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

elektor

electronics worldwide

R32C/111 & Relatives

upgrading to a 32-bit micro was never easier

T-Reg

a high-voltage regulator for tube amps

M16C TinyBrick

programmable in BASIC

Hyper Flea Mart

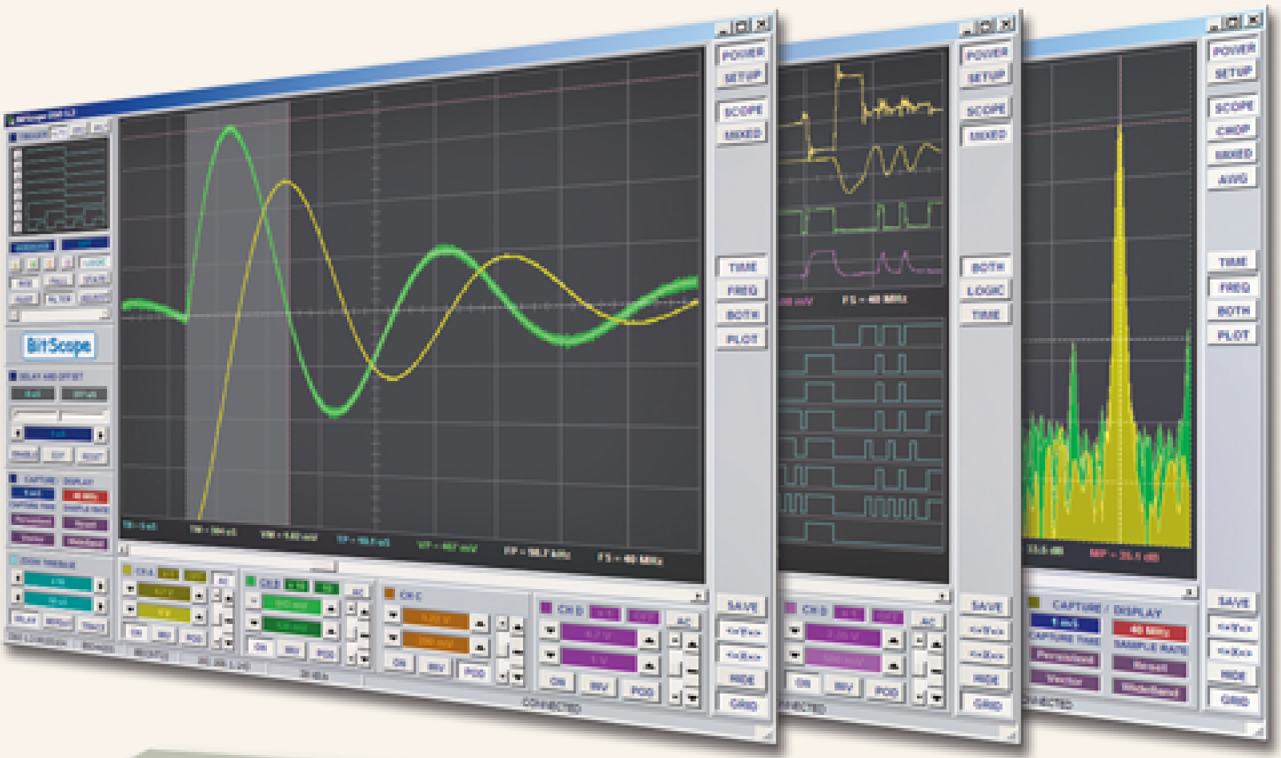
sourcing in China

Sinewaves

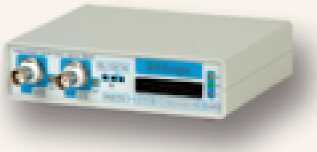
in programmable logic

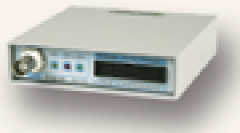
BitScope PC Oscilloscopes & Analyzers

DSO Test Instrument Software for BitScope Mixed Signal Oscilloscopes



4 Channel BitScope


2 Channel BitScope


Pocket Analyzer

Digital Storage Oscilloscope

✓ Up to 4 analog channels using industry standard probes or POD connected analog inputs.

Mixed Signal Oscilloscope

✓ Capture and display up to 4 analog and 8 logic channels with sophisticated cross-triggers.

Spectrum Analyzer

✓ Integrated real-time spectrum analyzer for each analog channel with concurrent waveform display.

Logic Analyzer

✓ 8 logic, External Trigger and special purpose inputs to capture digital signals down to 25nS.

Data Recorder

✓ Record anything DSO can capture. Supports live data replay and display export.

Networking

✓ Flexible network connectivity supporting multi-scope operation, remote monitoring and data acquisition.

Data Export

✓ Export data with DSO using portable CSV files or use libraries to build custom BitScope solutions.

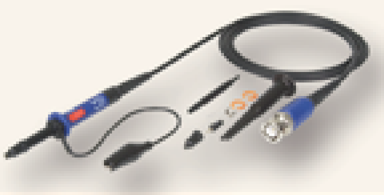
BitScope DSO Software for Windows and Linux

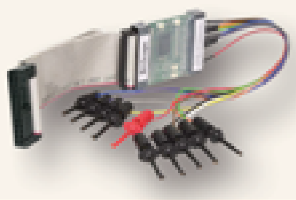
BitScope DSO is fast and intuitive multi-channel test and measurement software for your PC or notebook. Whether it's a digital scope, spectrum analyzer, mixed signal scope, logic analyzer, waveform generator or data recorder, BitScope DSO supports them all.

