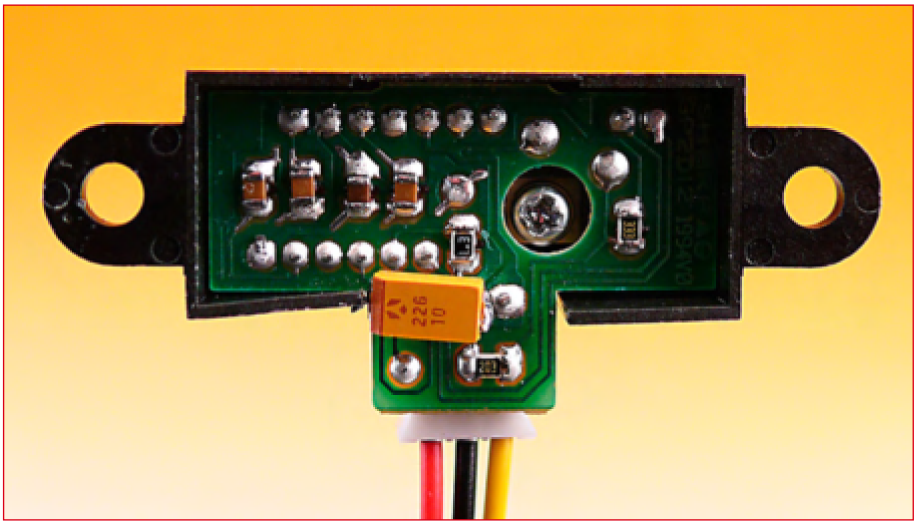


Figure 5. The sensor needs an additional SMD electrolytic capacitor for decoupling.

From the final formula for calculating the value of *x*, it should in any case be clear that our little 8-bit microcontroller has far too little processing power for continual measurement of the distance to any given object. There are other methods that can be used to determine the distance from the sensor output voltage without using a lot of processing power, but we don't want to get bogged down in theoretical aspects here. What matters now is putting the theory into practice!

IRDMS in practice

The focus of this project is using infrared distance sensors made by the Japanese manufacturer Sharp [2]. They go by the moniker 'IRDMS', which stands for 'infrared distance measurement sensor'. There are two different sorts of IRDMS sensors. One sort has digital outputs with an internal comparator set for a specific distance [3], while the other sort has analogue outputs. Here we use only sensors with analogue outputs. Several sensors suitable for different distance ranges are listed in **Table 1**. For our experiments, we selected the GP2Y0A02YK0F, which is intended to be used with distances of 20 to 150 cm. However, any other type listed in Table 1 can also be used, so you can select the type that best suits your particular application.

As you can see from **Figure 4**, the sensor output signal is highly non-linear. The distance cannot be derived directly from the signal without linearisation. However, this is not necessary for our initial experiments.

In theory, all you have to do to obtain a sensor signal with a range of up to approximately 2.7 V is to connect a 5-V supply voltage to the sensor. The IR diode operates in pulse mode and emits short, powerful flashes, which create a high peak load on the power supply. It is thus recommended to connect an electrolytic decoupling capacitor close to the sensor. Incidentally, the emitted light is in the near infrared range and is just barely visible to the naked eye in a dark environment, but it is readily visible on the monitor of a digital camera.

The internal linear array CCD has approximately 100 active pixels. As a result, the level of the output signal changes in steps of approximately 20 mV. A small ripple voltage with around the same amplitude is superimposed on the output signal, so a low-pass filter is a good idea. The 10-bit A/

D converter of the Mega88 has a resolution of around 5 mV with the external 5-V reference voltage, which in theory is adequate for this application. However, the C code developed for this project selects the microcontroller's internal 1.1-V reference voltage, which yields a resolution of approximately 1 mV. Caution: make sure that REF jumper JP2 is **not** fitted.

A voltage divider consisting of a 5.6-kΩ resistor and a 4.7-kΩ resistor must be connected ahead of the input to match the signal to the measuring range. With this arrangement, the measuring range of the microcontroller extends to 2.4 V. A 1-μF capacitor connected across one leg of the voltage divider lets it act as a low-pass filter as well.

The additional hardware is quite minimal. Aside from the two resistors for the voltage divider, you only need two capacitors. First you have to solder a capacitor with a value of 10 to 100 μF as close as possible to the sensor. It's beyond us why Sharp didn't

Table 1

Sharp IR distance sensors with analogue outputs, suitable for various distance ranges.

Type designation	Range [cm]
GP2D120XJ00F	4–30
GP2D12J0000F	10–80
GP2D15J0000F	10–80
GP2Y0A02YK0F	20–150
GP2Y0A710K0F	100–500

simply include this on the PCB in the sensor package. **Figure 5** shows this 'user enhancement' implemented with an SMD capacitor fitted directly to the PCB.. If you don't want to monkey with the sensor PCB, you can fit a small electrolytic capacitor externally, which means soldering it to the pins – but keep the leads as short as possible. Then you have to put together the combined voltage divider and low-pass filter, which as previously mentioned consists of a 6.8-kΩ resistor and a 4.7-kΩ resistor (preferably with a tolerance of 1% or better). Then solder a 1-μF capacitor across the 4.7-kΩ resistor (see **Figure 6**). Connect the junction of the voltage divider to the AD6 input. The full circuit on the prototyping board is shown in **Figure 7**. Connect buttons S1, S2 and S3 to PB3, PB4 and PB5, and connect the PC0 and PC1 outputs to any desired inputs (one each) of the ULN 2003 so they can be used to drive the associated LEDs (these connections are not shown in

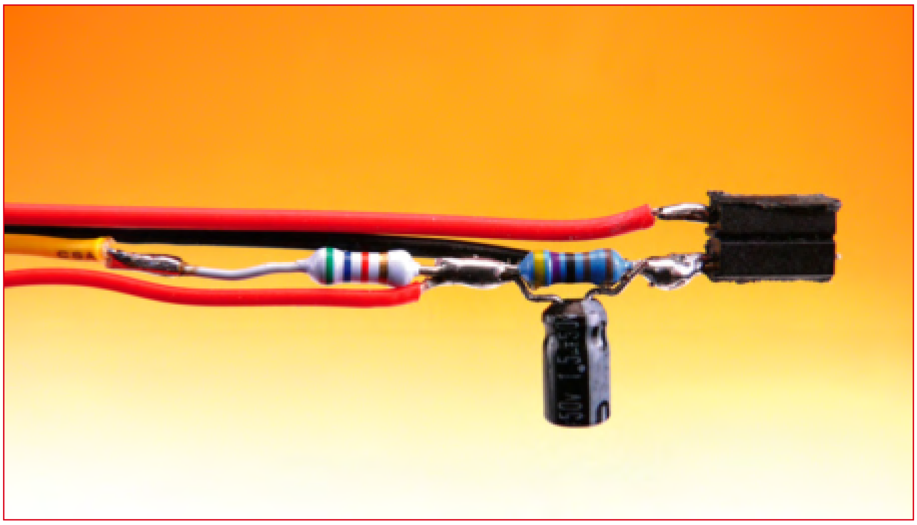


Figure 6. Two resistors and an electrolytic capacitor form a combined voltage divider and low-pass filter for the sensor signal.

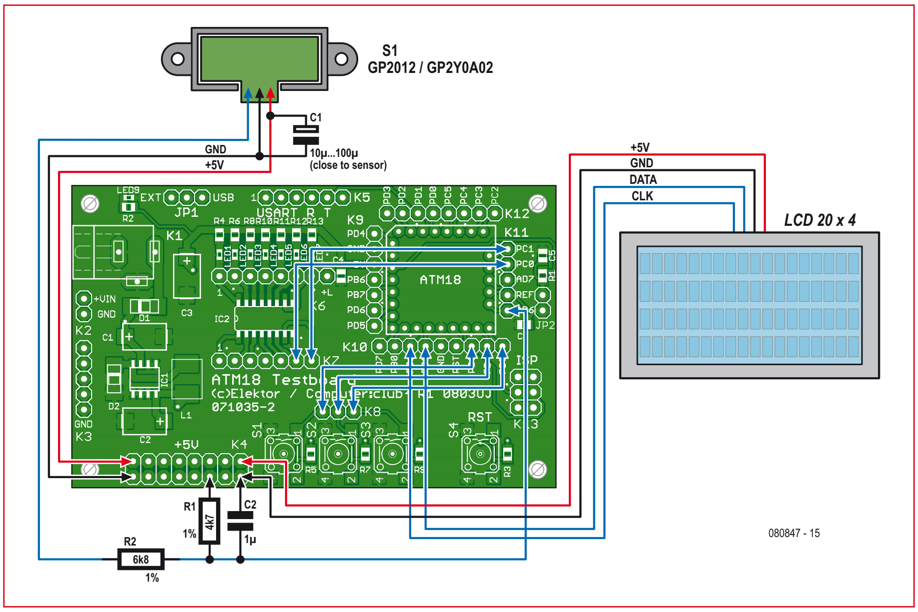


Figure 7. All connections at a glance.

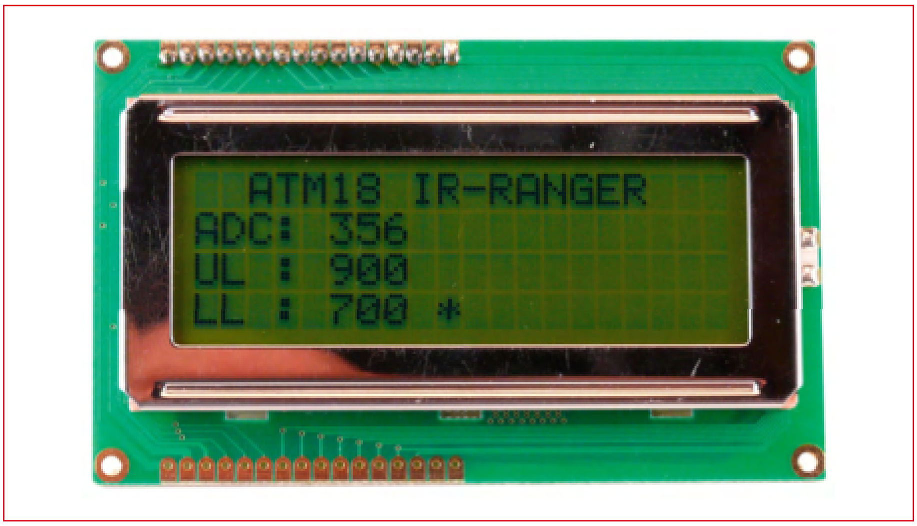


Figure 8. Displayed sensor values.

Listing 1

Distance calculation in Bascom

```
Sub Calculate_s
    D = Getadc(6)
' U = D/1023 * (4.7+5.6)/4.7
    'U = D * 5
    U = D * 1.1
    U = U * 10.3
    U = U / 4.7
    U = U / 1023
    'Print U
'           0.008271
   + 939.6 x Us
' S =  ----------------
   -----------------
'     1 - 3.398 x Us +
   17.339 x Us x Us
    If U > 0.4 Then
      S1 = 930.6 * U
      S1 = S1 + 0.008271
      S3 = U * U
      S3 = 17.339 * S3
      S2 = 3.398 * U
      S2 = 1 - S2
      S2 = S2 + S3
      S = S1 / S2
    Else
      S = 0
    End If
    Print S
End Sub

End
```

the photos).

Software

The C software for this project (ATM18_IRDMS_GP2xxx, downloadable from www.elektor.com) is quite straightforward in use. Two limit values are defined, and the program monitors these values and uses the LEDs to indicate the switching points. If the LC display is connected, three values are displayed: the output of the A/D converter (ADC: xxx), the upper limit (UL: xxxx), and the lower limit (LL: xxxx) (see **Figure 8**).
You can press S1 (left button) to set lower limit to the current sensor value, or press S3 (right button) to set the upper limit to the current sensor value. If you press the middle button (S2), the upper limit and lower limit are set to the default values.
After the limit values have been set, you can move almost any desired object around in the acquisition range, and the voltage generated by the sensor will be shown on the display. If the either of the limit values is reached, the corresponding LED on the prototyping board lights up. A possible application for this arrangement would be controlling a robot so that it never gets trapped in a corner and avoids obstacles. Of course, the distance parameter values could also be adjusted dynamically by the software according to the speed of motion.
For developing your own applications, we can provide a small tip here. You can determine the distance with reasonably good accuracy by using the following simple formula:

$$\text{Distance} = \frac{0.008271 + 939.6 \times U_S}{1 - 3.398 \times U_S + 17.339 \times U_S \times Us}$$

Here *Us* is the sensor signal voltage, which ranges from 2.5 V at a distance of 20 cm to 0.45 V at a distance of 150 cm.
Naturally, this can be calculated much faster than the previously stated formulas. It can also be used in a Bascom-based solution.
If you need to make distance measurements in an application and convert them to physical units, you naturally want to use the fastest possible method, which means using a look-up table. This involves creating a table of sensor output voltages for the entire distance range and having the software read values from this table.
However, the implementation described here provides an adequate starting point for enabling a mobile object to decide which action to take, similar to the way a bat navigates with its ultrasonic localisation system. If an obstacle is looming, the motors can be stopped, and if the object keeps on coming, they can be put into reverse. That's something even a bat can't do!

Lamp control in Bascom

The Bascom program (downloadable from www.elektor.com) uses the sensor for a simple lamp control instead of displaying the measured distance to an object. The lamp in question is a desk lamp, which is controlled via all bits of Port B. One option is to use the ULN2003 driver IC on the board to drive a relay.
In use, the distance sensor is aimed at the work station. If someone approaches the desk, the lamp goes on automatically. If they leave the work station, the lamp is switched off after a delay of 100 seconds. The movements of the person working at the desk are also monitored. With normal desk work, people constantly move around by more than 3 cm. If motion is no longer observed, the person being monitored has probably fallen asleep. In this case, the desk lamp is switched off in the interest of a good office nap. However, it switches back on immediately if the boss comes by and wakes his employee.
The function Calculate_s (**Listing 1**) makes a measurement and converts the result into the distance s in centimetres. The calculation must be performed in individual steps in Bascom; writing the full expression in a single line with lots of parentheses won't work here. The voltage measurement code takes into account the voltage divider (6.8 kΩ / 4.7 kΩ) and the internal reference voltage (1.1 V). The calculated distance is also sent to the PC via a 9600-baud link. Experience shows that the accuracy of the distance measurement is relatively good, with an error of around 10%. If no object is present in the visible range, a value of zero is output. The lines for an alternative sensor connection without a voltage divider, which requires using the 5-V supply voltage as the reference voltage, have been commented out.

(080847-1)

References and links

[1] Contactless Distance Measurement, Elektor Electronics, April 2002

[2] www.sharpsma.com

[3] Distance Measurement using Infrared, Elektor Electronics, July/August 2002

The ATM18 project at Computer:club²

ATM18 is a joint project of Elektor and Computer:club² (*www.cczwei.de*) in collaboration with Udo Jürsz, Chief Designer of *www.microdrones.de*. The latest developments and applications of the Elektor ATM18 are presented by Computer:club² member Wolfgang Rudolph in the CC²-tv programme broadcast on the German NRW-TV channel. The IR distance sensor and ATM18-AVR board combination described here was featured in ***instalment 25*** of CC²-tv.

CC²-tv is broadcast live by NRW-TV via the cable television network in North Rhine–Westphalia and as a LiveStream programme via the Internet (*www.nrw.tv/home/cc2*). CC²-tv is also available as a podcast from *www.cczwei.de* and – a few days later – from *sevenload.de*.

Starter Kit Professional for ARM

a low cost entry into ARM development

40% DISCOUNT!
Order now at www.elektor.com/arm-kit

This bundle is perfect for low cost entry into ARM development and is great for engineers who need to develop prototypes of electronic systems quickly using the super fast ARM 7 core from Atmel. The pack is fully compatible with the range of E-block accessories. Datasheets on each individual item are available separately.

Flowcode 3 for ARM (CD-ROM)	£119.00
E-blocks ARM board	£89.20
E-blocks LED board	£15.15
E-blocks LCD board	£20.00
E-blocks Switch board	£15.15
USB cable	£3.70
Power supply	£12.30
Total value	**£274.50**
Special offer	**£162.00**

Free P&P!

Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom
Tel. +44 20 8261 4509

elektor SHOP

Further information and ordering at www.elektor.com/arm-kit

Decibit 2.4 GHz RF Transceiver Development Kit

Jan Buiting (Elektor UK/USA Editorial)

Decibit DDK contents

- CD-ROM: programmer, AVRasm2, WinAVR, example code files, USB driver
- USB programmer
- USB cable
- 2 DCBT-24AX modules
- 1 DCBT-24R6 Tiny RC
- 2 Handheld Testers
- 6 AA batteries
- 1 printed datasheet

I never considered a battery holder for three AA batteries or a CD-ROM bulky components until I opened the DDK v. 2.1 RF development kit from Decibit. For sure, it's because of the tiny (25×10 mm) RF transceiver modules with even tinier ICs on them (like 5×5 mm and 4×4 mm).
Decibit rely on proven technology: Atmel's ATMega168 microcontroller in combination with an nRF24L01 transceiver chip from Nordic Semiconductor together from the IC complement of a series of intelligent 2.4 GHz (ISM band) transmitter/receiver (TX/RX) modules with ISP connectivity and, remarkably, the ability to run AVR firmware created by you, the user.

Plug & Play but AVRasm too

The DDK kit, advertised at US$ 69 at the time of writing (December 2008) has great educational value as there is so much to discover and develop for different levels of users. Schools, for example, may follow the plug and play approach: install CD on PC; plug module on USB programmer and run the batch file. Next, insert the programmed modules in HHTs (hand held testers), press a button on one HHT and observe the LEDs on the other located at the far end of the classroom.
More advanced users should explore the examples demonstrating different RF data transfer methods and perhaps expand or modify the example code chunks supplied. These cover one-way and two-way RF links using Shockburst protocols, with or without ACK(knowledge). There is even a ping-pong demo allowing you to check the TX/RX range by walking around (I was able to cover distances up to about 50 m in and around my home). You can see (well, kind of) the data send/resend actions on the HHT LEDs.

Development tools

The USB compatible programmer unit contained in the kit is a gem as it allows you to develop your own code for blowing into the ATmega on the modules. The ingredients: Decibit's own programming software; AVRasm2 from Atmel, the WinAVR C/C++ Compiler and a USB driver. If you can make an ATmega write "hello world", you're equally okay to talk to the Nordic TX/RX on the Decibit module as if it was just another peripheral device like an LCD. For example, CALL_label is sufficient to do a data transfer. The abilities of this system go far beyond those of commonly found 'passive' transceiver modules listening to AT-style commands — the Decibit DCBT-24 modules can be controlled right down (or should we say 'up'?) to AVR assembly code level which makes them highly interesting to developers designing RF remote control applications 'the embedded way'.

Modules: you choose

- **DCBT-24N** (low-power ultra small size TX/RX);
- **DCBT-24B** (low-power TX/RX w. external antenna connection);
- **DCBT-24C** (80 mW power amplified TX/RX with SMA antenna connector);
- **DCBT-24R6** (6-button key for remote control).

For the TX/RX modules, several hardware/software versions are available, which customers can specify using the extensive ordering information system.

According to Decibit, ETSI and FCC approvals are pending for the DCBT-24 products. Also, RF modules with more pins will be released in the future, or versions based on the latest AVR technology like XMEGA. For now, the ATmega168 does a fine job already.

(080868-I)

www.decibit.com

BASCOM AVR Course (6)

A DDS generator using the ATmega32

Burkhard Kainka (Germany)

The first five instalments of this course have already covered many programming techniques with an emphasis on practical applications. This final instalment builds on this theme giving a detailed insight into all the software routines needed to make a simple DDS generator. We also have an offer for you, see the final page of this article!

To get a good overview of some of the possibilities of the BASCOM language and the ATmega controllers it is worth looking through the BASCOM AVR help pages, particularly all the Config options (**Figure 1**). Some of the topics we have already covered in this course and in other ATM-18 projects include:

- COM-interface, hardware and software
- Ports, input, input pull-ups, output
- A/D converter
- Timer and counter
- Timer interrupts
- PWM outputs
- RC5 input
- I2C master
- Servo impulse
- One-wire bus (elsewhere in this edition)

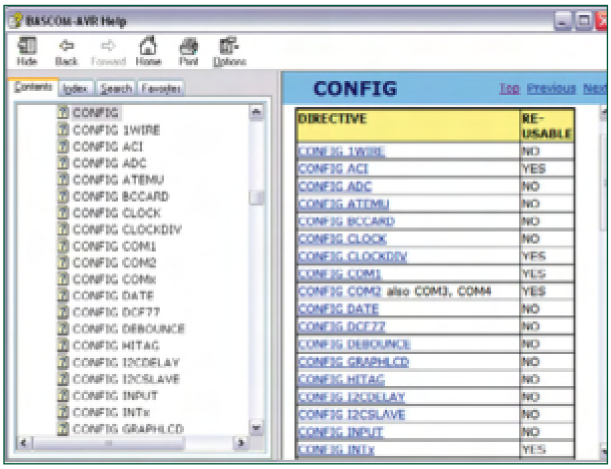


Figure 1. Bascom help for the config options.

There is of course much more internal and external hardware that can be used. The ATmega family share the same basic core but more specialised applications call for a careful study of the datasheets to find the version best suited to the task.

This month we build an AF generator using the principles of DDS to produce the signal. The microcontroller interfaces to an LCD which is driven from the port pins.

The DDS generator produces a sine wave signal from a digital PWM output. Two pushbuttons increment or decrement the output frequency in steps of 10 Hz. The output frequency is shown on an LCD and also sent to the COM interface port. For this test a Mega32 type controller has been used, it has many I/O pins and is a popular choice. It would be simple to make changes to allow the program to run on other controllers from the ATmega family.

The principle

The block diagram in **Figure 2** gives an overview of the external components connected to the microcontroller. The PWM signal is output from pin OC1B, and passes through a low-pass filter to produce a sine wave. The filter design is very simple but for test purposes is sufficient. A better solu-

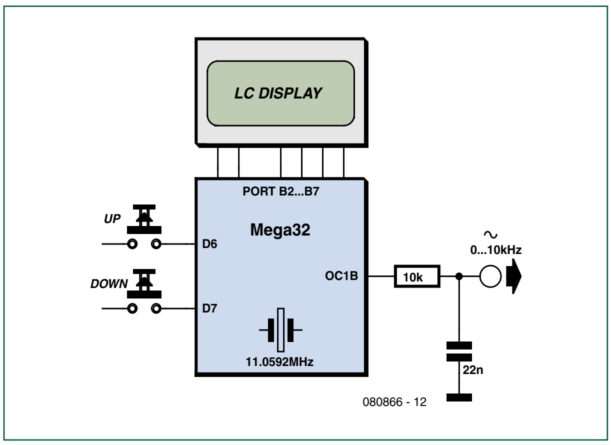


Figure 2. Block diagram of the DDS generator.

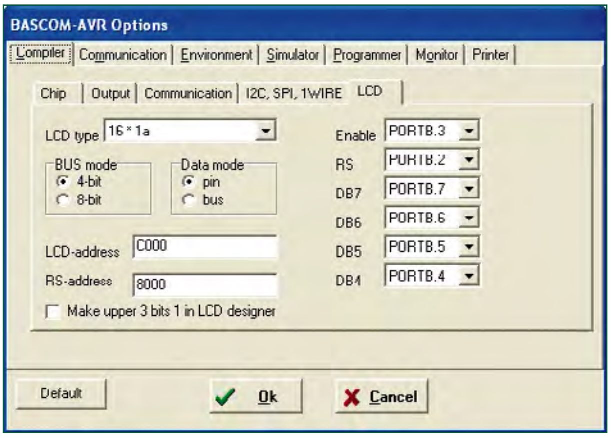


Figure 3.
The LCD setup page.

Listing 1

Initialising and writing to the LCD

```
Config Lcdpin = Pin , Db4 = Portb.4 ,
    Db5 = Portb.5 , Db6 = Portb.6 ,
    Db7 = Portb.7 , E = Portb.3 , Rs =
   Portb.2
Config Lcd = 16 * 2
Initlcd
Cls
Lcd "DDS"
```

Listing 2

Sine wave lookup table and frequency selection

```
For N = 1 To 256
  A = N - 1
  A = A * 3.1415
  A = A / 128
  B = Sin(a)
  B = B * 120
  B = B + 128
  Table(n) = Int(b)
Next N

Freq = 10
Do
  Locate 2 , 1
  Lcd Freq
  Lcd " Hz   "
  If Pind.6 = 0 Then
     Freq = Freq + 10
     Print Freq
  End If
  If Pind.7 = 0 Then
     Freq = Freq - 10
     Print Freq
  End If
  Waitms 10
  A = Freq
  '43200/65535
  B = A / 0.65918

  F = Int(b)
Loop
```

tion would use a (many-poled) filter with a much steeper response and a cut-off frequency of around 15 kHz. A simple piezo buzzer can be directly connected to the output without the need for any filter at all.

LC display

Whenever a design calls for an LCD to display lines of characters it can be achieved using six port pins for the drive signals. The LCD usually works in 4-bit mode with each data byte split into two before being sent to the display. Two control signals, E and RS, are also needed. In addition to these six signals we need three for the power supply and contrast setting.

The port pin assignments can be defined in the Menu Options/Compiler/LCD (**Figure 3**). A better method is to make the assignments in the source file (Config Lcdpin). This ensures that it runs successfully on different systems. It is also necessary to define the type of LCD used (Config Lcd = 16 * 2). When the system is first switched on one line of the LCD will be dark and the other light. After the display is initialised (Initlcd) the dark line becomes light. Characters (Lcd "Text") or a variable (LCD Freq) can now be sent to the display. After each character is written to the display the position is automatically incremented ready for the next character, but the second line does not automatically follow from the first. Writing to the second line it is necessary to define the position (Locate 2 , 5). The entire screen can be cleared at any time using the Cls command.

The example given in **Listing 1** writes the text 'DDS' to the first line. The frequency value is sent to the second line of the display (**Listing 2**) followed by the units 'Hz'. The displayed frequency value will not always have the same number of characters so it is necessary to include enough spaces to ensure that all characters previously displayed will be overwritten by the new value.

Sine table and frequency selection

A sine function lookup table must be written into memory for use by the DDS generator. The table size is 256 bytes. These represent the analogue values of the wave which are sent to the PWM output to produce the sine wave.

At start up the generator has an output frequency Freq = 10 Hz (**Listing 2**). The two pushbuttons PD6 and PD7 increment and decrement the output frequency in 10 Hz steps. The output frequency value is written to the two line display and each time the frequency is changed the new value is sent to the PC. Operation without an LCD is therefore possible. The generator can be used for example to tune an instrument; the standard 'A' note (440 Hz) produced on a tuning fork can be selected without problem. The frequency in Hz is scaled to give the variable F. Every change of F immediately affects the output frequency. This is made possible by the use of an interrupt routine.

Timer and DDS

The sine wave generator uses the DDS (Direct Digital Synthesis) principle with values of a sine waveform stored (as bytes) in a lookup table. A phase accumulator (the variable Accu) is increased by the value of the variable F to point to the next value in the lookup table. Only the high byte of the 16-bit Accu is used as a pointer to the table. When F has the value 1 it will therefore take 256 timer interrupts before the next value in the table is used and produces an output sine wave with a frequency of 0.65918 Hz. This is the resolution of the frequency generator. As the value of F increases the pointer steps through the table more quickly. When its value reaches 256 the pointer will start to jump over individual values but the output will still be sinusoidal. At the highest frequency of 10 kHz only around four values are used to produce a complete sine wave cycle. The lowpass filter ensures that a good approximation to a sine wave will still appear at the output.

The program uses two timers. Timer 1 generates the 8-bit PWM signal. In this setup the PWM frequency is 11059200 Hz / 256 = 43200 Hz. The 8-bit Timer 0 without any prescaler overflows at a rate of 43.2 kHz. This is therefore the rate at which the interrupt service routine is called, a new value is fetched from the lookup table and written to the PWM register before the next interrupt occurs (**Listing 3**).

Without any prescaler an 8-bit timer will interrupt every 256 clock cycles. Between interrupts the controller must not only execute all the instructions in the interrupt service routine, but also push all the working registers onto a stack and lastly pop them off again. In some cases the timing could be a little tight. It is important to be sure that there will be enough time to carry out all the activities. The simplest way to indicate how long the controller spends servicing the interrupt is to get it to set a port pin (Port.0 = 1) as it enters the ISR and reset it (Portb.0 = 0) when it exits. With an oscilloscope probe on the pin we can now observe the mark/space ratio directly to see how much time is available. In the example here the pin is high for less than 50% of the time. The main routine can only execute its tasks when this waveform is low. Using a simple software delay like Delayms will produce noticeably longer delay times than expected.

Two lines are 'commented out' in the interrupt routine. When these comment characters are removed the signal generator now has a sweep function. The frequency is incremented each time an interrupt occurs, the generator now sweeps from 0 to 10 kHz approximately three times per second. The oscilloscope display (**Figure 4**) shows the resulting output waveform after the low-pass filter. A piezo buzzer connected to the output will produce a characteristic twittering sound.

(080866-I)

Downloads and more info

Go to the project page at www.elektor.com/080866 for more information and the program downloads. We welcome your feedback in the Elektor forum.

BASCOM-AVR Reader Offer

Exclusive to Elektor readers, the download version of MCS BASCOM-AVR is now available at £ 55.00 (€ 69.00), a discount of more than 20% compared to the normal price of £ 71.00 (€ 89.00). As a bonus, pdf copies of all six BASCOM-AVR course instalments that appeared in Elektor are included with the download. The offer is valid from **19 January 2009 through 9 February 2009**. Further details at www.elektor.com/bascom-avr. US readers please check US$ prices on website.

Listing 3 The DDS

```
Config Timer1 = Pwm , Prescale = 1 , Pwm =
   8 ,
   Compare A Pwm = Clear Down ,
   Compare B Pwm = Clear Down
Config Timer0 = Timer , Prescale = 1
On Ovf0 Tim0_isr
Enable Timer0
Enable Interrupts
Pwm1a = 127
Pwm1b = 0

Tim0_isr:
'Timer 43.2 kHz at 11.0592 MHz
   Portb.0 = 1
   Accu = Accu + F
   N = High(accu)
   Pwm1b = Table(n)
   'F = F + 1
   'If F > 15000 Then F = 1
   Portb.0 = 0
Return
```

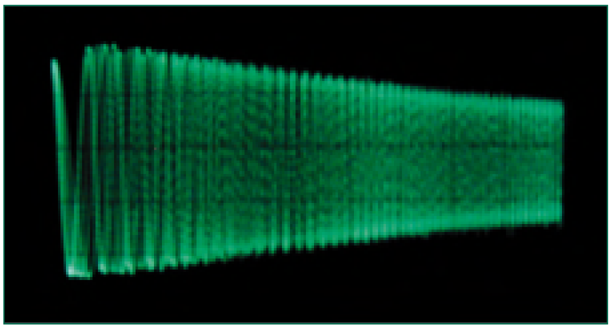


Figure 4. Oscilloscope display of the swept output signal.

Model Railway Car L

Märklin compatible & programmable

Graham Guthrie (South Africa)

One area that many model train enthusiasts have never been totally happy with is the lighting of coaches. Until now. The lighting controller described in this article is a combination of an SMD LED strip and a PIC12F683 for compatibility with the Märklin system. The LEDs allow easy adaptation to different colours while three different lengths of the board enable the project to fit all common coach types. The controller can be assigned an address without removing it from the coach.

Being able to turn the lighting on and off in a model railway coach used to be a challenge, as a separate decoder in the coach was at times not easy to conceal, and when installed, the address for the decoder was not easy to set or change. This challenge led the author to develop a solution that would address the following list of criteria:

- Cost effective with regard to buying commercial units.
- Easily adaptable to various colours.
- Switching individual coach lighting on and off.
- Controlling the intensity of lighting in individual coaches.
- Turning coach marker ('tail') lights on and off.
- Changing addresses of the decoders without having to open the coaches.
- Having a single feed for power linking coaches.
- Eliminating flicker due to the unbalanced Märklin track signal.
- Different length units to fit all common coach types.

The result has been the building of a coach lighting strip with an on-board controller. The board uses a PIC12F683 8-pin microcontroller to decode the incoming Märklin signals and control the intensity of the LED lighting. This microcontroller contains a PWM (pulsewidth modulator) and it can be obtained for a very reasonable price, keeping the cost of the unit low.

LEDs, colours & board sizes

In principle, any colour SMD LEDs may be used. When purchasing LEDs ensure that they are rated at least 100 mcd (milli-candelas) with a viewing angle of at least 110 degrees. White LEDs in a 0805 package with 140 mcd output and a viewing angle of 140 degrees work very well. For yellow(-ish) lighting, LEDs in a 1206 package are also a good choice. Experiment with a number of different LEDs until you find the ones that you think provide the colour that you like best.

The board is available in three lengths to accommodate different model coach lengths. The short board has a length of 110 mm and has four illuminating LEDs. The medium board has a length of 190 mm and accommodates eight LEDs. Finally, the long board is 230 mm in length and has 10 LEDs. All boards are single-sided and surface mount components are used throughout.

The circuit

As shown in **Figure 1,** the Märklin system supply voltage arrives via the rails on the bridge diodes D1–D4 via PC1 (Märklin RED) and PC5 (Märklin Brown). The output of the bridge is clamped by a 27-volt zener diode, D2, to protect the regulator from excessive voltage spikes that may occur on the track layout. The 22 μF capacitor C3 acts as a reservoir device to prevent the microcontroller from resetting owing to short periods of power loss. This power configuration also eliminates the flickering of the LEDs due to the unbalanced Märklin track signal. The 16-19 V raw DC output from the bridge is fed to the 5 volt regulator IC1 that in turn feeds the PIC microcontroller and the TLE4913 Hall effect switch, IC3.

The Hall switch pulls its output Low if a magnet is detected in its vicinity. Using software, this condition is used to enter the board address setup procedure discussed further on.

Output GP2 from the PIC microcontroller then drives an MMBF170 surface mount FET (T2) to turn the coach illumination LEDs on and off via PWM. A second output (GP4) and a second FET (T1) control two external LEDs that form the left and right tail lights on the coach. The tail lights should be connected between terminals 'A' (anode) and 'C' (cathode) and provided with their own current limiting resistor (2.2 kΩ is suggested). If you do not need tail lights, T1 and R9 may be omitted from the board.

It should be noted that the circuit diagram shown in Figure 1 is generic for the project. Three options exist for the construction — you decide.

ighting Decoder

with a magnet!

Quick project specs

- Märklin compatible
- PIC12F683 microcontroller
- Free software
- PWM light intensity control
- 3 board sizes available for 4, 8 or 10 LEDs
- Adaptable to various colours
- 'Live' changing of decoder address

The short board uses LEDs D6, D8, D10 and D15 only.
The medium board uses LEDs D6, D8, D10, D15, D7, D9, D11 and D13 only.
The long board uses all LEDs shown, i.e. D6, D8, D10, D15, D7, D9, D11, D13, D12 and D14.

The PIC micro

We're sorry if this sounds patronising to microcontroller boffins but the PIC 12F683 device used in this project is (1) an SMD device and (2) has to be programmed before it has any functionality: so, either you buy it ready-programmed from the Elektor Shop (order code **080689-41**) or you obtain a blank device from your favourite supplier and do the programming yourself using your own programming system and the software that's supplied free of charge through the Elektor website (archive file **080689-11.zip**). Assembly language and hex files are provided. The project software was developed using Microchip MPLAB IDE v3.14.

Construction

The boards all employ SMD components and those of not you not used to handling these tiny parts (or recovering them from mom's vacuum cleaner all the time) may want to seek help from fellow modellers in a club or on the Elektor forums, also when it comes to ordering parts and boards, using the motto: *Strength in Numbers!* Elektor does its part by supplying bundle discounts for the PCBs and the microcontroller, see the Shop section of our website.

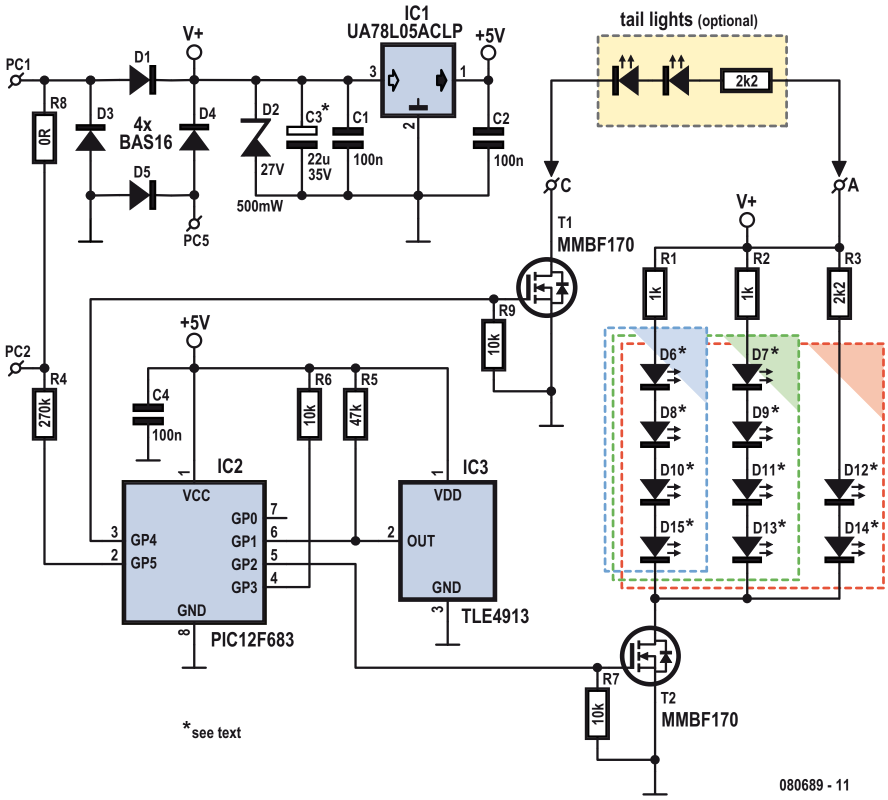


Figure 1. Circuit diagram of the coach lighting controller: basically, it's no more than a PIC micro and a series of LEDs under PWM control.

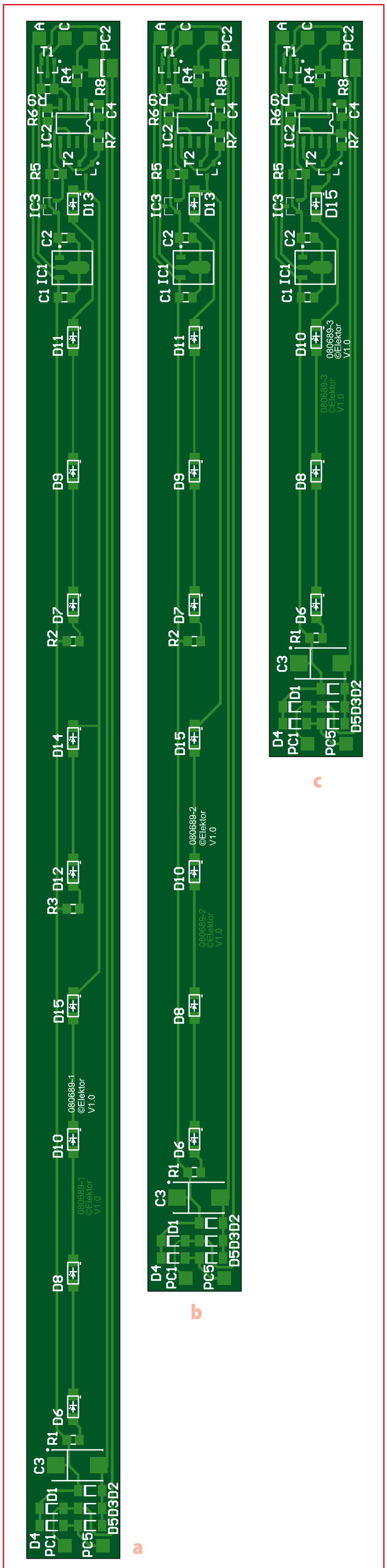


Figure 2. The control board comes in three sizes: long (a), medium (b) and short (c) to suit most model coach sizes. Full-size PCB artwork for all three board versions is available free of charge from the Elektor website.