Capture deep buffer one-shots or display waveforms live just like an analog scope. Comprehensive test instrument integration means you can view the same data in different ways simultaneously at the click of a button.

DSO may even be used stand-alone to share data with colleagues, students or customers. Waveforms may be exported as portable image files or live captures replayed on other PCs as if a BitScope was locally connected.

BitScope DSO supports all current BitScope models, auto-configures when it connects and can manage multiple BitScopes concurrently. No manual setup is normally required. Data export is available for use with third party software tools and BitScope's networked data acquisition capabilities are fully supported.





www.bitscope.com

logic

Tube & Micro on the front cover

Giel and Mart, two illustrious, long standing colleagues ruling Elektor's graphics department have a special awareness for electronics stuff that catches the eye. For the purpose of virtually all photographic imagery you can admire in Elektor every month, they pull electronics from the hands of lab workers and the occasional editor to get illustrations ready in time. From time to time, their cameras get to see what they consider highly photogenic stuff. All relative of course — after all, this is electronics! So, high grades every month for shiny things and the weird paraphernalia I've a habit of hauling up the stairs of Elektor House for the purpose of my *Retronics* instalments.

When the time comes to design yet another Elektor front cover, Giel and Mart will not hesitate to pull photos on the canvas of items not easily thought of as belonging together in terms of technology. Like a Renesas R32C/111 microcontroller and a Raytheon JAN6528 tube. Personally, I have no preference — the micro I consider a black box apparently designed to spawn error messages until I get the programming right; the tube, an impressive looking device with far fewer pins, more tolerant to my errors, and with a much shorter datasheet.

If a specialist magazine claims to cover electronics there is no way the publisher or editor can avoid considering the delicate balance described as the amount of coverage given to microcontrollers (and all things embedded) on the one hand, and analogue, traditional technology on the other. Strong predilections and prejudice should be avoided though, as well as false reasoning like "no computer stuff on a sailboat". *Marine Weather Data with the SDR* on page 44 proves the point.

Although this March 2009 edition was announced in Elektor's publishing plan as a 'microcontroller issue' a long time ago, and it cannot be denied that these wonderful devices rock the electronics boat, we've done our best to respect the balance by including articles that are hard to associate with bits, bytes and number crunching: *T-Reg*, a high-tension regulated power supply, *Hyper Flea Mart* on component sourcing in China, *Design Tips* and *Retronics*. A pity there's so little space on the front cover.

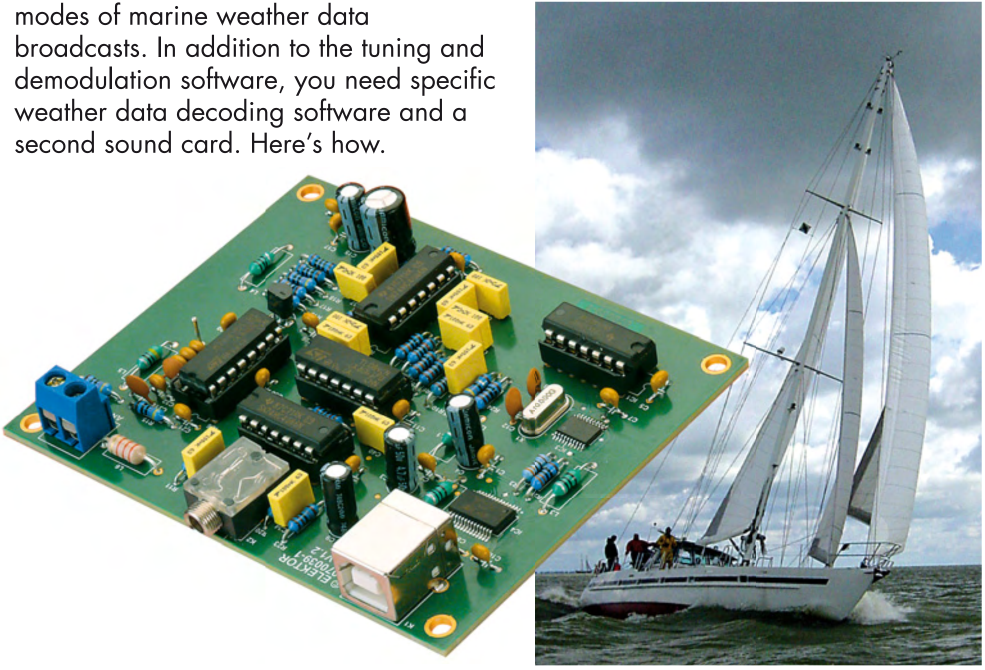
Jan Buiting
Editor

R32C and 18

Come see us at
Embedded World 2009, Nürnberg Germany, March 3–5, 2009.
Embedded Systems Conference Silicon Valley, San Jose USA, March 30–April 3, 2009

44 Marine Weather Data with the SDR

The Elektor SDR receiver requires some enhancements to the basic design to accommodate the special operating modes of marine weather data broadcasts. In addition to the tuning and demodulation software, you need specific weather data decoding software and a second sound card. Here's how.



CONTENTS

Volume 1
March 2009
no. 3

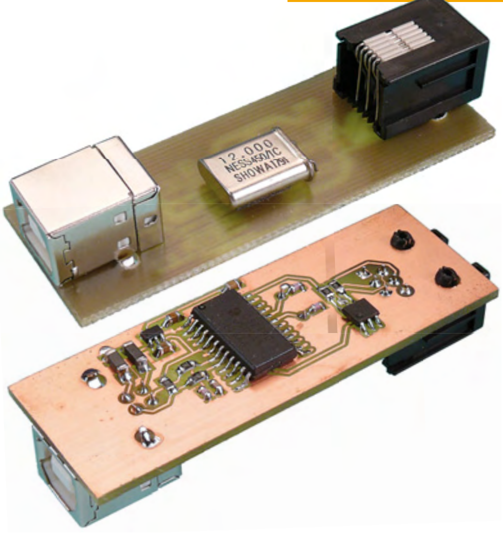
its Relatives

Elektor's famous 'Tom Thumb' R8C has just gained a talented sibling. The R32C/111, his mathematically gifted big brother, is blessed with 32-bit architecture, Floating Point Unit arithmetic and a 50 MHz clock rate. The R8C and the R32C are both from Renesas' M16C family, making the upgrade to the more versatile 32-bit system extremely simple.

22 T-Reg

Here we describe a novel design for a high-voltage regulator that is very simple yet offers excellent performance. The output voltage is fully adjustable with a single resistor without any changes in performance. You can use several types of tubes or MOSFETs for the pass device in the circuit.

54 USB-I2C Bridge



Although heritage parallel or serial ports see declining use in practice, to designers, a portless PC is a nuisance. To solve at least a part of the problem, we present an interface for the I2C bus for connection to an USB connection on your PC. It works a treat when used in the LabVIEW environment.

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.

MARCH 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-usa.com

Elektor USA magazine 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 MA (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.

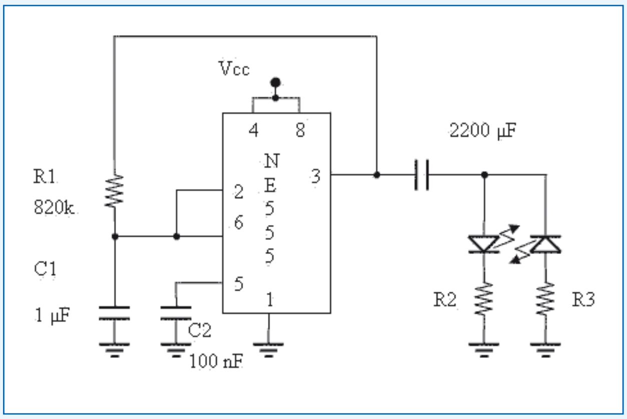
Disclaimer
Prices and descriptions of publication-related items subject to change.
Errors and omissions excluded.

Printed in the USA

LED Blinker and Rigol scopes

Dear Editor — I have two comments I would like to make on published articles.

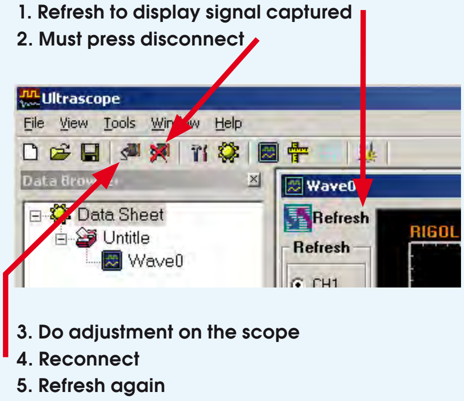
'LED Blinker' by Thomas Scarborough represents an excellent idea. One can use a 555 if only two LEDs are used. Some 555s can also drive up to 200 mA on pin 3. The circuit shown here can be used.



Resistors R2 and R3 can be left out, probably with the effect of significantly reducing the life expectancy of the LEDs. Well in most cases this will not be a problem because the hobby planes do not fly that often and losing the LED would not have any significant effect except with extremely high light outputs. R2 and R3 can be combined depending on the LED's used. I have also tested it with one white and one red LED.

Regarding the cheap scopes on review in the October 2007 issue I share some of the frustrations of the author. The illustration shows an idea to use the Rigol scopes on the USB port without disconnecting the USB cable.

1. Refresh to display signal captured
2. Must press disconnect



3. Do adjustment on the scope
4. Reconnect
5. Refresh again

My scope is the Rigol DS 5202CA (200 MHz 1 GS/s), not the fancy model with the digital analyzer but I had to buy it for further studies. In South Africa, even these Rigol scopes are very expensive for private use. This scope was nearly worth my month's full salary, I had to dig into my savings to buy it.

Paul Badenhorst (South Africa)

Help with Delphi code (2)

Dear Editor — I just received the January 2009 issue and I saw Ceri Clatworthy's message about HID Delphi code in 'Mailbox'.
I am also electronic engineer and programmer working with C++ Builder, the 'brother' of Delphi in RAD development. For HID access handling I am using 'JvHidDeviceController from Project JEDI and it served me well since now even though I didn't get too far with it.

By the way, I'm a contributing author to the C++ Builder Developers Journal (www.bcbjournal.org).
Please forward my contact info to your correspondent if more help is needed.
I've been reading Elektor since it was first published here in Greece and continue to be a subscriber.
George Tokas (Greece)

Many thanks for responding George and we'll happily put you through to Ceri.