The component mounting plans for the board are given in **Figure 2a** (long board), **2b** (medium board) and **2c** (short board). All three board sizes are available individually and ready-made from the Elektor Shop (order codes **080689-1**, **-2**, **-3**).

In the Elektor labs, it was proved that populating these boards is feasible with a steady hand, a fine tipped solder iron and nothing in the way of special SMD soldering tools. This is probably due to the fact that the parts are small but generously spaced on the board. You should start, however, by mounting the programmed microcontroller. A few more remarks. The length and spacing of the copper pads for the LEDs allow a wide range of devices to be used including ‘0805’ and ‘1206’.

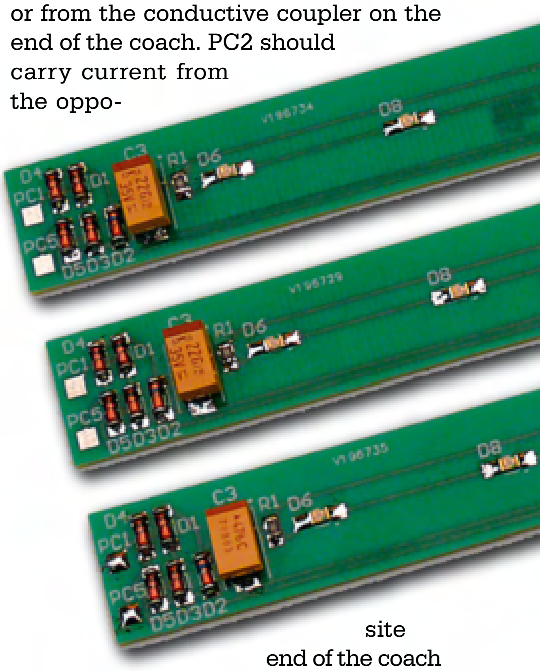
Be sure to know, verify again, and discuss with your friends, the polarity of the electrolytic capacitor C3 before fitting it on the board with absolute certainty. The electrolytics on our prototype boards had their **positive** (+) terminal marked with a red bar. C3 may be increased if you require the coach lights to stay on longer in the absence of power on the rails. You may even consider replacing C3 with a ‘Goldcap’.

Regarding the SMD diodes, when in doubt, do a polarity measurement or seek advice from your supplier.

It is recommended to test the board before mounting it in a coach. This is easily done by providing a temporary wired connection to the rails. Our thanks are due to Mr. Henk Prince for testing the three prototype boards in his privately owned Märklin layout.

Mounting it in the coach

The light strip should be attached to the inside roof of the carriage using Prestic adhesive or double sided tape. Ground (Märklin Brown) should be connected to ground pickup springs on a set of the coach’s wheels and routed to PC5 of the lighting board. The centre shoe pickup Red wire should be connected to PC1 on the lighting strip or from the conductive coupler on the end of the coach. PC2 should carry current from the opposite end of the coach to the opposite end conductive coupler to feed positive supply voltage to the following coach.



Setting up the board address

Having digested that in a Märklin system each device — whether locomotive, semaphore or turnouts — has its unique, individual device address, it’s no surprise that the coach lighting control is just another device to be incor-

COMPONENT LIST

Resistors
(all SMD 0805 except R8)
R1,R2 = 1kΩ
R3 = 2kΩ2
R4 =270kΩ
R5 = 47kΩ
R6,R7,R9 = 10kΩ
R8 = 0Ω (wire link) (SMD 1206)

Capacitors
C1,C2,C4 = 100nF (SMD 0805)
C3 = 22µF 35V tantalum (in D or X case)

Semiconductors
D1,D3,D4,D5 = BAS16
D2 = 27V 500mW zener diode in SOD-80C case, e.g. BZV55C27
D6-D14 = white or yellow SMD LED, min. 100mcd, min. 110 degrees (see text)
T1,T2 = PMBF170 (or equivalent for BS170 in SOT-23 case)
IC1 = 78L05 (SOT-89-3 case, e.g. L78L05ABUTR)
IC2 = PIC12F683-E/SN or –I/SN, programmed, Elektor Shop # **080689-41**
IC3 = TLE4913 (Hall sensor)

Miscellaneous
PCB, long version (l = 230mm), Elektor Shop # **080689-1***
PCB, medium version (l = 190mm), Elektor Shop # **080689-2***
PCB, short version (l = 110mm), Elektor Shop # **080689-3***

* select length to fit coach size.

porated in the system. The board is configured as follows, assuming it's working correctly, connected up and installed in a coach.

1. Place coach on rails.
2. Reset the Märklin control unit: Press and hold the stop and go buttons until control unit resets.
3. Enter address of engine, e.g. 06.
4. Hold a magnet over the end of the coach until the coach lights blink.
5. Remove magnet, then wait till light stops flashing.
6. Turn on engine lights using F0.
7. Adjust speed dial to adjust desired illumination level of lights.
7. Turn on F1 to save illumination level.
8. The coach light will blink a few times to indicate acceptance of the level.

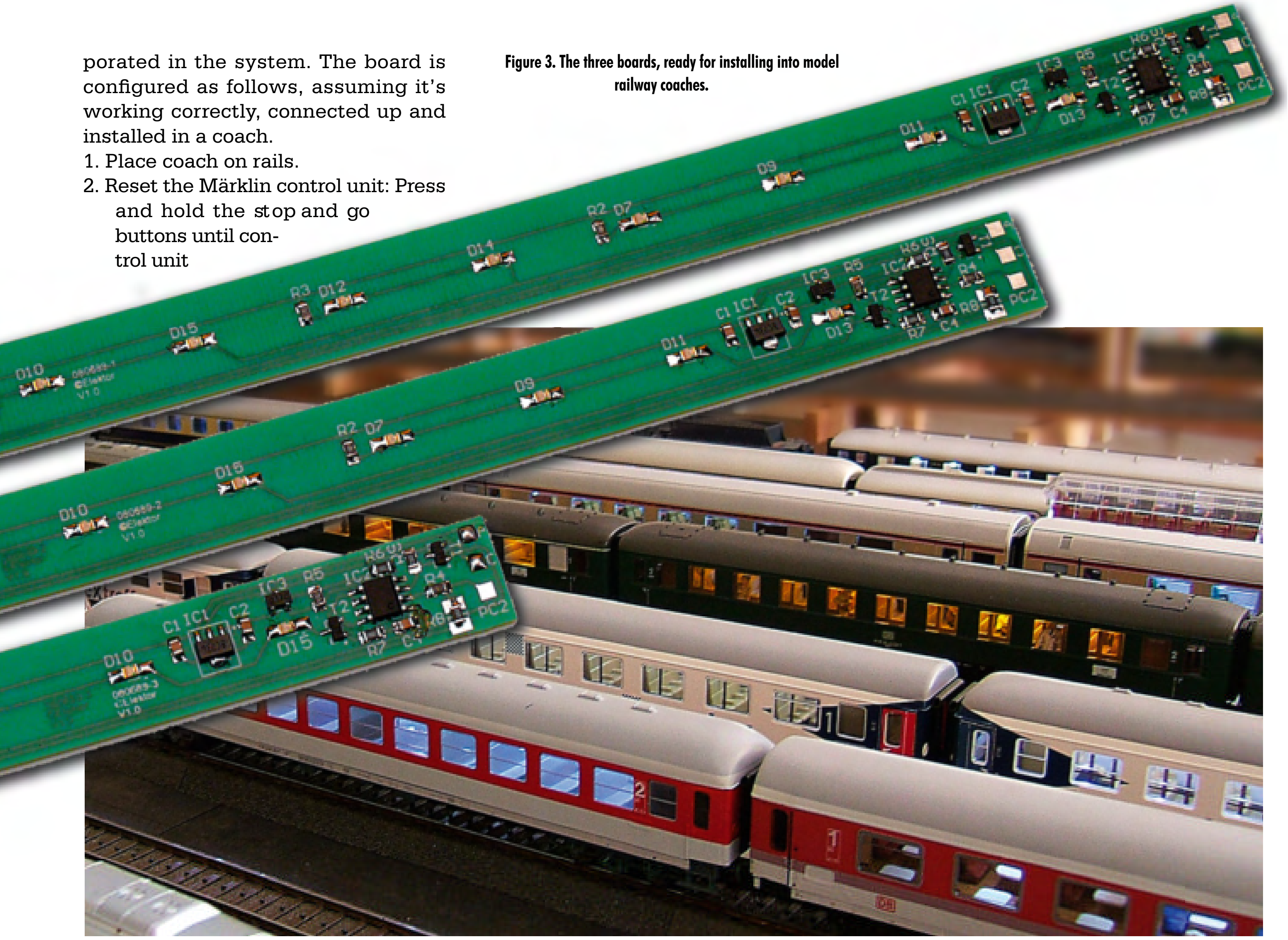


Figure 3. The three boards, ready for installing into model railway coaches.

The decoder is now programmed to address 06. Lights can now be turned on and off with the F0 buttons. If you have connected marker lights, then they can be operated using the F3 buttons. F1 has no function after setting the illumination level during programming setup.

Model railway fans, please send us your photographs of coaches fitted with the controller described here!

(080689-I)

Programmer settings

Device: PIC12F683 SOIC-8

Oscillator internal RC, no clock

Watchdog: enabled

Power Up Timer: enabled

CPD: disabled

Brown out: enabled

SBOREN: disabled

MCLR pin: enabled

Internal External Switch Over: disabled

Fail-Safe Clock Monitor: enabled

About the author

Graham Guthrie worked as an IBM customer Engineer for 31 years on mainframes and entered early retirement at the end of 1999.

After retiring he took up model trains as a hobby to keep himself occupied. Graham now makes various decoders and controllers for the local model train fraternity in his spare time.

He is also involved in helping others in the hobby with computer automation of their layouts.

Email: grahamg7@telkomsa.net

Programming FPGA

C-to-Hardware synthesis using Alt

Volker Brandstetter (Altium Germany)

FPGAs are normally programmed in special-purpose programming languages such as VHDL, which unfortunately can be hard to learn. Now, however, it is possible to program hardware in good old-fashioned C! Compared with conventional sequential execution of a program in a microcontroller, the results are considerably faster. We look at how C code can be compiled into hardware, using the Altium Designer development suite as an example.

FPGAs are hardware devices whose internal structure is determined by software. Compared with sequential program execution in a microcontroller, an FPGA can offer a considerable increase in speed; compared with ASICs FPGAs are more flexible and overall design costs for small to medium quantities are much lower.

Traditionally FPGAs (and ASICs) have been designed by specifying the structure and function of the circuits inside the device at the register transfer level (RTL) via a hardware description language such as Verilog or VHDL. Once the design has been simulated and verified the RTL description is converted into GDSII data for mask making (in the case of ASICs) or into binary configuration files (in the case of FPGAs).

The RTL development approach allows the hardware designer to work with the final silicon implementation in mind, and hence to optimise the circuit for speed or for power. Sometimes, however, these factors can be less important than low development costs or short development time.

In these respects the RTL approach has a number of disadvantages:

- Development is carried out at a low level of abstraction. A program that might run to 100 lines in a high-level language might easily balloon into a 1000-line RTL description.
- There are limited opportunities for code re-use as the RTL description specifies the structure of a system explicitly.
- When using an RTL description optimisation is usually carried out at the sub-system level; optimisation at a higher level can often have a greater impact on cost and performance.

C with benefits

Now we shall look at how so-called C-to-Hardware synthesis can address these disadvantages. Parts of a C program, that normally would be run on a microcontroller, can be transformed into FPGA hardware using the development suite, with an immediate performance gain.
One powerful development suite offering such a feature is Altium Designer from Australian software maker Altium (http://www.altium.com/products/altiumdesigner). The company also sells development boards which offer such a wide range of peripherals and expansion possibilities that they can be used directly for hardware prototyping. An example is the NanoBoard (see text box), which was used to test the sample applications described in this article.
The C-to-Hardware synthesis feature of the design suite allows designers to develop FPGA-based prototypes without having to learn Verilog or VHDL. C is used as the hardware description language, supported by a comprehensive IP library of ready-made FPGA modules including soft processor cores (microcontrollers implemented within the FPGA), memory controllers and various peripherals.
The C-to-Hardware synthesis process in Altium Designer converts an ISO C/C++ program into synthesisable RTL code. The compiler generates the circuit on the basis of a list of functional units (such as adders, ALUs, MACs, dividers and so on) and their characteristics. In doing so it endeavours to obtain the highest possible performance, for example by constructing functional units in such a way that they can be run in parallel.
The advantages of C-to-Hardware synthesis come to the fore when a processor core is integrated into the FPGA. C code can then either be executed on the processor or be converted into RTL code. The embedded compiler and C-to-Hardware synthesis are closely integrated in Altium Designer. Which functions are executed by the processor and which in hardware can be selected with just a couple of clicks of the mouse.

s in C

ium Designer

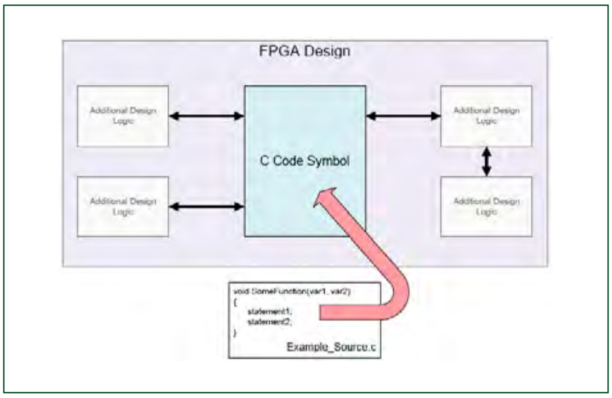


Figure 1.
A C function, represented using a C Code Symbol, is converted into an FPGA block and connected into the overall system.

The twofold way

In principle there are two ways of using the C-to-Hardware compiler.

1. The first use is in developing functional circuit blocks in C. In this case the compiler is used to convert individual C functions into hardware. A so-called 'C Code Symbol' makes it easy to integrate the function into the design. The result of the compilation process is an FPGA block that can be used as a module alongside other circuit blocks written in VHDL or Verilog, or ready-synthesised IP modules, connected via its defined inputs and outputs (**Figure 1**).
The C-to-Hardware compiler can convert the C function into a purely combinatorial circuit or into a multi-cycle circuit. This choice is made by setting the relevant property on the C Code Symbol (**Figure 2**).
In the case of combinatorial blocks only the parameters to the C function appear as inputs and outputs on the C Code Symbol. For multi-cycle circuits Altium Designer adds in the extra signals (such as clock and reset) that need to be connected.

2. The second use is in increasing the speed of a processor-based system. Here the C source code for selected software functions is converted into hardware in such a way that they can still be 'called' from the main embedded program, which is now running in a processor implemented in the FPGA. The C-to-Hardware compiler is used in conjunction with the embedded software compiler which is also part of the design suite. Implementing speed-critical functions in hardware can result in enormous increases in speed (**Figure 3**).

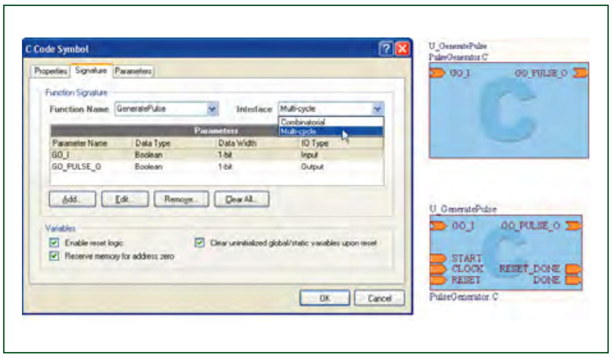


Figure 2.
Selecting a circuit type (combinatorial or multi-cycle) for a C Code Symbol.

Faster in parallel

Operations that are described as occurring sequentially in C but which could be run in parallel in hardware are converted into suitable RTL code by the compiler. For many algorithms, such as those used in image and signal processing, the parallel use of the available FPGA resources not only relieves the processor of computational burden but also provides a performance improvement.

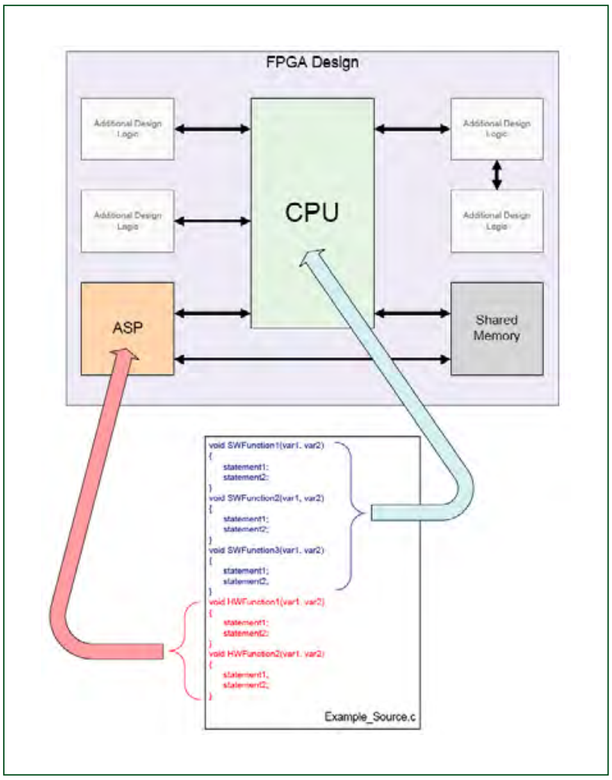


Figure 3.
Speed-critical C functions are implemented in hardware; less critical functions run on the CPU.

The so-called Application Specific Processor block (or ASP block) functions here as an interface between the operations being carried out in hardware and the remainder of the C program still running on the processor. When a hardware function is called from the embedded program, the processor passes the values of the function parameters to the ASP, which then initiates the function. When the hardware operation is complete it passes the results back to the main program.

The ASP therefore has access to a memory shared with the processor, through which the values of variables common to the hardware and software functions are communicated. The software can pass a pointer into this memory to the

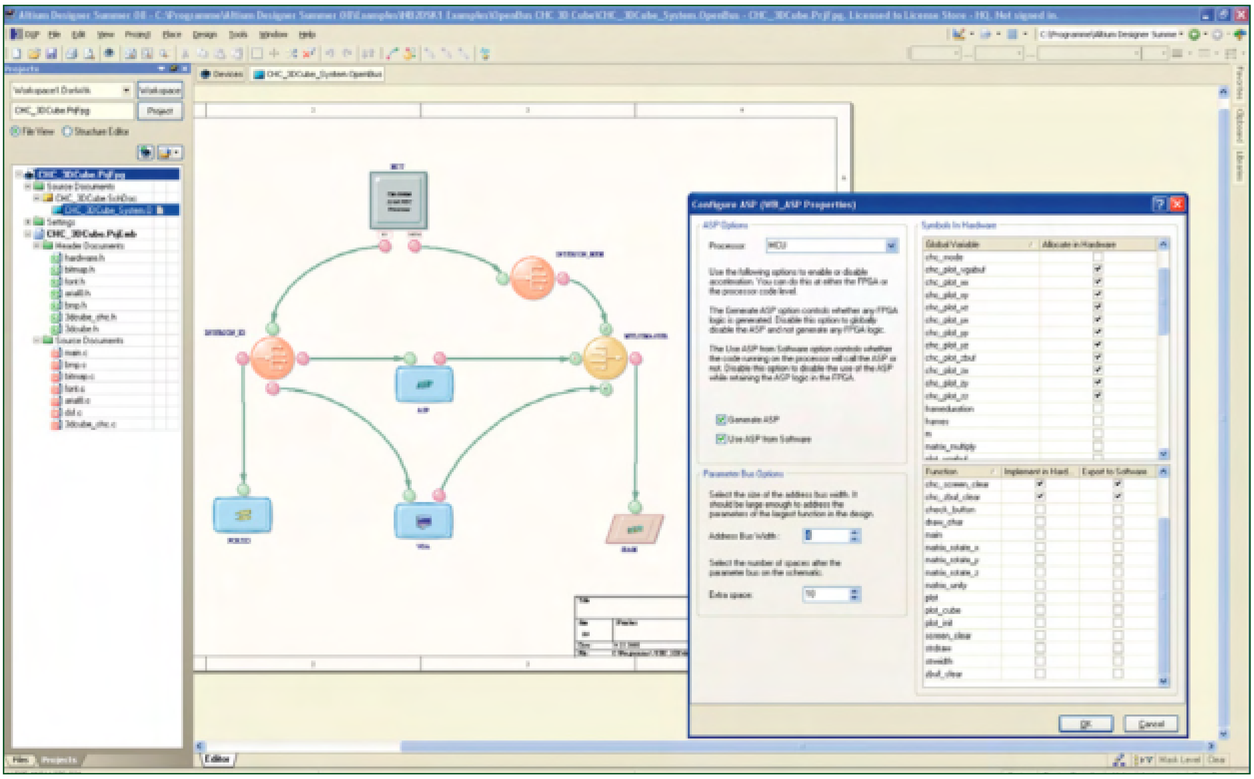


Figure 4. Intuitive configuration of an ASP block for C-to-Hardware synthesis.