Elektor US on newsstand distribution

Hello Editor — your introduction to the USA seems to be working. I was recently on holiday in San Diego and as a subscriber to Elektor's European edition I was pleased to see the US version of Elektor on the newsstand in the Westfield shopping mall in UTC (University Town Centre). Hopefully this will allow some new customers to browse the magazine and decide to buy.
I hope you are successful in introducing the people of the USA to the great magazine that is Elektor.
Richard Jones (UK)

Thanks for letting us know Richard, our US publisher is working hard to gradually expand the number of US and Canadian bookstores and newsstands stocking copies of Elektor USA. US readers, please let us know your findings and requirements.

Using the PLDM with a step-up converter

Hi Jan — I built this circuit (PLDM Power LED Driver Module, December 2008, *Ed.*) provisionally using point-to-point wiring, and it works perfectly. My question is: can this design also be used with a step-up converter? This option may be more attractive, since it's probably more common for the input voltage to be too low.
Hubert Eichler (Germany)

In principle, the circuit can also work with a step-up converter, including the types that use a 2.5-V reference voltage (assuming that you also use a 2.5-V reference IC). However, you must ensure that the output voltage is always higher than the input voltage. With a 12-V supply voltage, this means always having at least five white LEDs connected in series across the output, and so on. Another thing to bear in mind is that the efficiency degrades with decreasing input voltage for two reasons:

1. The forward voltages of the diodes become more significant.

2. The DC resistance of the coil and other losses take a larger toll due to the higher current levels.

For these reasons, in my opinion a step-down version is preferable. It's always possible to choose an appropriate transformer voltage.
Dr Thomas Scherer

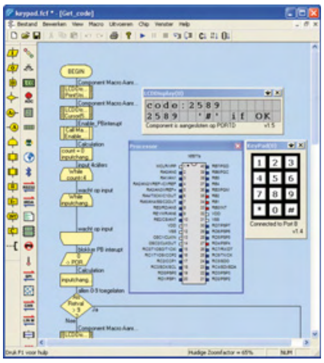
Using a keypad with PIC Flowcode

Dear Elektor people — if you are looking for an example of how to use a keypad with Flowcode, you won't find anything useful among the examples on the CD or on the Matrix Multimedia website. The only suggestion you will find for using the getkeypadnumber macro is to set PB0.2 high and wait until a 'portB change' interrupt occurs. This may work in the simulator, but it doesn't work in practice.

My solution for this is:

- Set PB0.2 high and configure PB4.7 as an input, as otherwise it cannot detect an interrupt.

- Provide a small wait time between configuring the port and enabling the interrupt, as otherwise an interrupt will be triggered immediately.



- After the interrupt, PB0.2 must be set low again, as otherwise the interrupt will continue to be triggered as long as a key is pressed.

- After PB0.2 is set high again, an interrupt will also be generated when a key is released, so you should be prepared to handle this.

The Flowcode file I've developed shows an example of entering a four-digit code. I hope that other enthusiasts will find this information useful.
Ivan Rigaux (Belgium)

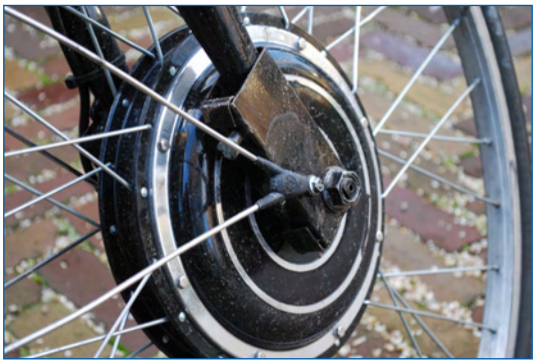
We're sure that a large number of Flowcode users will find this suggestion very handy. A portion of the program is shown here, and the complete .fcf file (no. 081048) is available on the Elektor website for free download (look under the Mailbox heading for the March 2009 issue).

Electric bicycle

Dear Jan — I read the 'e-bike' article in the November 2008 issue with some interest. I also undertook a similar project early last year, and since then I have ridden approximately 1500 km with considerable pleasure.

In the article, you mentioned that you used an old mountain bike. I would like to point out that many bikes nowadays are fitted with suspension front forks. They make a robust impression and even appear to be stronger than 'old-fashioned' metal forks. However, the fork tubes are made from aluminum alloy and are simply not suitable for use with an electric drive. The aluminum wheel mounts will quickly become cracked or break due to the enormous torque generated by the wheel. This can lead to very hazardous situations. My own bike has an aluminum suspension fork, and I made a simple metal reinforcement for it that converts it into a solid, safe assembly (see photo).
I do not agree with your statement that 'the bike behaves like normal in all other respects'. The maneuverability is distinctly reduced by the considerable extra weight of the front wheel. You get used to this fairly quickly, but it is certainly not the same as a normal bike. Incidentally, there are suppliers who have no trouble providing various wheel sizes, including 27 inch (for example). A rear-wheel version is also available.
It might be a good idea to publish one or more follow-up articles on topics such as selecting a battery, where you could examine aspects such as weight, dimensions, recharging time, price, and so on.
Based on my experience, the controller cuts out with annoying abruptness when the battery gets low, especially when the bike is accelerating. An auxiliary circuit that limits the 'throttle lever' in order to prevent this sort of cut-out would be a desirable addition.
André Cramer (The Netherlands)

The author of the article, Thijs Beckers, replies:
Thanks kindly for your message. Your remarks about the wheel mounts are certainly worth mentioning. We also thought it would be a good ideal to print a photo of your handiwork here.
What I meant with 'the bike behaves like normal' is that it still 'operates' the same way – in other words, you don't have to start anything or switch anything on or off when you want to go for a ride. Naturally, you are absolutely right that the handling of the bike is different due to the different wheel and the increased weight. Perhaps I should have worded this more clearly in the article.
In addition, the fact that wheels of this sort are also available in other sizes should have been pointed out. A German colleague ordered the wheel for us, and he was not sure which diameter was most common, so he just picked one.
To be honest, I doubt that there will be a follow-up article on choosing a battery. This sort of information would best be included directly with the article, but this was not possible due to lack of space. In addition, writing such an article for a specific application, such as an electric bicycle, would probably be regarded as too specialized. It would be better to write a general article on the various aspects of batteries and their uses.
However, an article on a circuit that limits the throttle might be interesting. If you feel the call, let us know!



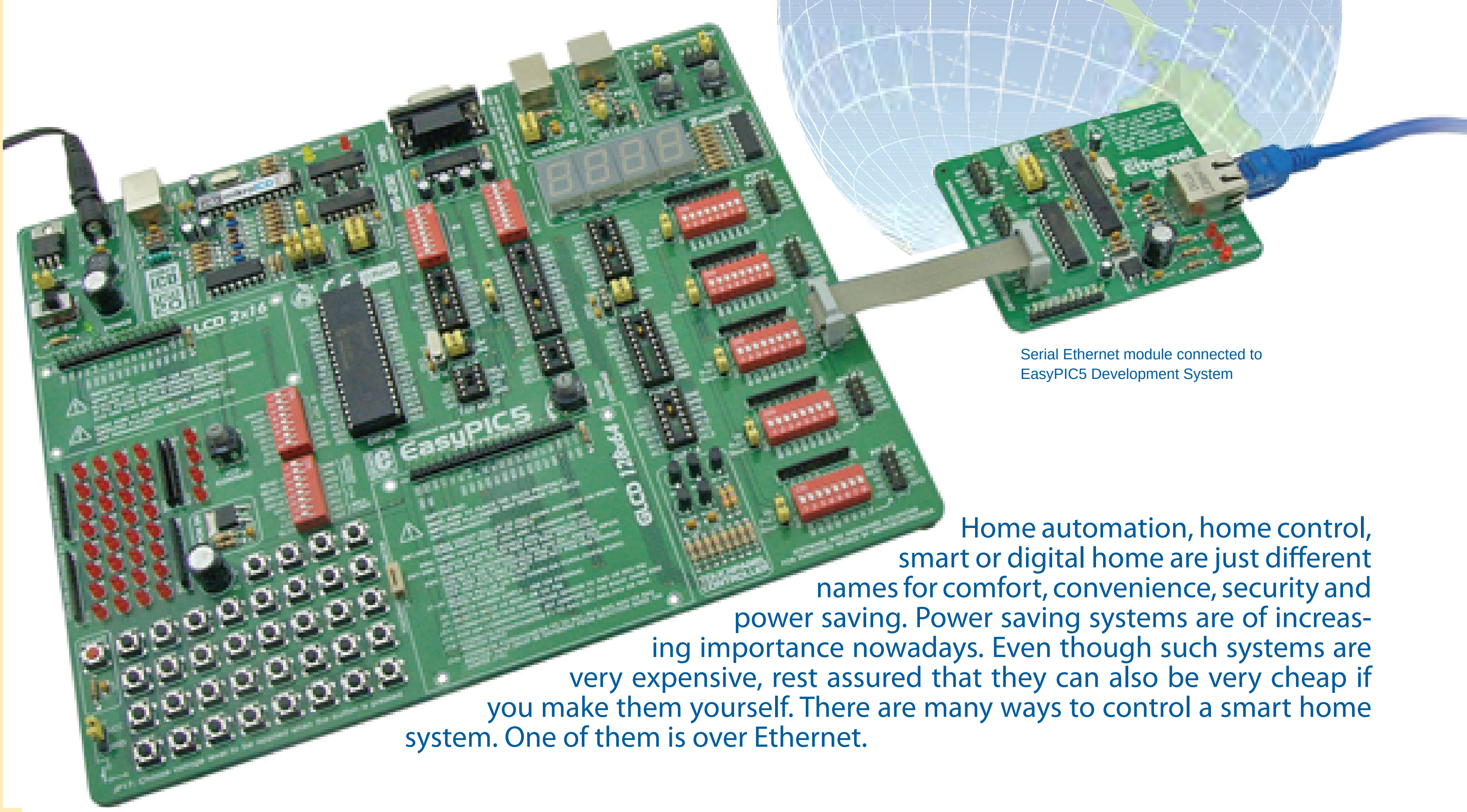
MailBox Terms

• 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 976,
Peterborough NH 03458-0876
USA.

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

OK. Now you need ... ETHERNET



Serial Ethernet module connected to EasyPIC5 Development System

Home automation, home control, smart or digital home are just different names for comfort, convenience, security and power saving. Power saving systems are of increasing importance nowadays. Even though such systems are very expensive, rest assured that they can also be very cheap if you make them yourself. There are many ways to control a smart home system. One of them is over Ethernet.