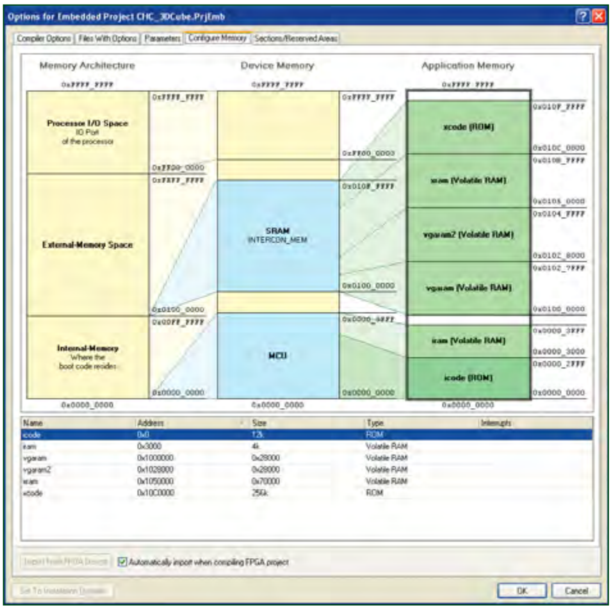


Figure 5. Configuring the memory map.

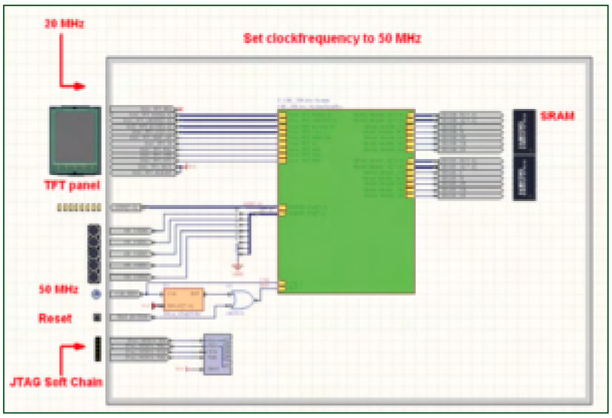


Figure 6. Connecting the FPGA to the resources provided by the NanoBoard.

ASP to indicate where the values of the function parameters are stored.
The ASP block is easily configured via a dialogue box (**Figure 4**). A list (upper right) allows the designer to specify which global variables are implemented in ASP block RAM. Access to this memory, internal to the ASP, is many times faster than access to memory which is also directly visible to the processor. A further list (lower right) lets the designer select which C functions are implemented in hardware and which in software.

Rotating 3D cube

Now we shall see how these ideas fit together using a demonstration example where we show a three-dimensional rotating cube on the NanoBoard's TFT display.

To enter the example system design into Altium we use the 'OpenBus' editor. The components needed to construct the system are selected from a list and then placed and connected together; all this can be done without needing to worry about the nitty-gritty of implementation.
The C source code is then entered using the integrated editor. Altium Designer includes a compiler, linker and debugger for the TSK3000 (a 32 bit RISC processor) as well as a range of other 8 bit and 32 bit processors.

When the individual components have been created and connected together the system can be configured using the corresponding forms. This includes the ASP configuration form described earlier as well as (for example) facilities for setting the address ranges occupied by various peripherals and the quantities of internal and external memory available to the processor (**Figure 5**). The memory map is displayed graphically, and any change made is automatically propagated to the embedded software project. This is one of the advantages of an integrated development environment.

The C program can be tested and verified in real time using the NanoBoard. The resources available on the NanoBoard and the interfaces to the FPGA daughter board are selected using special symbols from a supplied library, and connected to the FPGA circuit in the overall circuit diagram (**Figure 6**). For our 3D cube example we need to represent the TFT panel, the clock and reset signals, a LED display and two SRAMs. The document describing the FPGA circuit is wired into the schematic in a hierarchical fashion using an automatically generated symbol to represent it. It is connected to the resources of the NanoBoard in the overall circuit diagram.

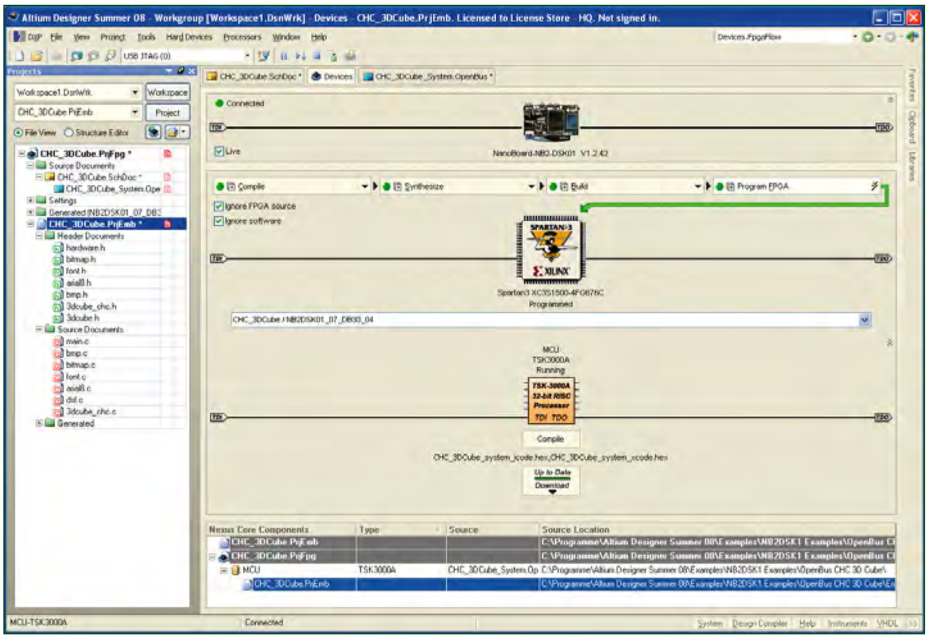


Figure 7. Altium Designer's Devices View, the 'cockpit' for the build and real-time verification process.

Chocks away!

When the system specification is complete the build process can be initiated (**Figure 7** shows the so-called 'cockpit'). The build process takes place in four steps. First, the C-to-Hardware synthesis step takes the selected functions and compiles the corresponding C source code. The subsequent steps (RTL synthesis, placement and routing) then create a binary configuration file suitable for programming the FPGA. The C program can be debugged by the design software using the so-called 'soft JTAG chain' while the program is running in the processor core implemented in the FPGA.

In our example the buttons arranged below the TFT panel allow the user to choose whether or not the C functions synthesised into hardware are used to compute the individual frames of the 3D cube display **(Figure 8)**.

A test shows that the processor working on its own can manage to compute around 2.7 frames per second. Using the ASP (that is, with the speed-critical functions implemented in hardware), the system runs around ten times faster, at approximately 28 frames per second.

(080625-I)

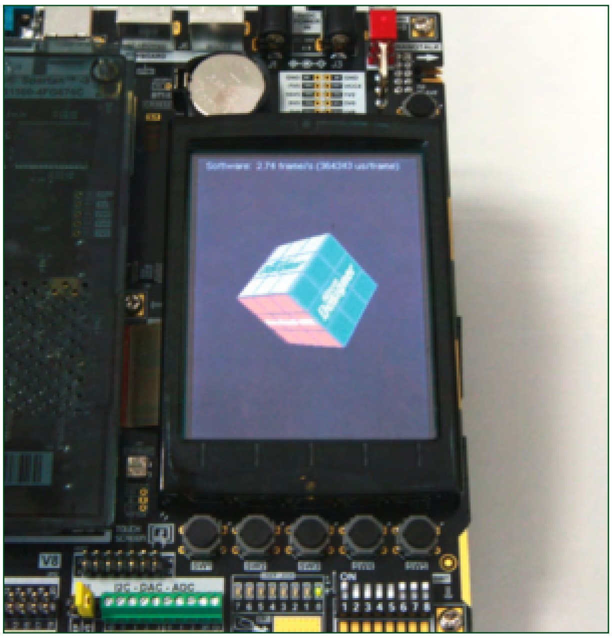


Figure 8. Using C-to-Hardware synthesis can give a factor of ten increase in speed!

NanoBoard

Alongside the Altium Designer development software, which provides an environment for circuit design, the development of embedded software and the design of FPGA-based systems, Altium also offers a reconfigurable hardware platform called the 'NanoBoard' (http://www.altium.com/products/thenanoboard).

A selection of plug-in peripheral boards supports a wide range of I/O and other hardware functions. Using swappable daughter boards various FPGAs and processors can be tested in the early stages of prototype system development.

C Sharp

PC programs using .NET and C# Part 1

Veikko Krypczyk (Germany)

Electronics people often need to write short PC programs, for instance to evaluate test and measurement data. Windows PC users will find the .NET Framework particularly handy for this task. It works with various languages, eliminates most of the burden of programming and doesn't cost a penny — not even the development environment. Here's a basic introduction to the C# language and its many advantages.

Today's electronics enthusiasts increasingly need to be software developers as well. An obvious situation is programming microcontrollers. Frequently the need arises for a short program written from scratch, perhaps for outputting control data to an interface card or for displaying measurement data on screen. Alternatively, an existing program may need to be adapted or expanded.

Software development is to a large extent bound up with the target platform and the operating system. In this series of articles we shall be looking at the development of programs running under Microsoft Windows XP and Vista. The underlying foundation is a runtime library that goes by the name of .NET Framework (pronounced 'dotnet'), which is supplied with Vista and is offered gratis as a retrofit for XP. Anyone using older versions of Windows need not go away empty-handed, as the support pages of Microsoft may offer a solution. Finally, it's well worth mentioning that an Open Source version of the .NET Framework has appeared in the meantime and this so-called 'Mono' framework [1] even runs on Linux computers!

Classes as programming modules

The introduction of the .NET Framework brought a significant change to software development for Windows. Greatly simplified, we're talking about a powerful library of Base Classes, offering a wide variety of 'oven ready' approaches for setting problems and getting answers, saving programmers a mountain of work. In the process the .NET Framework has averted a lot of long-winded discussion about the 'best' programming language. The choice of language has become a secondary consideration, as the Base Classes provide this themselves, and a common type system exists. Regardless of the language of the source code, a program is always translated into the same 'intermediate' code, which is subsequently executed by the .NET run-time environment (more correctly, it is recompiled when the program starts). The use of this 'Common Intermediate Language' brings with it security advantages and also enables programming that is substantially independent of the operating system. Porting across involves no more than changing the (lean) runtime environment.

Variants of C from Microsoft

For the.NET Framework there are languages like Visual Basic for .NET (from Microsoft) or Delphi for .NET (from Embarcadero Technologies). An overview of the languages compatible with the .NET Framework can be found for example at [2].

These two languages mentioned have been merely adapted for .NET (VB.NET is certainly not completely compatible with classic VB, although VB6 source code is fairly simple to adapt). On the other hand C# (pronounced 'C Sharp') has been developed by Microsoft from scratch specifically for the task. The complexity of the C and C++ languages, with their heavy learning curves, finds no place in C#. For example, pointer arithmetic (relished by few programmers) has fallen by the wayside. On the other hand, plenty of effective and user-friendly concepts from other languages have been incorporated. And another thing in favour of this variant of C: there exists an extremely compact .NET Framework for programming microcontrollers, which even includes a gratis compiler, in fact for C# [3]!

Understandably this two-part mini series cannot be a complete training course on C# but it may well encourage readers like you to get deeper involved with C#. In this first

Some language concepts of C#

Some of the language elements in C# may be familiar from other programming languages. All syntax questions can be resolved using the online help pages at [6].

1. C# treats upper and lower case letters as different, so the variable 'total' would be handled differently from the variable 'Total'.
2. Single-line comments are started with //, whereas /*...*/ indicates a comment with several lines of text.
3. Important fundamental data types include: byte, int, float, double, decimal and string.
4. Variables are declared and initialised by prefixing the data type:
   int Total = 5;
   string text ="one character string";
5. Basic operators: +, –, *, /
6. The flowing loops exist: for, while, do and foreach. The last-named iterates for example Objects assembled in listings (Collections).
7. Selection instructions are: If, else if, else and switch (...) case.
8. User-defined data types are declared with the keyword 'struct' (user-defined Classes are introduced with Class):

```
struct Dataset
{
        int Alter;
        string Name;
}
```

part we shall develop a small program with the help of the Visual Studio development environment, which is free to download. The second part will take us into object orientation, the foundation of the.NET Framework and 'made with C#'. Our practical example from the realm of 2-D graphics will create an application that should be particularly interesting for electronically-minded people.

And off we go!

To get developing, the first thing we need to do is make sure our computer is set up with everything we need. The only new item absolutely necessary is a C# compiler for translating the source code and as a bare minimum an editor can suffice for compiling the code. Far more practical and convenient, however, is a dedicated tool for the job, the Integrated Development Environment or IDE. Here you have basically two options. The first is a Microsoft offering, its so-called Express editions of Microsoft Visual Studio, available as a free download. The functional capability is somewhat restricted compared with the professional versions but this should not cause any problems on small to medium-size projects. Express editions are available for each of the languages C#, Visual Basic for .NET and C++. For Web development there is a further tool available, Visual Web Developer Express. For our mini-series Visual C# Express in the 2008 Version is the correct choice. Information on the products mention and downloading details are all at [4].

A free Open Source alternative is Sharp Develop [5], which also enables programming in C# (among other languages). This option requires you to have the .NET-Framework SDK (Software Development Kit) already installed.

Your first program

To create our first application, we need to boot up Microsoft Visual C# 2008 Express Edition. You will be greeted by this development environment's start page, which among other things enables you to open the last project you worked on. To start a new project click on File | New | Project.... Out of the options offered select 'Windows Forms Application' (**Figure 1**). Your second task is to select a name for the project. After confirming with OK the IDE produces a frame code for this application. The result is displayed immediately on your screen. You can confirm that this is already a complete application by making a first test run (start menu Debug | Debugging using the green arrowhead on the symbol tool bar, see **Figure 2**). Initially the application contains just an empty window. Nevertheless the necessary functions of a Windows application are already implemented. So you can for instance shift the window around the screen, minimise or close the application using the system menu.

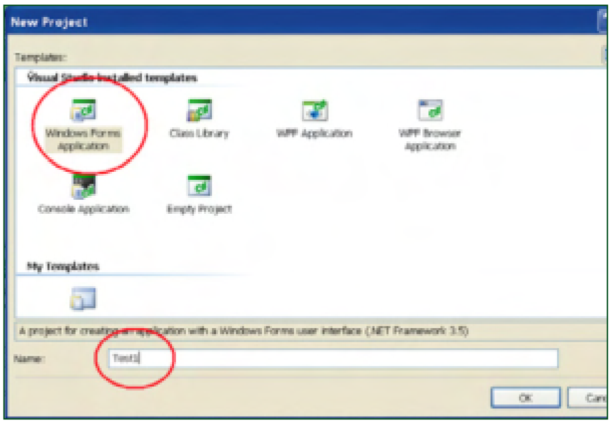


Figure 1. Constructing a Windows application (frame code) simply using the wizard.

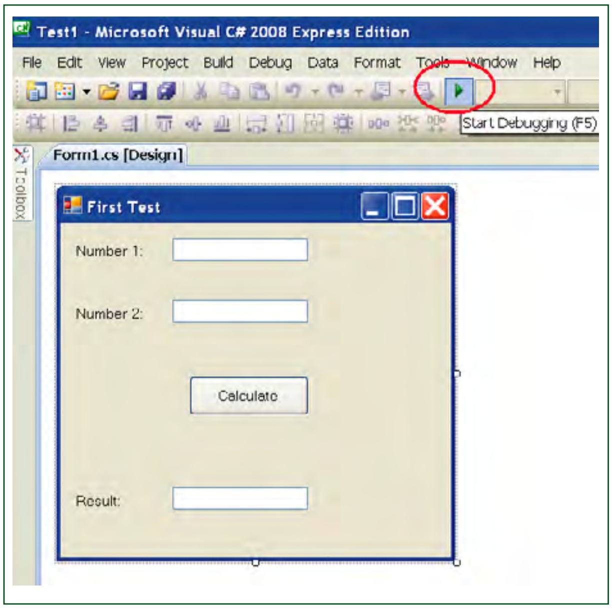


Figure 2. Starting an application in the development environment's Debug mode.

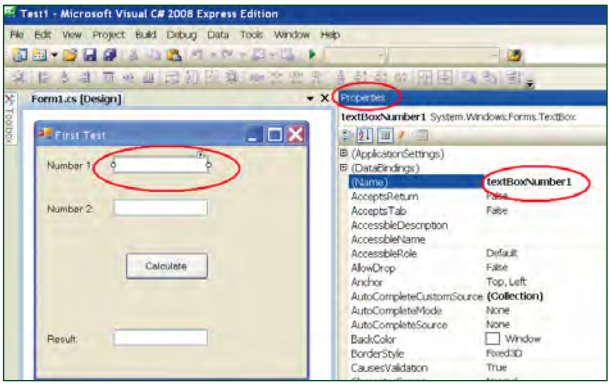


Figure 3. Editing an Object's characteristics in the Properties window is a breeze.

Figure 4. The first (extremely simple) application in action.

Ending the application reverts to the development environment, enabling us to expand the template with the first ideas of our own. So here's a simple example of how this can be done, gathering and adding together two figures to produce an output of the result in a text field. We'll start with creating the top-level information panel for the user. We require:

- three text fields for the figures and the result;
- three labels for the legend (descriptions);
- a button to start the calculation.

With the help of the IDE the formation proceeds apace. Call up the Toolbox (Appearance | Toolbox) in the active form. Now you can select the elements of the panel using the mouse and place them on the form. There are guide lines to help you get everything neat and tidy. Once an element has been installed on the panel we can edit its characteristics (Properties). This is done with ease using the Properties window (press the F4 function key for the Object highlighted). For example, we have assigned the tag 'textboxNumber' to the first text field's Property 'Name' (**Figure 3**).

Source code for the Button

Having designed the user panel it's time to start producing some code. The calculation needs to be performed when the corresponding button is clicked. A mouse click on this button while the program is running produces a so-called click event. It's the programmer's task to assign appropriate source code to events of this kind so that the code is then executed (in this case the addition of two numbers). To assist programmers the IDE produces the frame code automatically for the click event of a button, if the button is double-clicked during the development process. Inside the frame code we now enter:

```
float numberl1 = float.Parse(textBoxNumber1.Text);
float number2 = float.Parse(textBoxNumberl2.Text);
float result = numberl1 + number2;
textBoxResult.Text = result.ToString();
```

First the values of the text box (the type being a character string) are read out and converted into a number. This takes care of the Method 'float.Parse'. The addition follows next and finally the result is passed to the text box named 'textBoxResult'. Described more accurately, the text box then displays the value of the result if the value 'result.ToString' is

Activating interfaces

Electronics people may be wondering how they would control the computer interfaces their hardware is connected to. Particularly when a fundamental security principle of .NET (for desktop computers) states that interfaces cannot be activated directly by the application program.