By Srdjan Tomic
MikroElektronika - Software Department

All you need is a PIC18F4520 microcontroller and an ENC28J60 serial Ethernet chip. This chip is a great solution for other microcontroller families as well such as AVR, dsPIC etc. The CviLux CJCBA8HF1Y0 RJ-45 connector is used for connection to the Ethernet network. An LED connected to the microcontroller PORTB.pin0 simulates a home appliance we want to control.

The *mikroC for PIC* compiler contains the *SPI_Ethernet* library that will considerably simplify the process of writing a program for the microcontroller. By using a few routines from this library, it is possible to create the program that will enable electrical appliances in your home to be controlled via a web browser.

It is necessary to perform the following operations within the program:

Step 1. Create an html page to run the microcontroller through. Import it in the code as a string.

Step 2. Set IP, DNS, Gateway addresses and Subnet mask obtained from your internet provider.

For example, our local network parameters are as follows:

IP : 192.168.20.60 (Control System address)
DNS : 192.168.20.1 (Domain Name System address)
GATEWAY : 192.168.20.6 (Gateway address)
SUBNET : 255.255.255.0 (Subnet mask)

Step 3. Disable PORTB analogue inputs. The microcontroller pin should be cleared and configured as an output.

Step 4. Initialize the SPI module of the PIC18F4520 microcontroller.

Step 5. Initialize the Serial Ethernet module chip ENC28J60.

Step 6. Write the code within the *Spi_Ethernet_userTCP* function that will, after receiving command via web browser, turn on/off the LED connected to the PORTB.pin0.

Step 7. Read received data in an endless loop.

The most important part of the program is the Spi_Ethernet_userTCP function, processing all received commands

After the web browser “GET” request is received, sent from your computer to the control system IP address, the microcontroller will respond with a web page stored in its memory. This page will then be automatically displayed on the computer screen by the browser.
When the ON command is received, the LED connected to the PORTB.pin0 will be turned on.
Likewise, when the OFF command is received the LED will be turned off. If you have a relay instead of LED, it is possible to control any appliance such as lighting, security system, heating system etc.

The control of any home appliance consists of entering control system IP address in the web browser and specifying the desired commands.

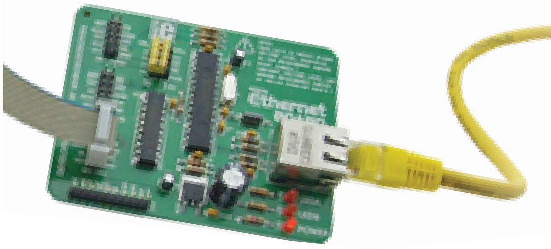
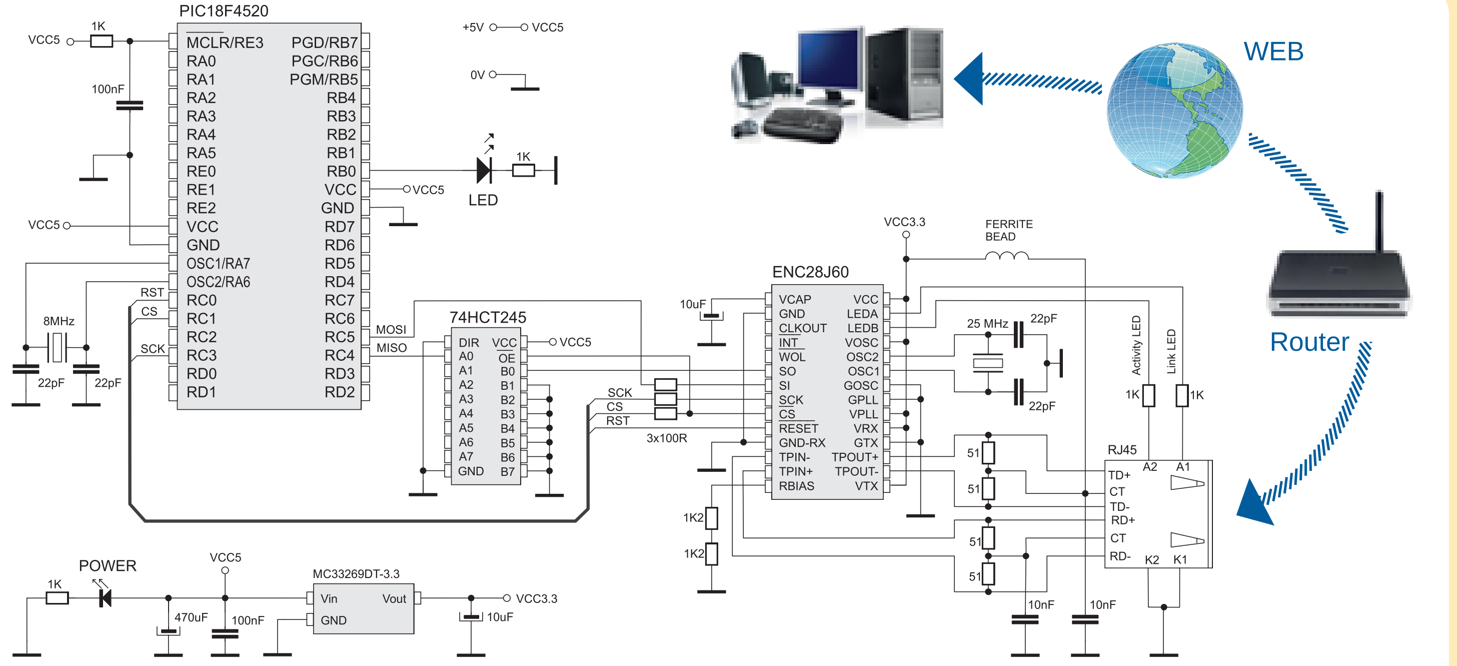
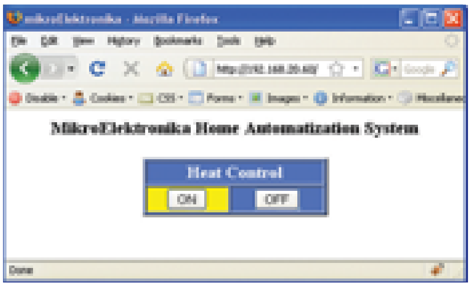