C#- or VB.NET applications tend to access the corresponding Windows driver and in most cases this must first be installed. With off-the-shelf hardware (including most test gear) this is generally supplied by the manufacturer.

Programming a driver tends to be hardware-specific and under Windows C++ would be a good choice. For reasons of space we cannot go deeper into driver programming here. However, here is a small example of how an existing driver is called up. In this case it's for a parallel port interface, which is used in many hook-ups on account of its simplicity.

This sample application is by Levent Saltuklaroglu [8] and uses the well-known driver inpout32.dll, which can be downloaded for free [9]:

```
using System;
using System.Runtime.InteropServices;

public class PortAccess
{
   [DllImport("inpout32.dll", EntryPoint="Out32")]
   public static extern void Output(int address, int value);
}
```

The Using directive incorporates important Base Class libraries. Access to the driver is encapsulated in a user-defined Class named PortAccess. This is where the 'Output' Method is defined with two parameters (parallel port address – for example decimal 888 for LPT1 together with the value to be output). The parameters within the Class are passed to the Out32routine of inpout32.dll, which finally places a byte at the parallel port.

After implementing the Class it is easy to export data from the C# application program, for example using:

```
PortAccess.Output(888, 255);
```

Using the serial interface is even easier. Versions 2.0 onwards of .NET Framework provide the Class System.Io.Ports.Serialport, which allows you to address the serial interface, without even involving a .dll file. Information on this is in the online documentation [10].

delivered to its Property 'Text'. As you can see, we can alter the information panel's Properties both during the development process (as we have just seen) and from inside the program. These are indicated in the code following the name of the information panel with a dot and the name of the Property, for example, (textBoxResult.Text).

So that the result of the test box (a floating-point number of the type Float) can be displayed, it must first be converted into a character string. To do this in the code we need to indicate the name of the Variable Result, followed by a dot and the name of the Method (ToString). To understand how to set the Properties and application of Methods of the Object 'textBoxResult' of the Class 'Textb' you do need some prior experience of object-oriented programming. Sorry, but it has to be like this, since object orientation is at the very heart of C# (and VB.NET). It's not only the top-level elements but all the variables that are objects!

The next steps…

After starting an application, the result should look like **Figure 4**. If you feel like it, you could carry on programming such as a pocket calculator handling all the main types of calculation, which shouldn't be too difficult to turn into reality.

If you are now taking a shine to C#, you should find plenty of technical literature in any decent bookshop, both at 'Dummies' level for newcomers and more advanced tomes for seasoned users. There are significantly more books on C# than for VB.NET (and classic VB, which VB.NET is increasingly supplanting). In developer periodicals and on the Internet, problem solutions in the .NET world are being written almost exclusively in C#.

As already mentioned, the .NET programmer doesn't have to do all the work himself but can serve him/herself in a gigantic .NET Class library (all Objects are arranged in 'Classes' that determine which type of Properties and Methods an Object has). There is also online documentation from Microsoft that you can use as a reference work at [6]. Assistance is also available in an offline version as part of Visual Studio (warning: this is extremely comprehensive!). For each Class, the help file shows all the elements (Methods, Properties, Events) (**Figure 5**). Frequently, a sample application is given as well, structured according to the constituent programming language. Since there are frequently several ways of calling the Methods of a Class (so-called Overloads), a glance at the documentation is often absolutely necessary. So, for example, the 'Show' Method of the Class 'MessageBox' for displaying an information window) includes a total of 20 (!) call conventions. These vary in the parameters by which the calls must be passed.

When you are inputting source code the 'IntelliSense' feature of the development environment (**Figure 6**) can also be extremely helpful. As an example, when you are entering the name of an Object in a specific Class, all the Properties and Methods are indicated interactively.

In the second part of this series we will make a quick inspection into object-oriented programming. A section on 2-D vector graphic programming provides a practical task. Source code for the examples can be found on the Elektor project page [7].

(080668-I)

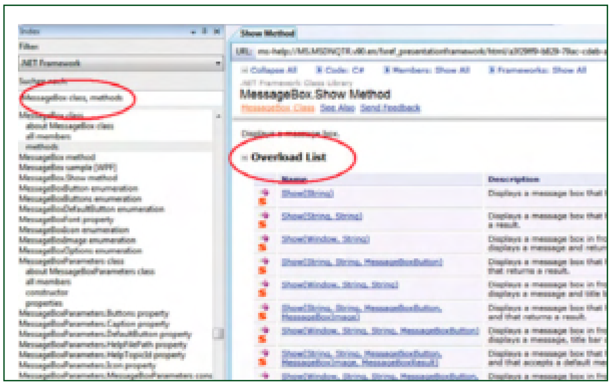


Figure 5. The Help system provides a comprehensive reference work for the Base Classes.

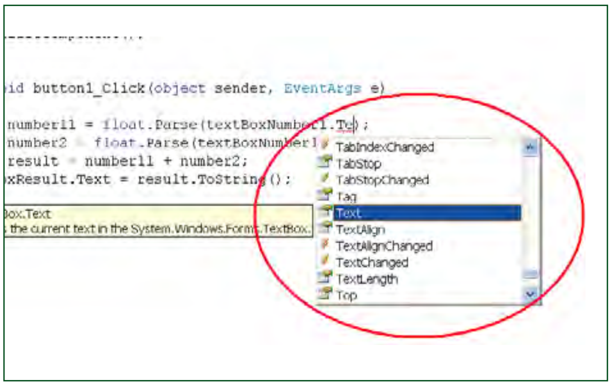


Figure 6. Interactive support for programmers with the help of 'IntelliSense'.

Internet Links and Further Reading

[1] www.mono-project.com/Main_Page

[2] http://dotnetframework.de/dotnet/produkte/sprachen.aspx (all onward links are in English)

[3] www.elektor.com/080450

[4] http://www.microsoft.com/express/product/default.aspx

[5] www.icsharpcode.net/OpenSource/SD

[6] http://msdn.microsoft.com/en-gb/vcsharp/default.aspx

[7] www.elektor.com/080668

[8] www.codeproject.com/KB/cs/csppleds.aspx

[9] http://logix4u.net/

[10] http://msdn.microsoft.com/en-gb/library/system.io.ports.serialport.aspx

See also Wikipedia (http://en.wikipedia.org/wiki/C_Sharp_(programming_language)) for other background on C#. Also the tutorial on programming in C# at http://www.java2s.com/Tutorial/CSharp/CatalogCSharp.htm

About the author

Veikko Krypczyk studied Business Information Technology. An enthusiastic programmer, he has been involved with C# and the .NET Framework for several years. His special interest is in application programming and the development of 2-D graphics applications (you can follow up his current project at http://easy-grafiker.de/cms/index.php and select the English menu). As an electronics hobbyist he cannot resist picking up his soldering iron now and again to try out interesting projects. You can e-mail Veikko at veikko2000@yahoo.de.

EMC DIRECTIVE

From 1 January 1996, home-made equipment must take into account emc Directive 89/336/eec (emc = ElectroMagnetic Compatibility). Basically, the directive states that no equipment may cause, or be susceptible to, external interference. Here, interference means many phenomena, such as electromagnetic fields, static discharge, mains pollution in the widest sense of the word.

Legislation
Home-made equipment may be taken into use only when it is certain that it complies with the directive. In the United Kingdom, the dti (Department of Trade and Industry) will, in general, only take action against offenders when a complaint has been made. If the equipment appears not to comply with the directive, the constructor may be sued for damages.

ce label
Home constructors need not affix a ce label to their equipment.

Elektor Electronics and the Directive

The publishers of Elektor Electronics intend that designs published in the magazine comply with the directive. Where necessary, additional guidelines will be given in the article. However, the publishers are neither obliged to do so, nor can they be held liable for any consequences if the constructed design does not comply with the directive. This column gives a number of measures that can be taken to ensure that EE-designed equipment complies with the directive. However, these are needed only in some designs. Other measures, particularly in case of audio equipment, are not new and have been applied for some time.

Why emc?
The important long-term benefit for the user is that all electrical and electronic equipment in a domestic, business and industrial environment can work harmoniously together.

Radiation
The best known form of emc is radiation that is emitted spuriously by an apparatus, either through its case or its cabling. Apart from limiting such radiation, the directive also requires that the apparatus does not impart spurious energy to the mains—not even in the low-frequency range.

Ferrite through-filters as illustrated are used for feeding cables through a panel.

Immunity
The requirements regarding immunity of an equipment to emc are new. Within certain limits of ambient interference, the apparatus must be able to continue working faultlessly. The requirements are fairly extensive and extend to a wide range of possible sources of interference.

Computers
Computers form the prime group for application of the directive. They, and microprocessors, are notorious sources of interfering radiation. Moreover, owing to the way in which their internal instructions are carried out sequentially, they are also very sensitive to interference. The notorious crash is but one manifestation of this.

Enclosures
A home-made computer system can comply with the emc directive only if it is housed in a metal enclosure. A minimum requirement is that the underside and rear of the enclosure is an l-shaped frame. All cabling must converge on this area or be filtered. If there are connectors on the front panel, a u-shaped metal frame should be used.

Even better results are obtained if a 20 mm wide, 1 mm thick copper strip is fixed along the whole width of the rear wall with screws at 50 mm intervals. The strip should have solder tags at regular distances for use as earthing points.

A closed case is, of course, better than an l-shaped or u-shaped frame. It is important that all its seams are immune to radiation ingress.

Power supplies
In any mains power supply, account should be taken of incoming and outgoing interference. It is good practice to use a standard mains filter whose metal case is in direct contact electrical contact with the enclosure or metal frame. Such a filter is not easily built at home. It is advisable to buy one with integral mains entry, fuse holder and on/off switch. This also benefits electrical safety in general. Make sure that the primary of the filter is terminated into its characteristic impedance—normally a series network of a 50 Ω, 1 W resistor and a 10 nF, 250 V capacitor.

Mains transformers must be provided with rc-networks at the primary and secondary side. Bridge rectifiers must be filtered by rc-networks. The peak charging current into the reservoir capacitor must be limited by the internal resistance of the transformer or by additional series resistors. It is advisable to use a 250 V, 2 W varistor between the live and neutral mains lines. At the secondary side, it is sometimes necessary to use a transient suppressor, preferably following the reservoir capacitor.

If the supply is used with digital systems, a common-mode inductor in the secondary a.c. lines may prove beneficial for limiting radiation. For audio applications, an earth screen between primary and secondary is advisable. This screen must be linked via a short wire with the earthing strip.

The supply must be able to cope with a mains failure lasting four periods and with mains supply variations of +10% and −20%.

Peripheral equipment and earthing
All cables to and from peripheral apparatus, such as measurement sensors, control relays, must be fed through the metal wall of the enclosure or frame. The earth lines of such cables must be connected directly to the earthing strip at the inside of the enclosure or frame via a wire not longer than 50 mm. When plugs are used, the cable earth, if any, must be connected to the earth pin or the metal surround of the connector.

Basically, all non-screened signal lines must be provided with a filter consisting of not less than a 30 mm ferrite bead around the cable or bunch of wires. This bead may be outside the enclosure (for instance, around the pc-to-monitor cable).

Leads that may have a resistance of 150 Ω must be provided with a 150 Ω series resistor at the inside of the connector shell. If technically feasible, there should also be a capacitor from this point to earth. Commercial feed-through t-filters or π-filters may, of course, be used. In all other cases, screened cable must be used for connections within the enclosure. Symmetrical lines must consist of twisted screened cable and be earthed at both ends.

The earth plane of printed-circuit boards must be linked as firmly as feasible with the earthing strip, for instance, via a flexible flat metal strip or flatcable.

Electrostatic discharge (esd)
All parts of an equipment that can be touched from outside must preferably be made from insulating, antistatic material. All parts that can be touched and enter the enclosure, such as potentiometer and switch spindles, must be earthed securely. All inputs and outputs whose wires or connector pins can be touched must be provided with an earth shield, for instance, an earthed metal surround via which any electrostatic discharge are diverted. This is most conveniently done by the use of connectors with sunken pins, such as found in sub-d connectors, and a metal case.

Audio equipment
Immunity to radiation is the most important requirement of audio equipment. It is advisable to use screened cables throughout. This is not always possible in case of loudspeaker cables and these must, therefore, be filtered. For this purpose, there are special high-current t-filters or π-filters that do not affect bass reproduction. Such a filter must be fitted in each loudspeaker lead and mounted in the wall of a metal screening box placed around the loudspeaker connections.

Low-frequency magnetic fields
Screened cables in the enclosure do not provide screening against the low-frequency (< a few kHz) radiation of the mains transformer. Therefore, these cables must run as close as possible to the walls of the enclosure. Moreover, their braid should be linked at one end to the earthing strip. In extreme cases, the power supply should be fitted in a self-contained steel enclosure. Special transformers with a shading ring that reduce the stray field can lower the hum even further.

High-frequency fields
High-frequency fields must not be allowed to penetrate the metal enclosure. All external audio cables must be screened and the screening must be terminated outside the enclosure. This again necessitates the use of all-metal connectors. All cable braids must be linked to the earthing strip inside the enclosure.

Owing to the skin effect, it is important to choose an enclosure with a wall thickness ≥2 mm to ensure that internal and external fields are kept separate. Any holes must be either small (<20 mm) or be covered with a metal mesh.

Heat sinks
Heat sinks should preferably be inside the enclosure and be earthed at several points. Non-earthed heat sinks in switch-mode power supplies often create problems. If possible, place an earth screen between transistor and heat sink. Ventilation holes must be covered with metal mesh unless they are smaller than 20 mm. Ventilators should be fitted inside the case.

Standard mains filters built into a mains entry together with an on/off switch. The metal shell must be in firm contact with the enclosure.

Cables
Cables often function as transmit/receive aerials. This applies equally well to screened cables. The braid of a coaxial cable must be terminated into a suitable connector such that it makes contact along the whole circumference. The braid may be used as the return path to obtain r.f. magnetic screening. For a.f. magnetic screening it is better to use twisted-pair screened cables. In a ribbon (flat) cable, each signal wire should be flanked, if at all possible, by earthed wires. The cable should be screened along one surface or, preferably, all around. Cables that carry signals ≥10 kHz that are not filtered in the enclosure, must be provided with a ferrite bead functioning as a common-mode inductance.

Printed-circuit boards
Elektor Electronics printed-circuit boards are provided with coppered fixing holes that are connected to the earth of the circuit. This arrangement, in conjunction with metal spacers, ensures good contact between the board and the circuit earth. Where this is important, boards have a special earth plane that can be connected, where feasible, to the earthing strip via a flatcable. These boards normally have no other earthing points and their fixing holes are, therefore, not coppered.

(960006)

T-filters and π-filters ensure that interference cannot em te from, or enter, the equipment via signal lines. They are available in various current ratings and for various frequency ranges.

Mini VHF FM Receiver

Small in size, big on sound

Ton Giesberts (Elektor Labs), from an idea by Karl Odenthal (Germany)

These days tiny FM radios are often integrated in many portable devices such as mobile phones and MP3 players. But why not make a simple receiver ourselves? There are currently several nice ICs available that contain a (nearly) complete receiver.

For this project we cast our eye on the TDA7021T from ST-NXP Wireless (formerly a part of Philips Semiconductors), an IC that is well over 20 years old, but is still readily available. Good designs, apparently, have a much longer life than many modern ICs of which a new version seems to appear every few months.
This project is a complete receiver circuit with excellent receive and sound qualities. The only disadvantage of using this IC, from the DIY enthusiast's perspective, is the fact that it is only available in a 16-pin SMD package. To make the construction easier we designed a small printed circuit board for this project. And to ensure that the completed circuit would be quite compact, the remainder of the circuit is also built using SMD parts, most of them 0805 size. The dimensions of this tiny PCB are only 3.2 × 2.7 cm! The circuit contains no difficult coils, only the VCO requires an air-cored inductor with only 4 turns.

Inner workings

Figure 1 shows a block diagram of the internals of the TDA7021T. This is a (nearly) complete integrated receiver circuit which has been specifically designed for portable radios and the like, requiring only a minimum of external components. As a result the final dimensions of the radio can be kept very small.

The IC uses a frequency-locked-loop (FLL) system with an intermediate frequency of 76 kHz. The selectivity is obtained with the aid of active RC filters. The only 'calibration' adjustment in the circuit is the resonance frequency of the oscillator (for the tuning).
The RF signal enters at pin 12 and is amplified first, after which it is transformed down by the mixer and passes through two IF filters. It is subsequently limited in amplitude. The IF limiter also supplies a signal for the optional signal strength indicator (via pin 9). The limited FM signal then goes to the demodulator and the correlator which decides whether the signal is tuned in properly. The demodulated

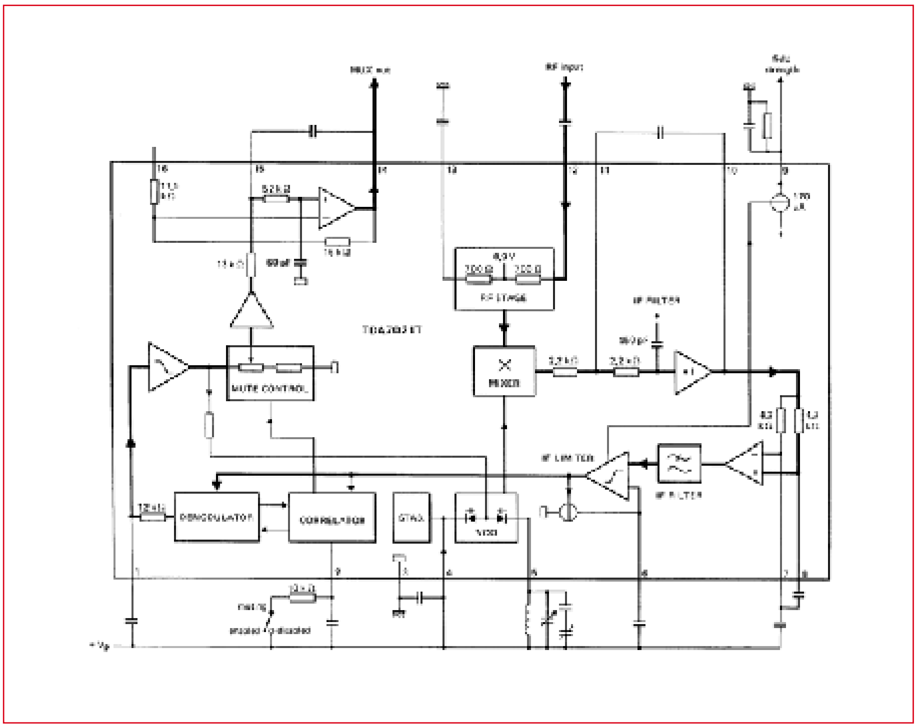


Figure 1. The internal design of the TDA7021T.

signal now passes through a loop-filter and a mute controller and then finally arrives at the audio output amplifier. The output of the loop filter also generates a control signal for the oscillator (VCO), which together with the external resonance network provides the tuning function.

For mono audio, the output amplifier delivers enough power (via a resistor) for a set of small earphones. You can, however, also connect a decoder to this output to generate a stereo signal. The output signal on pin 14 is not limited in frequency, that is, both the 19-kHz pilot signal as well as the L-R signal

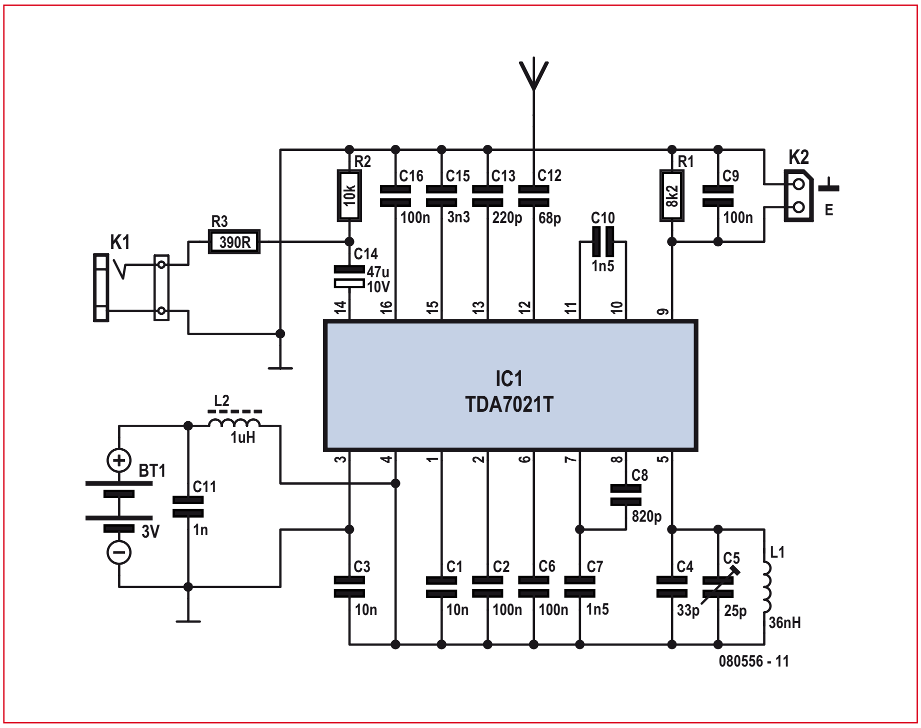


Figure 2. The complete receiver consists only of the IC and a few passive components.

are entirely available. This is also the reason that it is not recommended to connect the audio output directly to the line input of an audio system.