Figure 1. MikroElektronika’s *Serial Ethernet* module with ENC28J60 chip



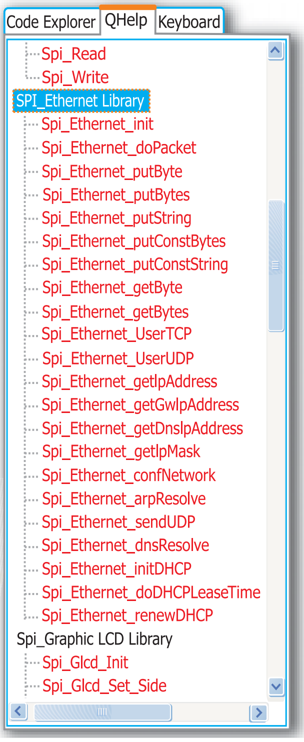
Schematic 1. Connecting the *Serial Ethernet* module to a PIC18F4520

Of course, it is possible to control more than one microcontroller pin, which enables you to govern a large number of appliances or complete home automation system as well.


mikroElektronika - Mozilla Firefox
MikroElektronika Home Automatization System
Heat Control
ON
OFF
Done

The screendump illustrates the web page displayed by the web browser after entering the control system IP address. In our example, ON and OFF button clicks cause the LED to be turned on and off, thus simulating the heating control system.

Below is a list of ready to use functions contained in the *SPI Ethernet Library*.This library is integrated in *mikroC for PIC* compiler.



- Spi_Read
- Spi_Write
- SPI_Ethernet Library
  - Spi_Ethernet_init
  - Spi_Ethernet_doPacket
  - Spi_Ethernet_putByte
  - Spi_Ethernet_putBytes
  - Spi_Ethernet_putString
  - Spi_Ethernet_putConstBytes
  - Spi_Ethernet_putConstString
  - Spi_Ethernet_getByte
  - Spi_Ethernet_getBytes
  - Spi_Ethernet_UserTCP
  - Spi_Ethernet_UserUDP
  - Spi_Ethernet_getIpAddress
  - Spi_Ethernet_getGwIpAddress
  - Spi_Ethernet_getDnsIpAddress
  - Spi_Ethernet_getIpMask
  - Spi_Ethernet_confNetwork
  - Spi_Ethernet_arpResolve
  - Spi_Ethernet_sendUDP
  - Spi_Ethernet_dnsResolve
  - Spi_Ethernet_initDHCP
  - Spi_Ethernet_doDHCPLeaseTime
  - Spi_Ethernet_renewDHCP
- Spi_Graphic LCD Library
  - Spi_Glcd_Init
  - Spi_Glcd_Set_Side

Spi_Ethernet_Init()*	**Init ENC28J60 controller**
Spi_Ethernet_Enable()	**Enable network traffic**
Spi_Ethernet_Disable()	**Disable network traffic**
Spi_Ethernet_doPacket()*	**Process received packet**
Spi_Ethernet_putByte()	**Store a byte**
Spi_Ethernet_putBytes()	**Store bytes**
Spi_Ethernet_putConstBytes()	**Store const bytes**
Spi_Ethernet_putString()*	**Store string**
Spi_Ethernet_putConstString()*	**Store const string**
Spi_Ethernet_getByte()*	**Fetch a byte**
Spi_Ethernet_getBytes()	**Fetch bytes**
Spi_Ethernet_UserTCP()*	**TCP handling code**
Spi_Ethernet_UserUDP()	**UDP handling code**
Spi_Ethernet_getIpAddress()	**Get IP address**
Spi_Ethernet_getGwIpAddress()	**Get Gateway address**
Spi_Ethernet_getDnsIpAddress()	**Get DNS address**
Spi_Ethernet_getIpMask()	**Get IP mask**
Spi_Ethernet_confNetwork()*	**Set network parameters**
Spi_Ethernet_arpResolve()	**Send an ARP request**
Spi_Ethernet_sendUDP()	**Send an UDP packet**
Spi_Ethernet_dnsResolve()	**Send an DNS request**
Spi_Ethernet_initDHCP()	**Send an DHCP request**
Spi_Ethernet_doDHCPLeaseTime()	**Process lease time**
Spi_Ethernet_renewDHCP()	**DHCP renewal request**

*** SPI Ethernet Library functions used in program**

Other *mikroC for PIC* functions used in program:

Spi_Init()	**Initialize microcontroller SPI module**
memcpy()	**Copy microcontroller RAM memory locations**
memcmp()	**Compare microcontroller RAM memory locations**