Circuit

The complete circuit for the FM receiver is shown in **Figure 2**. The design is virtually identical to the test circuit shown in the datasheet for the IC, because this is difficult to improve even a little without adding a lot of additional electronics. Now we only need a few resistors and capacitors plus a coil.

In **Figure 3** you can see the little PCB that we designed for this receiver. As mentioned earlier, SMD parts are used everywhere to keep the dimensions as small as possible. Soldering these small parts requires a bit of practice however. We have already explained how to deal with SMD components on several occasions in earlier issues of Elektor. For example, have a look at the articles 'SMDs? Don't panic' in the January and February 2003 issues.

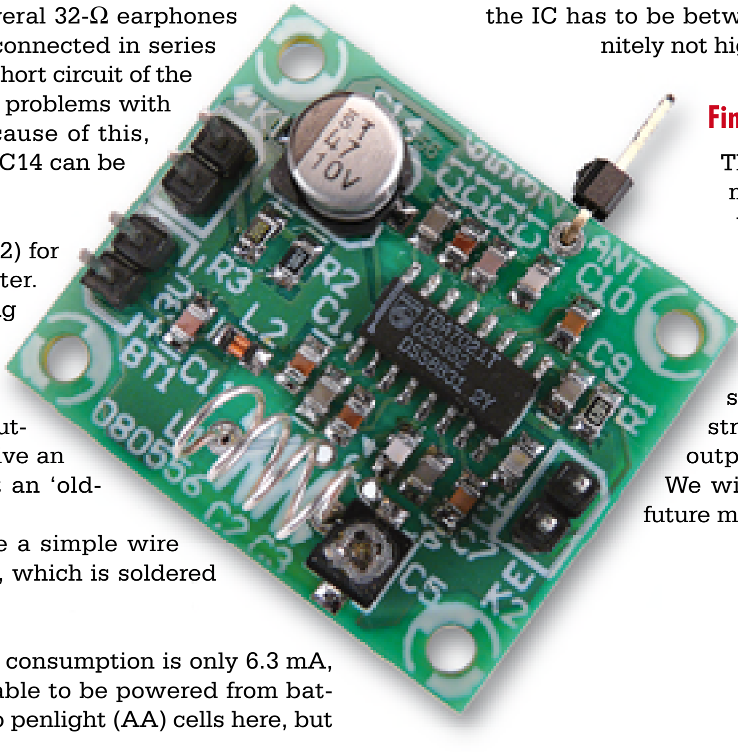
Practical matters

Moving on to a few practical instructions for the construction:

The inductor for the VCO is an air-cored coil with 4 turns. It is best if you use silver-plated wire for this. The easiest way is to wind 4 turns on a 4-mm drill and then stretch the coil slightly until the ends of the coil line up exactly with their respective mounting holes in the PCB. Place the coil a few millimetres above the board, so that you can easily adjust it slightly later on (by stretching or compressing) so that the tuning circuit correctly covers the entire VHF FM broadcast band from 88 to 108 MHz. Tuning is done with trimmer capacitor C5. This does however require a small screwdriver and a little patience, but you generally tend to listen to one and the same radio station anyway.

If you want, you can optionally (connected with wires as short as possible) replace the SMD trimmer capacitor with a proper tuning capacitor. Make sure you obtain the correct tuning range. This should be possible by changing the value of C4 and perhaps adding a series capacitor for C5.

All connections to the board are made with 2-way pin headers, to which you can connect your own choice of connectors. For the earphones you could use a stereo 3.5-mm jack socket, for example (with the L and R terminals connected in parallel). The output of the IC supplies sufficient power to drive several 32-Ω earphones directly. A 390-Ω resistor is connected in series with the output to prevent a short circuit of the output and prevent potential problems with longer screened cables. Because of this, resistor the output capacitor C14 can be kept quite small.



There is also a connection (K2) for a simple signal strength meter. Via resistor R1 and decoupling capacitor C9, pin 9 supplies a DC voltage which is a measure of the received signal strength. At 170 μA the output current is too small to drive an LED, but you could connect an 'old-fashioned' moving coil meter.
For the antenna you can use a simple wire antenna of about 75 cm long, which is soldered directly to the PCB.

Considering that the current consumption is only 6.3 mA, the circuit is eminently suitable to be powered from batteries. We decided to use two penlight (AA) cells here, but even using a lithium cell for the power supply will give nearly 20 hours of running time. It is also possible to use a mains adaptor. But note that the power supply voltage for the IC has to be between 1.8 and 6 V (most definitely not higher!).

Finally

This receiver is, to be sure, a mono implementation, but at the output (as already mentioned) the entire multiplexed signal (up to 53 kHz) is available. By using a PLL stereo decoder, such as the TDA7040T, a stereo signal can be generated in a straightforward way from the output signal of the TDA7021T. We will describe this circuit in a future mini project.

(080556-I)

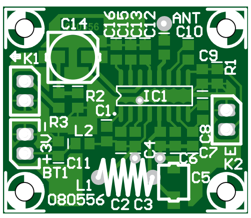


Figure 3. The receiver can be built on this tiny circuit board.

COMPONENT LIST

Resistors (SMD 0805)
R1 = 8kΩ2
R2 = 10kΩ
R3 = 390Ω

Capacitors (SMD 0805)
C1,C3 = 10nF
C2,C6,C9,C16 = 100nF
C4 = 33pF
C5 = 25pF trimmer (Murata type TZB4Z250AB10R00)
C7,C10 = 1nF5
C8 = 820pF
C11 = 1nF
C12 = 68pF
C13 = 220pF
C14 = 47μF 10V (Nichicon UWX1A470MCL1GB 5.5mmL chip type)
C15 = 3nF3

Inductors
L1 = 36nH (4 turns 0.5mm silver-plated wire, inside diameter 4mm; length 7mm)
L2 = 1μH, SMD case 0805 (fres > 300 MHz)

Semiconductors
IC1 = TDA7021T (SMD in SO16 case)

Miscellaneous
K1,K2 = 2-way pinheader
BT1 = 2-way pinheader + battery holder for 2-4 AA batteries

Opacity Measuremen

Jose Luis Basterra (Spain)

Opacity is a measure of impenetrability of objects or substances to radiation. Here, we are concerned with light, specifically within the spectrum visible to the human eye. In this month's instalment of the Modding & Tweaking series, we examine how the PC can do paper opacity measurements for us — all experimentally and on a budget of course!

The (simplified) relative sensitivity curve of the human eye shown in **Figure 1** tells us that he eye is more sensitive to green light than to yellow.

In order to be able to measure the visual opacity we ideally need a photodetector whose spectral sensitivity is similar to that of the human eye. A commonly available photodetector like the BPW34 has good sensitivity within a part of the spectrum perceived human eye, although its response curve is dissimilar, see **Figure 2**.

The response also depends on the type of light with which the object is illuminated. If we use a yellow LED we can look forward to an emission distribution roughly as in **Figure 3**.

The use of a yellow LED as the light source causes the detector to see the yellow part of the spectrum only, resulting in a better approach of the spectral response of the human eye. The comparison is illustrated by the graphs in **Figure 4** and **Figure 5**. Possibly, the use of mixed yellow and green light is even better although the results with yellow only are quite good.

The electronics

The opacity meter is based on frequency variation obtained from an oscillator that's controlled by the intensity of light detected by a BPW34 photodetector. **Figure 6** shows the schematic. The oscillator consists of a NAND gate from a 4093 package. Its frequency (10-100 Hz) is determined by C3 and R3, together with preset P1 and the BPW34 (D3) in the feedback circuit. The 'LCO' (light-controlled oscillator) output signal is digitized by three more gates from the same IC. The oscillator and the LEDs (one yellow, one red) are powered by a voltage regulator around a 7809. The red LED is an on/off indicator and helps to position the reader unit over the paper surface.

The circuit of the PC interface is shown in **Figure 7**. The interface with the computer is by means of the Centronics port and a series of optocouplers; four in the PC817 chip of which three are used. The connections between the circuit and the LPT1 ('parallel printer') port on the computer are as follows:

Circuit terminal	Pin no. on DB25
E1	1 (/STR)
E2	2 (D0)
E3	3 (D1)
GND	25 (GND)

The oscillator output signal enters the interface at J1. The circuit charges a 47 μF tantalum electrolytic capacitor (C1) to about 4.5 V, at which level zener diode D2 starts to pass current, illuminating the LED in optocoupler IC2. This causes a logic Low at E1, signalling to the program running on the computer to stop counting. The process comprises the following phases.

1. By means of a High level supplied to input E2, the PC effectively blocks the arrival of oscillator pulses.

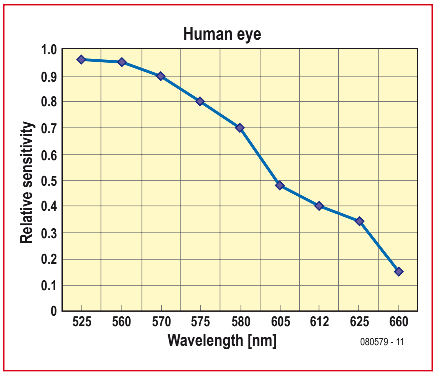


Figure 1. Plot human eye sensitivity as a function of light wavelength and you get a graph like this.

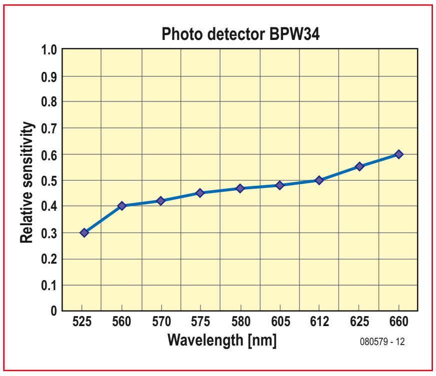


Figure 2. The response of the BPW34 within the spectrum seen by the eye is by no means flat.

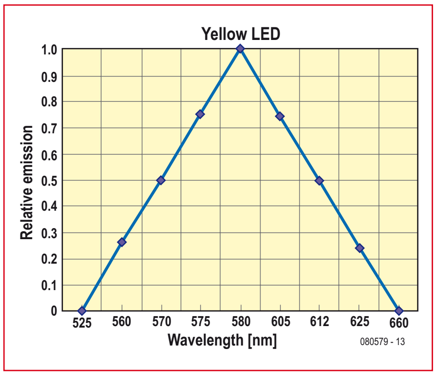


Figure 3. Heavily simplified spectral distribution of light emitted by a yellow LED.

t by PC

2. With C1 discharged, output E1 is held High i.e. deactivated.

3. Via E2, the software enables the arrival of oscillator pulses again.

4. The period it takes for the capacitor to charge to 4.5 V and pull E1 Low represents a time T, which is compared to a table entry in the program in order to determine the relative opacity value.

The value so obtained will be a reasonable approach of the actual opacity in the majority of cases. The table stored in the program is calibrated for values of opacity between 78 and 92%.

Software

This series of articles being free from rocket science, the very simplest form of Basic was used for the control program. True to the Modding & Tweaking tradition, all readers are invited to improve the program, come up with alternatives, perhaps develop code for Linux, C, 32-bit ARM platforms, what have you! Here, only the nitty-gritty is shown to serve as an incentive.
The BASIC program was written to measure, store and 'graph' instantaneous and average opacity values of paper sheets. You can download the program as file # **080579-11.zip** from the Elektor website.
The address &H378 is the base address of printer port LPT1 on the PC; &H37A is the control register at [base+2]. The first is used to send commands to the reader interface via LPT data lines D0 and D1, the second, for reading the capacitor charge status using the /STR (inverted strobe) line (which is bidirectional).

(080579-I)

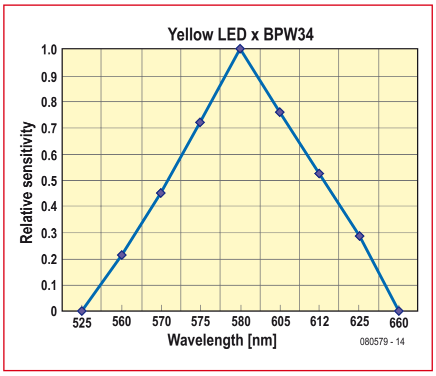


Figure 4. Response obtained from the combination: BPW34 + yellow LED.

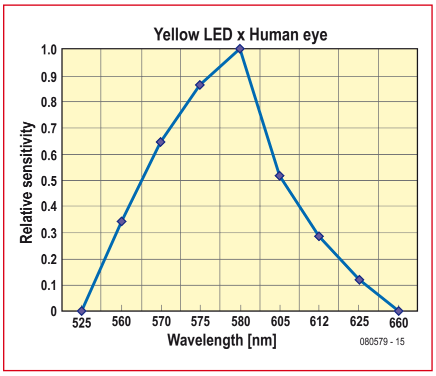


Figure 5. As Figure 4, but for the combination: BPW34 + human eye.

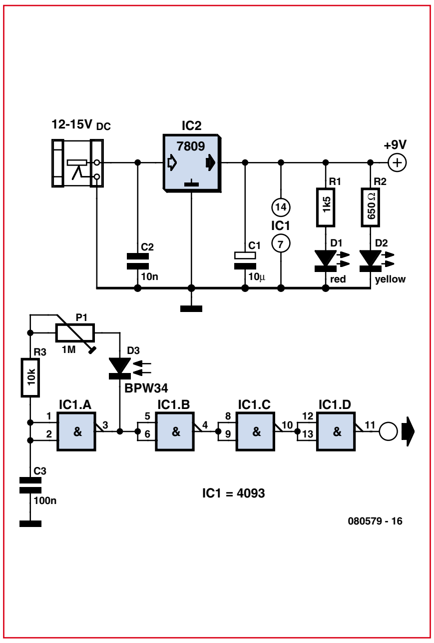


Figure 6. Schematic of the BPW34-controlled oscillator.

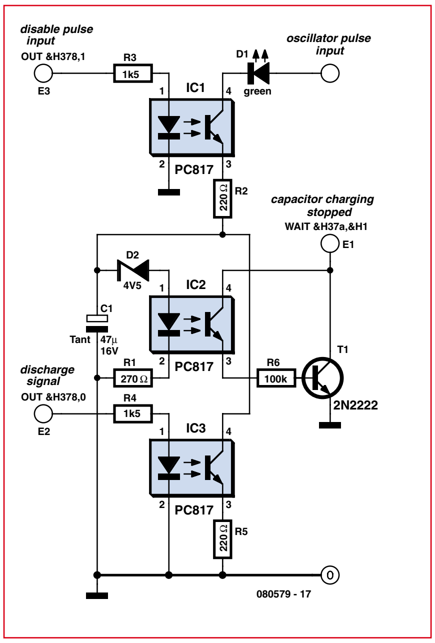


Figure 7. The PC interface consists mainly of optocouplers connected to Centronics port pins.

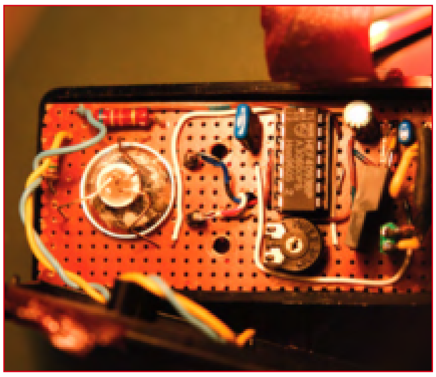


Figure 8. Internal view of the opacity reader.

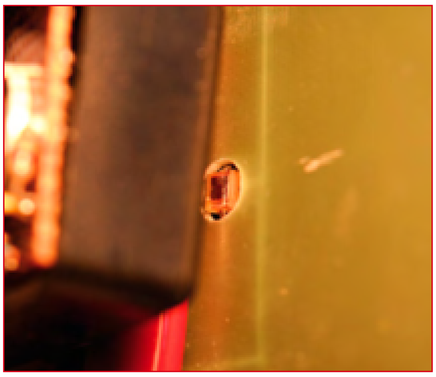


Figure 9. The face of the BPW34 photodetector protrudes from a small hole.

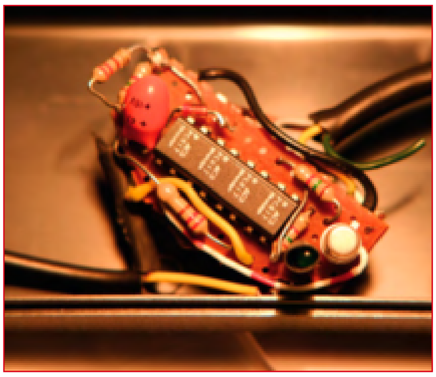
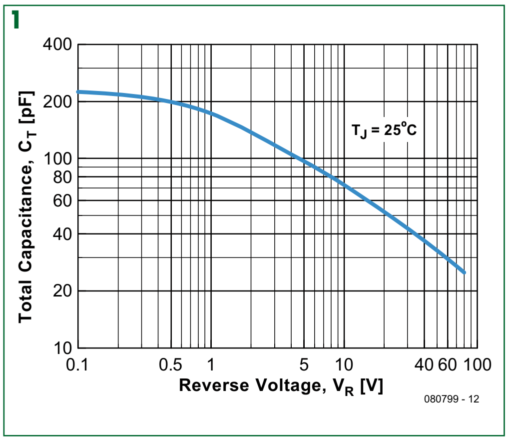


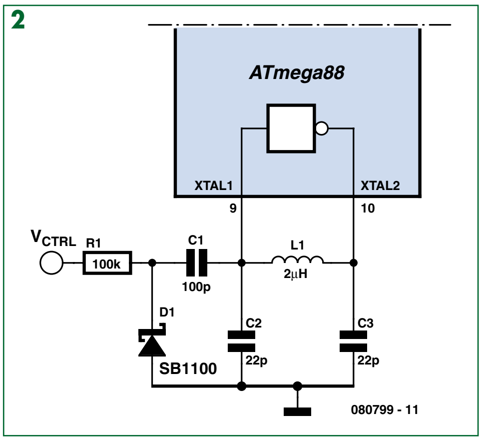
Figure 10. The PC817 quad optocoupler is by far the largest part in the PC interface.

No Varicap? Try a Schottky diode

Martin Ossmann (Germany)

Varactor diodes (Varicaps) along with many other exotic components which are often specified in HF and RF designs are sometimes difficult to obtain and in some cases may even have gone out of production. Not very encouraging if you had been planning to build a design that we featured in an earlier edition of Elektor. Anyone finding themselves in this situation may be surprised to learn that help may be on hand from an unexpected source…





Any general-purpose silicon diode has capacitive properties when reverse biased; the depletion region at the PN junction acts as a dielectric. Increasing the reverse bias widens the region, reducing the capacitance. Power engineers are aware that high voltage rectifiers can store high levels of charge at their depletion region. This property is not beneficial in rectification applications but using a little lateral thinking and with the motto "It's not a bug… it's a feature" in mind, we can maybe use this characteristic to substitute a diode for a varactor. The datasheet of an SB1100 Schottky power diode indicates that at 4 V reverse bias the diode has a "total capacitance" of 110 pF, and looking at the curve of this characteristic it is indeed proportional to voltage (see **Figure 1**).

To test the idea a simple LC oscillator was built (**Figure 2)** around the crystal oscillator inverter circuit integrated in an AtMega88 microcontroller. The application note AN456 Philips/NXP "Using LC oscillator circuits with Philips microcontrollers" [1] is a useful reference here. The finished voltage-controlled LC oscillator (VCLCO) actually works surprisingly well. The values of effective small signal capacitance (as measured on a capacitance meter) are given in the **Table** along with the oscillator output frequency.

The next time you are stumped for a varicap, instead of pulling your hair out it would be worthwhile taking a look through your collection of power diodes, you may well find something suitable there.

(080799-I)

Internet Link

[1] Philips/NXP Application note AN456:

www.nxp.com/acrobat_download/applicationnotes/AN456.pdf

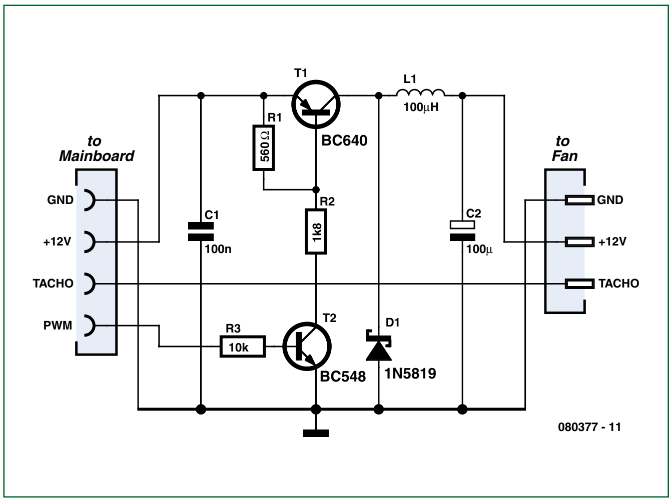
Table

V_{ctrl} [V]	C_{eff} [pF]	F [MHz]
0	125	19.49
1	82	19.70
2	64	20.00
3	54	20.19
4	48	20.34
5	44	20.46

Three-from-four fan control

Rainer Reusch (Germany)

Most PC-motherboards have multiple connectors for plugging in the cooling fans. These are increasingly frequently the four-way connectors. These connectors are mechanically and electrically compatible with the more common three-way connectors, which carry the power supply and tacho signal. On the fourth connection the motherboard supplies a pulsewidth modulated TTL signal, which is used to control the speed of the fan. A separate potentiometer or a temperature dependent speed controller to reduce the noise of the fan is now really history. But the availability of fans with a four-way plug is still rather limited and these fans are generally also more expensive. With the small, additional circuit that is proposed here, it nevertheless becomes possible to use these cheap and ubiquitous fans with their three-way connectors.



The transistor T1, the Schottky diode, the inductor and the capacitor C2 are, just like the 'old-fashioned' speed control circuit, connected in series with the fan. A reasonably clean power supply voltage for the three-wire fan is generated from this PWM signal — and with good efficiency, we may add. The tacho signal generated by the fan is routed directly back to the motherboard.

Building the circuit is not difficult. The capacitors must have a voltage rating of at least 16 V and the inductor has to be able to handle a minimum of 200 mA. Obtaining the right four-way plug with the necessary crimp contacts could be a bit of a problem however. As an alternative you could use the sockets that are designed to mate with the common pin-headers (2.54 mm pitch). If you decide to use these you will have to be careful and make sure you do not plug the connector in the wrong way around!

(080377-I)

Hexadoku

Puzzle with an electronics touch

This February edition of Hexadoku is again intended to try your patience and perseverance in solving a puzzle. Fortunately, looking at the vast number of correct solutions we get every month, lots of Elektor readers not only have a great time with Hexadoku, but also care to enter a prize draw for an E-blocks Starter Kit Professional and three Elektor Shop vouchers.

The instructions for this puzzle are straightforward.

In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4x4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

All correct entries received for each month's puzzle go into a draw for a main prize and three lesser prizes. All you need to do is send us the numbers in the grey boxes. The puzzle is also available as a **free download** from the Elektor website

SOLVE HEXADOKU AND WIN!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for an

E-blocks Starter Kit Professional

worth **£300**

and three **Elektor SHOP Vouchers** worth £40.00 each.

We believe these prizes should encourage

all our readers to participate!

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

PARTICIPATE!

Please send your solution (the numbers in the grey boxes) by email to:

hexadoku@elektor.com - Subject: hexadoku 02-2009 (please copy exactly).

Note: new email address as of this month!

Include with your solution: **full name and street address.**

Alternatively, by fax or post to: **Elektor Hexadoku**
PO Box 876 - Peterborough NH 03458-0876 USA
Fax 603-924-9467

The closing date is **1 March 2009.**

PRIZE WINNERS

The solution of the December 2008 Hexadoku is: **E5071**.

The **E-blocks Starter Kit Professional** goes to: Nicole Stahlschmidt (Germany).

An **Elektor SHOP voucher worth £40.00** goes to: Pau Sastre (ES), Guido Claes (B) and Helmut Balcke (F).

Congratulations everybody!

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

(c) PZZL.com

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

Elektor 'Digisplay' (1976-77)

Dirk Dral (The Netherlands)

While cleaning out the instrument cabinet in our company shop, I came across a Digisplay unit that I built and used many years ago with considerable pleasure.

At that time, I regularly put together circuits with gates, timers, counters and readouts, and it was a godsend to be able to tap 14 to 16 signals at once with an IC test clip and see the states of all these signals at the same time on a screen.

In May 1976 Elektor published a circuit for connection to an oscilloscope that allowed the values of 16 signals to be read on the screen in the form of 1's and 0's. There was apparently a lot of demand for this circuit, since Elektor decided to develop and sell a PCB design for it (# EPS 9376) in 1977.

At that time I was planning to build a new oscilloscope to replace my vintage 1965 kit 'scope, and I had already bought a DG 732 CRT for this purpose. Shortly afterward, Hameg came out with an attractively priced oscilloscope kit and I bought it right away, which meant I had a DG732 left over. It thus seemed like a good idea to build a Digisplay as a self-contained unit including the 'scope tube and fitted with only one control: an on/off switch. No need to make things complicated when I already had something to sink my teeth into!

The PCB was not especially difficult, since the circuit basically consisted of a 74150 TTL IC that converted the 16 input signals into a BCD code. A pair of gates did duty as an internal clock generator, and a few counters, decoders and D/A converters generated the X and Y deflection signals for the 'scope tube.

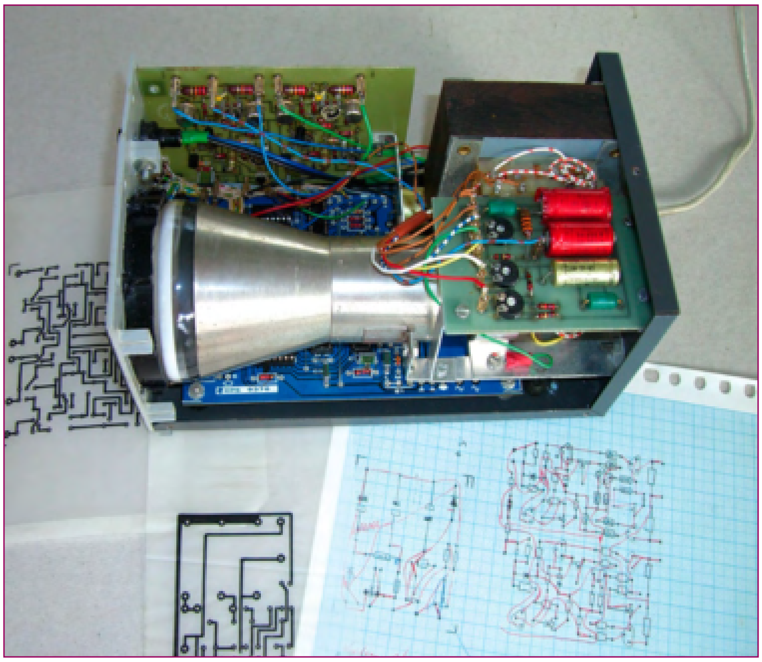
For the power supply of my simple oscilloscope, I used a design published in the December 1971 issue of Elektuur (Dutch edition). I designed a circuit board for this circuit with suitable trimpots for the adjustments.

The X and Y amplifiers were a different story, and after a bit of searching I decided to copy the amplifiers in the Hameg kit. I made PCB designs for both circuits using grid paper overlaid with tracing paper. The actual artwork was created using a few drawing pens. After this, everything was exposed on photosensitive PCB stock, etched and rinsed. Then it was just a matter of drilling the holes and fitting the components, and the boards were ready for use.

The power transformer that I used had only two secondary windings: a 250-V winding that provided 350 VDC after rectification, and a 6.3-V winding for the filament current of the 'scope tube and the power supply for the main circuit board.

During this process, I also managed to find a suitable enclosure: a Philips SQ4 case intended for audio equipment, which I had left over from some time earlier, was the right size to hold the tube, the power supply, and the circuit boards. I fitted a 22-way chassis-mount connector on the front panel for the 16 inputs, the four BCD outputs, and circuit ground.

The tube shield was fashioned from the liner of an old recessed spotlamp fixture, and the handle/stand was a cut-down carrying handle from an old transistor radio. That's recycling at its best, and you should remember that many electronics hobbyists work this way – they never throw anything away, since they can always find a use for it.



The exciting part was always trying out a new project: would everything work as you expected, or would it all go up in smoke?

After a few minor adjustments, which didn't involve any disasters, I could see the ones and zeros on the screen. A little miracle, which I could share with my family.

I was able to use the Digisplay several times in practice to help sort out problems with various circuits. However, I think that the greatest satisfaction lies in building something like this yourself, because you learn something new with each project and thus expand your knowledge.

Another moment of satisfaction, and certainly not the least one, comes when you plug it in after 30 years, switch it on, and it still works perfectly!

(080926-I)

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

ELEKTOR SHOWCASE

To book your showcase space contact Strategic Media Marketing Inc.

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

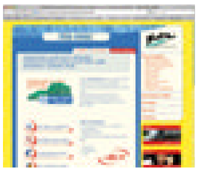
AVIT RESEARCH

www.avitresearch.co.uk

USB has never been so simple...
with our USB to Microcontroller Interface cable.
Appears just like a serial port to both PC and Microcontroller, for really easy USB connection to your projects, or replacement of existing RS232 interfaces.

See our webpage for more details. From £10.00.

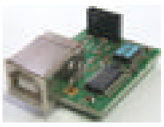
BETA LAYOUT

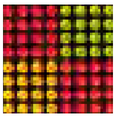


www.pcb-pool.com

Beta layout Ltd Award-winning site in both English and German offers prototype PCBs at a fraction of the cost of the usual manufacturer's prices.

BYVAC





www.byvac.com

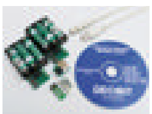
- USB to I2C
- Microcontrollers
- Forth
- Serial Devices

C S TECHNOLOGY LTD

www.cstech.co.uk

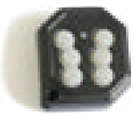
Low cost PIC prototyping kits, PCB's and components, DTMF decoder kits, CTCSS, FFSK, GPS/GSM, radio equipment and manuals.
PCB design and PIC program development.

DECIBIT CO.LTD.



www.decibit.com

- Development Kit 2.4 GHz
- Transceiver nRF24L01
- AVR MCU ATmega168



www.elektor.com

DESIGNER SYSTEMS

http://www.designersystems.co.uk

Professional product development services.

- Marine (Security, Tracking, Monitoring & control)
- Automotive (AV, Tracking, Gadget, Monitoring & control)
- Industrial (Safety systems, Monitoring over Ethernet)
- Telecoms (PSTN handsets, GSM/GPRS)
- AudioVisual ((HD)DVD accessories & controllers)

Tel: +44 (0)1872 223306

EASYDAQ

www.easydaq.biz

- USB powered, 4 relays + 4 DIO channels
- Will switch 240VAC @ 10 amps
- Screw terminal access
- LabVIEW, VB, VC
- Free shipping
- From £38

Design & supply of USB, USB Wireless, Ethernet & Serial, DAQ, Relay & DIO card products. info@easydaq.biz

EASYSYNC

http://www.easysync.co.uk

EasySync Ltd sells a wide range of single and multi-port USB to RS232/RS422 and RS485 converters at competitive prices.

ELNEC

www.elnec.com

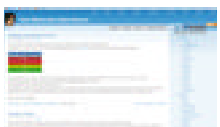
- device programmer manufacturer
- selling through contracted distributors all over the world
- universal and dedicated device programmers
- excellent support and after sale support
- free SW updates
- reliable HW
- once a months new SW release
- three years warranty for most programmers

YOUR ELECTRONICS OPEN SOURCE

http://dev.emcelettronica.com

Website full of Projects and Resources for Electronics Engineers and DIY.

- Tutorial
- Hardware (Schematic & Gerber)
- Firmware (Asm & C)
- Reference Design



Everyone can submit a story as a useful source!
'Share for life'

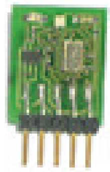
FIRST TECHNOLOGY TRANSFER LTD.

http://www.ftt.co.uk/PICProTrng.html

Microchip Professional C and Assembly Programming Courses.
The future is embedded.
Microchip Consultant / Training Partner developed courses:

- Distance learning / instructor led
- Assembly / C-Programming of PIC16, PIC18, PIC24, dsPIC microcontrollers
- Foundation / Intermediate

FLEXIPANEL LTD



www.flexipanel.com

TEAclippers - the smallest PIC programmers in the world, from £20 each:

- Per-copy firmware sales
- Firmware programming & archiving
- In-the-field firmware updates
- Protection from design theft by subcontractors

FUTURE TECHNOLOGY DEVICES

http://www.ftdichip.com

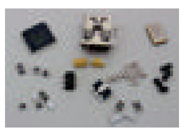
FTDI designs and sells USB-UART and USB-FIFO interface i.c.'s.
Complete with PC drivers, these devices simplify the task of designing or upgrading peripherals to USB

LONDON ELECTRONICS COLLEGE

http://www.lec.org.uk

Vocational training and education for national qualifications in Electronics Engineering and Information Technology (BTEC First National, Higher National NVQs, GCSEs and GCEs). Also Technical Management and Languages.

LCDMOD KIT



http://www.lcdmodkit.com

Worldwide On-line retailer

- Electronics components
- SMT chip components
- USB interface LCD
- Kits & Accessories
- PC modding parts
- LCD modules

products and services directory

MQP ELECTRONICS

www.mqp.com

- Low cost USB Bus Analysers
- High, Full or Low speed captures
- Graphical analysis and filtering
- Automatic speed detection
- Bus powered from high speed PC
- Capture buttons and feature connector
- Optional analysis classes

OBD2CABLES.COM

http://www.obd2cables.com

- Thousands of OBD cables and connectors in stock
- Custom cable design and manufacturing
- OBD breakout boxes and simulators
- Guaranteed lowest prices
- Single quantity orders OK
- Convenient online ordering
- Fast shipping

Visit our website, or email us at:
sales@obd2cables.com

ROBOT ELECTRONICS

http://www.robot-electronics.co.uk

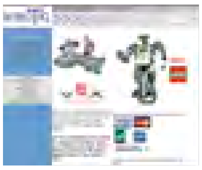
Advanced Sensors and Electronics for Robotics

- Ultrasonic Range Finders
- Compass modules
- Infra-Red Thermal sensors
- Motor Controllers
- Vision Systems
- Wireless Telemetry Links
- Embedded Controllers

ROBOTIQ

http://www.robotiq.co.uk

Build your own Robot!
Fun for the whole family!



- MeccanoTM Compatible
- Computer Control
- Radio Control
- Tank Treads
- Hydraulics

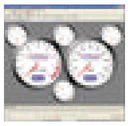
Internet Technical Bookshop,
1-3 Fairlands House, North Street, Carshalton,
Surrey SM5 2HW
email: sales@robotiq.co.uk Tel: 020 8669 0769

SCANTOOL.NET

http://www.scantool.net

ScanTool.net offers a complete line of PC-based scan tools for under £50.

- 1 year unconditional warranty
- 90 day money back guarantee
- For use with EOBD compliant vehicles
- Fast shipping
- Compatible with a wide range of diagnostic software



Visit our website, or email us at:
sales@scantool.net

www.elektor.com

USB INSTRUMENTS

http://www.usb-instruments.com

USB Instruments specialises in PC based instrumentation products and software such as Oscilloscopes, Data Loggers, Logic Analaysers which interface to your PC via USB.

VIRTINS TECHNOLOGY

www.virtins.com

PC and Pocket PC based virtual instrument such as sound card real time oscilloscope, spectrum analyzer, signal generator, multimeter, sound meter, distortion analyzer, LCR meter. Free to download and try.

CANDO - CAN BUS ANALYSER

http://www.cananalyser.co.uk

- USB to CAN bus interface
- USB powered
- FREE CAN bus analyser S
- Receive, transmit & log. CAN messages
- ISO11898 & CAN 2.0a/2.0b compliant
- Rugged IP67 version available

netronics

SHOWCASE YOUR COMPANY HERE

Elektor Electronics has a feature to help customers promote their business, Showcase - a permanent feature of the magazine where you will be able to showcase your products and services.

- For just $385.00 + VAT ($35.00 per issue for eleven issues) Elektor will publish your company name, website address and a 30-word description
- For $550.00 + VAT for the year ($50.00 per issue for eleven issues) we will publish the above plus run a 3cm deep full colour image - e.g. a product shot, a screen shot from your site, a company logo - your choice

Places are limited and spaces will go on a strictly first come, first served basis. So-please fax back your order today!

I wish to promote my company, please book my space:
• Text insertion only for $385.00 + VAT   • Text and photo for $550.00 + VAT

NAME: .... ORGANISATION: ....

JOB TITLE: ....

ADDRESS: ....

....

.... TEL: ....

PLEASE COMPLETE COUPON BELOW AND FAX BACK TO 1-978-281-7706

COMPANY NAME ....

WEB ADDRESS ....

30-WORD DESCRIPTION ....

....

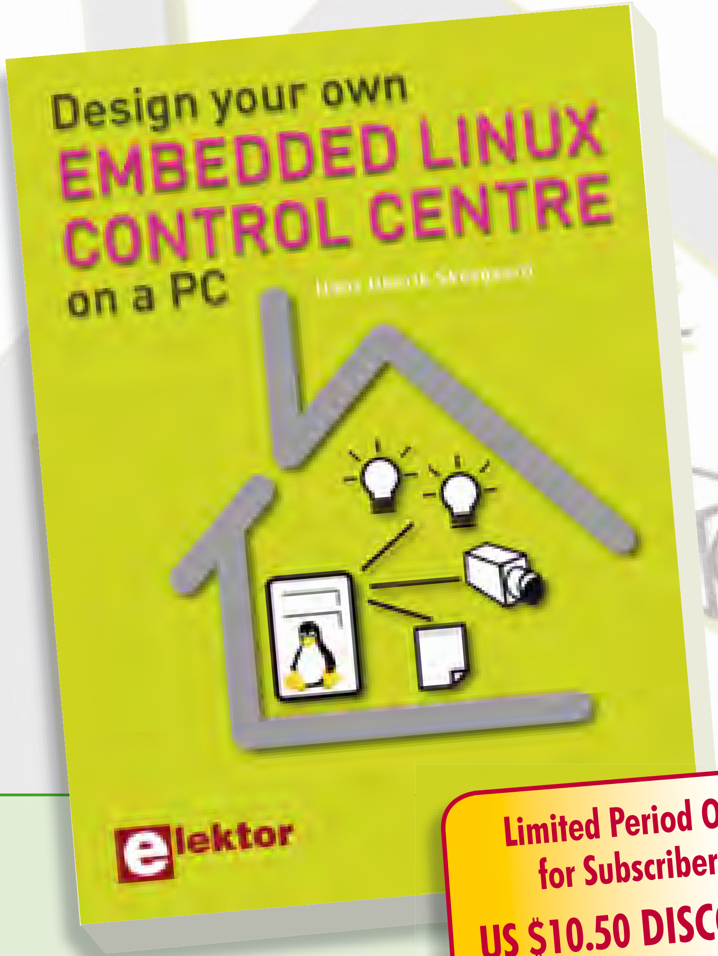
....

....

....

Going Strong

A world of electronics from a single shop!



Limited Period Offer for Subscribers!

US $10.50 DISCOUNT

www.elektor.com/usa/february

Books

A DIY system made from recycled components

Design your own
Embedded Linux control centre on a PC

This book covers a do-it-your-self system made from recycled components. The main system described in this book reuses an old PC, a wireless mains outlet with three switches and one controller, and a USB webcam. All this is linked together by Linux. This book will serve up the basics of setting up a Linux environment – including a software development environment – so it can be used as a control centre. The book will also guide you through the necessary setup and configuration of a webserver, which will be the interface to your very own home control centre. All software needed will be available for downloading from the Elektor website.

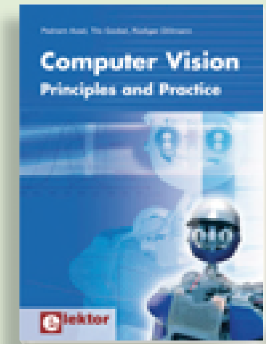
234 pages • ISBN 978-0-905705-72-9 • US $45.30

5.0, 6.0, VBA, .NET, 2005

Visual Basic for Electronics Engineering Applications

This book is targeted towards those people that want to control existing or self-built hardware from their computer. After familiarizing yourself with Visual Basic, its development environment and the toolset it offers are discussed in detail. Each topic is accompanied by clear, ready to run code, and where necessary, schematics are provided that will get your projects up to speed in no time.

476 pages • ISBN 978-0-905705-68-2
US $56.50



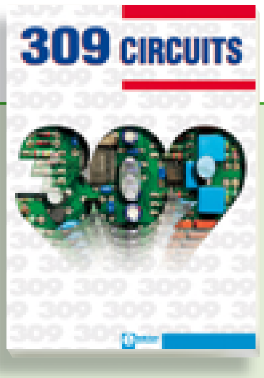
Principles and Practice

Computer Vision

Computer vision is probably the most exciting branch of image processing, and the number of applications in robotics, automation technology and quality control is constantly increasing. Unfortunately entering this research area is, as yet, not simple. Those who are interested must first go through a lot of books, publications and software libraries. With this book, however, the first step is easy. The theoretically founded content is understandable and is supplemented by many examples.

320 pages • ISBN 978-0-905705-71-2
US $60.40

Prices and item descriptions subject to change. E. & O.E

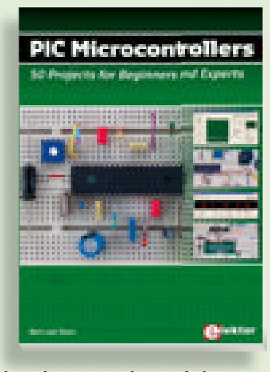


Fully elaborated electronics projects

309 Circuits

The present tenth edition of the popular '30x Circuits' series of books once again contains a comprehensive variety of circuits, sub-circuits, tips and tricks and design ideas for electronics. Among many other inspiring topics, the following categories are well presented in this book: test & measurement; RF (radio); computers and peripherals; audio & video; hobby and modelling; microcontrollers; home & garden; power supplies & battery chargers; etcetera.

432 pages • ISBN 978-0-905705-69-9
US $39.50

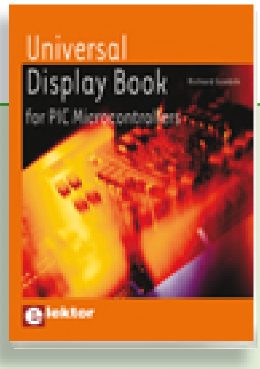


Silent alarm, poetry box, night buzzer and more

PIC Microcontrollers

This hands-on book covers a series of exciting and fun projects with PIC microcontrollers. You can built more than 50 projects for your own use. The clear explanations, schematics, and pictures of each project on a breadboard make this a fun activity. You can also use it as a study guide. The technical background information in each project explains why the project is set up the way it is, including the use of datasheets. Even after you've built all the projects it will still be a valuable reference guide to keep next to your PC.

446 pages • ISBN 978-0-905705-70-5
US $52.70



From LED to graphical LCD

Universal Display Book
for PIC Microcontrollers

This book begins with simple programs to flash LEDs, and eventually by stages to use other display indicators such as the 7-segment and alphanumeric liquid crystal displays. As the reader progresses through the book, bigger and upgraded PIC chips are introduced, with full circuit diagrams and source code, both in assembler and C. A tutorial is included using the MPLAB programming environment, together with the PCB design package and EAGLE schematic to enable readers to create their own designs.

192 pages • ISBN 978-0-905705-73-6
US $43.40

More information on the Elektor Website:

www.elektor.com

Elektor US
PO Box 876
Peterborough NH 03458-0876
USA

Toll-free: 888-924-9465
Phone: 603-924-9464
Fax: 603-924-9467
E-mail: custservus@elektor.com

elektor SHOP

Bestseller!

Modern technology for everyone

FPGA Course

FPGAs have established a firm position in the modern electronics designer's toolkit. Until recently, these 'super components' were practically reserved for specialists in high-tech companies. The nine lessons on this courseware CD-ROM are a step by step guide to the world of Field Programmable Gate Array technology. Subjects covered include not just digital logic and bus systems but also building an FPGA webserver, a 4-channel multimeter and a USB controller. The CD also contains PCB layout files in pdf format, a Quartus manual, project software and various supplementary instructions.

ISBN 978-90-5381-225-9 • US $27.40

Software Tools & Hardware Tips

Ethernet Toolbox

This CD-ROM contains all essential information regarding Ethernet interfaces! Ethernet Toolbox includes a collection of datasheets for dedicated Ethernet interface ICs from many different manufacturers. It provides a wealth of information about connectors and components for the physical layer (PHY) and specific software tools for use with the Ethernet (Software). To help you learn about the Ethernet interfaces, we have compiled a collection of all articles on this topic that have appeared in Elektor and complemented them with additional documentation and links to introductory articles on Ethernet interfaces. The documents are PDF files.

ISBN 978-90-5381-214-3 • US $36.80

CD-ROMs

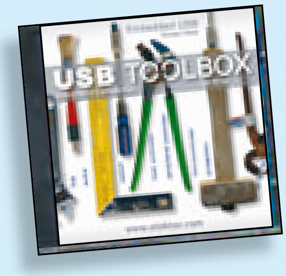
CD-ROMs

More than 68,000 components

ECD 4

The program package consists of eight databanks covering ICs, germanium and silicon transistors, FETs, diodes, thyristors, triacs and optocouplers. A further eleven applications cover the calculation of, for example, LED series droppers, zener diode series resistors, voltage regulators and AMVs. A colour band decoder is included for determining resistor and inductor values. ECD 4 gives instant access to data on more than 68,000 components. All databank applications are fully interactive, allowing the user to add, edit and complete component data. This CD is a must-have for all electronics enthusiasts.

ISBN 978-90-5381-159-7 • US $33.00



Embedded USB Know How

USB Toolbox

This CD-ROM contains all the essential information a designer needs to start working with the USB interface. It includes a large collection of data sheets for specific USB components from a wide range of manufacturers. USB Toolbox provides information on all ICs suitable for different applications. A sub-division has been made in controllers, hubs, microcontrollers and others. What is perhaps more interesting for many designers however, is the extremely extensive software collection which contains drivers, tools and components for Windows, Delphi and various microcontroller families. Of course, none of the Elektor articles on the subject of USB are missing on this CD-ROM.

ISBN 978-90-5381-212-9 • US $37.60

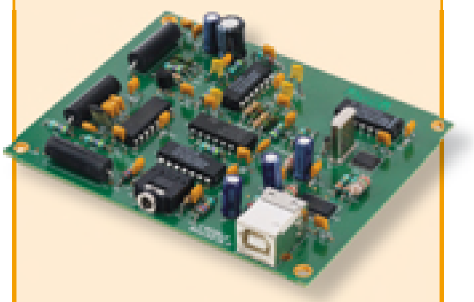
Bestseller!

Remote control by Mobile Phone

Remote control using mobile phones and SMS (Text Messaging) is in great demand but many systems on sale suffer from imperfections. This ingenious new design combines powerful capabilities with low technical overheads. It has programmable AC mains switching outlets plus status reports by text message and alarm-activated delivery of GPS data. Remote control by mobile was never easier, cheaper or more reliable!

Kit of parts, incl. PCB, programmed controller and all parts

Art. # 080324-71 • US $101.80



Software Defined Radio

SD radio receivers use a bare minimum of hardware, relying instead on their software capabilities. The Elektor SDR project (by Burkhard Kainka) demonstrates what's achievable, in this case a multi-purpose receiver covering all bands from 150 kHz to 30 MHz. It's been optimised for receiving DRM and AM broadcasts but is also suitable for listening in to the world of amateur transmissions. The designer's aim for this project was to create a receiver displaying high linearity and phase accuracy. Development was focussed on the characteristics that were most important for a top-notch DRM receiver and the end result is a receiver with remarkable interference rejection characteristics!

Ready-populated and tested board

Art. # 070039-91 • US $140.50

Stand Alone OBD-2 Analyser

Since the turn of the millennium more and more new car models have been fitted with the latest version of the on board diagnostic interface OBD-2. This handy analyser makes a simple job of rummaging through the information stored by the client-accessible part of your car's computer. It works with all current OBD-2 protocols and can read and erase trouble codes stored in the vehicle and reset the MIL display. All this without the help of a PC or a visit to a service station.

PCB, ready-assembled and tested (case and OBD-2 cable included)

Art. # 070038-93 • US $117.90

Elektor SMT Oven

The Elektor SMT reflow oven will faithfully handle most if not all your soldering of projects using surface mount devices (SMDs). The oven is particularly suited for use not just in Colleges, workshops, clubs and R&D laboratories, but also by the advanced electronics enthusiast. This precious workbench tool is at home where SMD boards have to be produced to a variety of requirements on size, components and soldering materials.

Size: 418 x 372 x 250 mm (16.5 x 14.6 x 10 inch)

Art. # 080663-91 • US $1665.00 (Excl. VAT)

Kits & Modules

Prices and item descriptions subject to change. E. & O.E

Product Shortlist

February 2009 (No. 386)	US $
Model Coach Lighting Decoder	
080689-1 ...... PCB, long (l = 230 mm)	10.95
080689-2 ...... PCB, medium (l = 190mm)	10.95
080689-3 ...... PCB, short (l = 110mm)	8.95
080689-41 .... PIC12F683, programmed	9.50
Transistor Curve Tracer	
080068-1 ...... Main PCB	42.00
080068-91 .... PCB, populated and tested	see www.elektor.com
January 2009 (No. 385)	
Radio for Microcontrollers	
071125-71 .... 868 MHz module	13.60
ATM18 on the Air	
071125-71 .... 868 MHz module	13.60
Meeting Cost Timer	
080396-41 .... ATmega168, programmed	16.10
Capacitive Sensing and the Water Cooler	
080875-91 .... Touch Sensing Buttons Evaluation kit	51.90
080875-92 .... Touch Sensing Slider Evaluation kit	51.90
Three-Dimensional Light Source	
080355-1 ...... Printed circuit board	47.00
Moving up to 32 Bit	
080632-91 .... ECRM40 module	60.40
December 2008 (No. 384)	
PLDM	
071129-1 ...... Printed circuit board	11.00
Hi-fi Wireless Headset	
080647-1 ...... Printed circuit board: Transmitter	14.90
080647-2 ...... Printed circuit board: Receiver	14.90
LED Top with Special Effects	
080678-71 .... Kit of parts incl. SMD-stuffed PCB and programmed controller	73.60
November 2008 (No. 383)	
Motorised Volume Pot	
071135-41 .... Programmed controller ATMEGA8-16PU	11.20
Speed Camera Warning Device	
080615-1 ...... Printed circuit board	29.30
080615-41 .... Programmed controller PIC16F876A-I/SO	22.30
Remote Control by Mobile Phone	
080324-1 ...... Printed circuit board	33.70
080324-41 .... Programmed controller ATMEGA8-16PU	11.20
080324-71 .... Kit of parts	101.80
Tracking Hot Spots	
080358-1 ...... Printed circuit board	17.30
ATmega meets Vinculum	
071152-91 .... VDIP1 module	42.50
October 2008 (No. 382)	
Communicating with CAN	
071120-71 .... PCB, partly populated	103.50
Elektor SMT Precision Reflow Oven	
080663-91 .... Ready to use oven (230VAC only)	1665.00
Multi-purpose GPS Receiver	
070309-41 .... Programmed controller PIC18F2520	21.90
ATM18 Relay Board and Port Expander	
071035-72 .... Relay PCB with all components and relays	69.60
071035-95 .... Port Extension PCB, populated with SMD	25.30
RF Sweep Frequency Generator / Spectrum Analyser	
040360-41 .... Programmed controller ATmega8535	41.10
September 2008 (No. 381)	
DCC Command Station	
070989-71 .... Kit of parts incl. programmed ARM module	166.90

Bestsellers

Books	1	Embedded Linux Control Centre	ISBN 978-0-905705-72-9	US $45.30
	2	Universal Display Book	ISBN 978-0-905705-73-6	US $43.40
	3	PIC Microcontrollers	ISBN 978-0-905705-70-5	US $52.70
	4	309 Circuits	ISBN 978-0-905705-69-9	US $39.50
	5	Computer Vision	ISBN 978-0-905705-71-2	US $60.40
CD-ROMs	1	FPGA Course	ISBN 978-90-5381-225-9	US $27.40
	2	ECD 4	ISBN 978-90-5381-159-7	US $33.00
	3	Ethernet Toolbox	ISBN 978-90-5381-214-3	US $36.80
	4	USB Toolbox	ISBN 978-90-5381-212-9	US $37.60
	5	Home Automation	ISBN 978-90-5381-195-5	US $26.20
Kits & Modules	1	Remote control by Mobile Phone	Art. # 080324-71	US $101.80
	2	Software Defined Radio	Art. # 070039-91	US $140.50
	3	Stand Alone OBD2 Analyser	Art. # 070038-93	US $117.90
	4	Elektor SMT-oven	Art. # 080663-91	US $1665.00
	5	Communicating with CAN	Art. # 071120-71	US $103.50

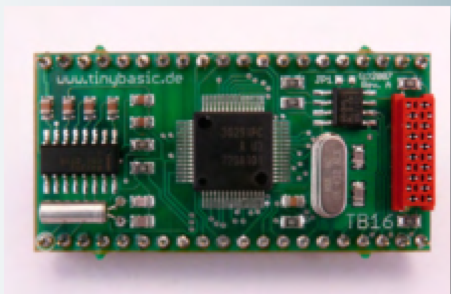
Order quickly and securely through www.elektor.com/usa/shop or use the Order Form near the end of the magazine!

elektor SHOP

Elektor US
PO Box 876
Peterborough NH 03458-0876
USA
Toll-free: 888-924-9465
Phone: 603-924-9464
Fax: 603-924-9467
E-mail: custservus@elektor.com

4 to 16 — Port Extension for Micros

Ever so often a microcontroller simply doesn't have enough ports to control all functions in your circuit. Our clever port extension boards provide the solution, requiring no more than 16 (SPI) control lines. The first board sports 16 digital outputs; more may be obtained by 'cascading' boards. The second, equally simple to drive board gives access to an HD44780 compatible LCD. The hardware is completed by free software examples for AVR controllers.



TinyBrick M16C Module

The M16C display board published in May 2008 got a good response and is now complemented by a small M16 module with a DIL footprint called 'TinyBrick'. Like the display, TinyBrick is programmable in TinyBasic. For development purposes we also present a carrier board, and there's more: as an example of a real life application, we describe a home alarm system capable of sending SMS Text messages.

From R8C to R32C

The article series we published in 2006 on the Elektor 'Tom Thumb' R8C-base miniature microcontroller board was a resounding success and we now take the next step by presenting R32C, a Renesas micro with 32 bits architecture, floating point and 50 MHz clock.

As with the R8C13 projects, the hardware consists of a carrier board containing the microcontroller, and an application board with various interfaces and a prototyping area. Brand new, we reckon, is the OLED display control developed for this system.

Article titles and magazine contents subject to change, please check 'Magazine' on www.elektor-usa.com

w.elektor-usa.com www.elektor-usa.com www.elektor-usa.com www.elektor-usa.

Elektor on the web

All magazine articles back to volume 2000 are available online in pdf format. The article summary and parts list (if applicable) can be instantly viewed to help you positively identify an article. Article related items are also shown, including software downloads, circuit boards, programmed ICs and corrections and updates if applicable. Complete magazine issues may also be downloaded.

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

Also on the Elektor website:

- Electronics news and Elektor announcements
- Readers Forum
- PCB, software and e-magazine downloads
- Surveys and polls
- FAQ, Author Guidelines and Contact

Order Form 02-2009

Description	Price each	Qty.	Total	Order Code
Microcontroller Systems Engineering NEW	US $54.70			
CapSense Buttons Kit (080875-91) NEW	US $51.90			
CapSense Sliders kit (080875-92) NEW	US $51.90			
Universal Display Book for PIC Microcontrollers	US $43.40			
Design your own Embedded Linux Control Centre on a PC	US $45.30			
Starter Kit Professional for ARM	US $305.40			
	Sub-total			
	Shipping & Handling US$ *(Airmail from Europe)*		20.00	
	Total paid			

Prices and item descriptions subject to change. The publishers reserve the right to change prices without prior notification. Prices and item descriptions shown here supersede those in previous issues. E. & O.E.

Name/Company

Street & No.

City/State/Zip *Country*

Tel *E-mail*

EL02 *Date* – – *Signature*

METHOD OF PAYMENT
(Please see next page for ordering information)

- ☐ Check or Money Order
- ☐ Mastercard
- ☐ VISA
- ☐ Discover
- ☐ American Express

Expiry date: ______________________

Verification code: __________________

Send order form to:

Elektor US
PO Box 876
Peterborough NH 03458-0876
USA

Toll-free: 888-924-9465
Phone: 603-924-9464
Fax: 603-924-9467
E-mail: custservus@elektor.com
www.elektor.com

elektor electronics worldwide

Subscription Order Form 02-2009

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

(11 issues including summer double issue)

I pay only US $39.95!*

* *Offer available in US & Canada only.*
Canada please add $11.00 per year for postage.

METHOD OF PAYMENT
(Please see next page for ordering information)

- ☐ Check or Money Order
- ☐ Mastercard
- ☐ VISA
- ☐ Discover
- ☐ American Express

Expiry date: ______________________

Verification code: __________________

Name/Company

Street & No.

City/State/Zip *Country*

Tel *E-mail*

EL02 *Date* – – *Signature*

Send order form to:

Elektor US
PO Box 876
Peterborough NH 03458-0876
USA

Toll-free: 888-924-9465
Phone: 603-924-9464
Fax: 603-924-9467
E-mail: custservus@elektor.com
www.elektor.com

ORDERING INFORMATION

To order contact customer service:

Toll-free: 1-888-924-9465
On-line at www.elektor.com/shop
Fax: 603-924-9467
Mail: Elektor US
PO Box 876
Peterborough NH 03458-0876 USA
E-mail: custservus@elektor.com

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

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

COMPONENTS

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

PAYMENT

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

TERMS OF BUSINESS

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

SUBSCRIPTIONS (US & CANADA ONLY)

1Yr. $39.95 (Canada add $11 per year for postage)

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

Elektor US
PO Box 876
Peterborough NH 03458-0876 USA

E-mail: custservus@elektor.com

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

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

For gift subscriptions, please include gift recipient's name and address as well as your own, with remittance. A gift card will be sent on request.

Subscriptions may be cancelled at any time for a refund of all unmailed issues.

Index of Advertisers

Advertising space for the issue of 16 March 2009
may be reserved not later than 17 February 2009
with Strategic Media Marketing, Inc. – 2 Main Street – Gloucester, MA 01930 – USA – Telephone 1.978.281.7708 – Fax 1.978.281.7706 – e-mail: ElektorUSA@smmarketing.us to whom all correspondence, copy instructions and artwork should be addressed.