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

Example 1: Program to demonstrate control over Ethernet

```
 // duplex config flags
#define Spi_Ethernet_HALFDUPLEX    0x00                         // half duplex
#define Spi_Ethernet_FULLDUPLEX    0x01                         // full duplex

const char httpHeader[] = "HTTP/1.1 200 OK\nContent-type: "; // HTTP header
const char httpMimeTypeHTML[] = "text/html\n\n";              // HTML MIME type
const char httpMimeTypeScript[] = "text/plain\n\n";           // TEXT MIME type

// default html page
char   indexPage[] =
"<html><head><title>mikroElektronika</title></head><body>\
<h3 align=center>MikroElektronika Home Automatization System</h3>\
<form name=\"input\" action=\"/\" method=\"get\">\
<table align=center width=200 bgcolor=#4974E2 border=2><tr>\
<td align=center colspan=2><font size=4 color=white><b>Heat Control</b></font>\
</td></tr><tr><td align=center bgcolor=#4974E2><input name=\"tst1\" width=60 \
type=\"submit\" value=\"ON\"></td><td align=center bgcolor=#FFFF00>\
<input name=\"tst2\" type=\"submit\" value=\"OFF\"></td></tr></table>\
</form></body></html>";

// network parameters
char   myMacAddr[6] = {0x00, 0x14, 0xA5, 0x76, 0x19, 0x3f}; // my MAC address
char   myIpAddr[4]  = {192, 168, 20, 60};                    // my IP address
char   gwIpAddr[4]  = {192, 168, 20,  6};                    // gateway IP address
char   dnsIpAddr[4] = {192, 168, 20,  1};                    // dns IP address
char   ipMask[4]    = {255, 255, 255, 0};                    // subnet mask
// end network parameters

unsigned char   getRequest[20];                               // HTTP request buffer

unsigned int SPI_Ethernet_UserTCP(   char *remoteHost, unsigned int remotePort,
                      unsigned int localPort, unsigned int reqLength)
{
 unsigned int    len;                            // my reply length
 if(localPort != 80) return(0);                  // I listen only to web request on port 80

 // get 10 first bytes only of the request, the rest does not matter here
 for(len = 0 ; len < 15 ; len++) getRequest[len] = SPI_Ethernet_getByte();
 getRequest[len] = 0;

 if(memcmp(getRequest, "GET /", 5)) return(0);  // only GET method

 if(!memcmp(getRequest+11, "ON", 2))            // do we have ON command
  PORTB.F0 = 1;                                 // set PORTB bit0
 else
  if(!memcmp(getRequest+11, "OFF", 3))          // do we have OFF command
   PORTB.F0 = 0;                                // clear PORTB bit0

 if (PORTB.F0)
  {
   memcpy(indexPage+340, "#FFFF00", 6);         // highlight (yellow) ON
   memcpy(indexPage+431, "#4974E2", 6);         // clear OFF
  }
 else
  {
   memcpy(indexPage+340, "#4974E2", 6);         // clear ON
   memcpy(indexPage+431, "#FFFF00", 6);         // highlight (yellow) OFF
  }

 len =  SPI_Ethernet_putConstString(httpHeader);               // HTTP header
 len += SPI_Ethernet_putConstString(httpMimeTypeHTML); // with HTML MIME type
 len += SPI_Ethernet_putString(indexPage);                     // HTML page first part
 return len; // return to the library with the number of bytes to transmit
}
unsigned int SPI_Ethernet_UserUDP(   char *remoteHost, unsigned int remotePort,
                      unsigned int destPort, unsigned int reqLength)
{
 return 0; // back to the library with the length of the UDP reply
}
void main()
{
 ADCON1 |= 0x0F ;                  // no analog inputs
 CMCON  |= 0x07 ;                  // turn off comparators

 PORTB.F0 = 0;
 TRISB.F0 = 0;                     // set PORTB.B0 as output (rele control pin)

 // starts ENC28J60 with: reset bit on PORTC.F0, CS bit on PORTC.F1,
 //                        my MAC & IP address,  full duplex
 Spi_Init();
 // full duplex, CRC + MAC Unicast + MAC Broadcast filtering
 Spi_Ethernet_Init (&PORTC, 0, &PORTC, 1,
                  myMacAddr, myIpAddr, Spi_Ethernet_FULLDUPLEX) ;

 // dhcp will not be used here, so use preconfigured addresses
 SPI_Ethernet_confNetwork(ipMask, gwIpAddr, dnsIpAddr);

 while(1) {                        // do forever
  SPI_Ethernet_doPacket();         // process incoming Ethernet packets
 }
}
```

events

March 31, April 1 and 2, 2009

Embedded Systems Conference Silicon Valley 2009

San Jose, CA - USA

Elektor USA magazine will be present on the Exhibition floor at Embedded Systems Conference in San Jose, CA on March 31, April 1 and 2, 2009. The ESC Conference with its renowned sessions spans March 29 through April 3, 2009.

At booth # 47 you will be able to meet up with Elektor USA representatives Hugo Vanhaecke (Publisher), Jan Buiting (Editor) and Peter Wostrel (Advertising). Visitors will also be able to see selected Elektor Shop items and recently published projects. Subscribers to Elektor USA are expressly invited to the booth to supply feedback on the quality and direction of the magazine.

Come see us at ESC 2009!

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

Rabbit: new wireless single-board computers

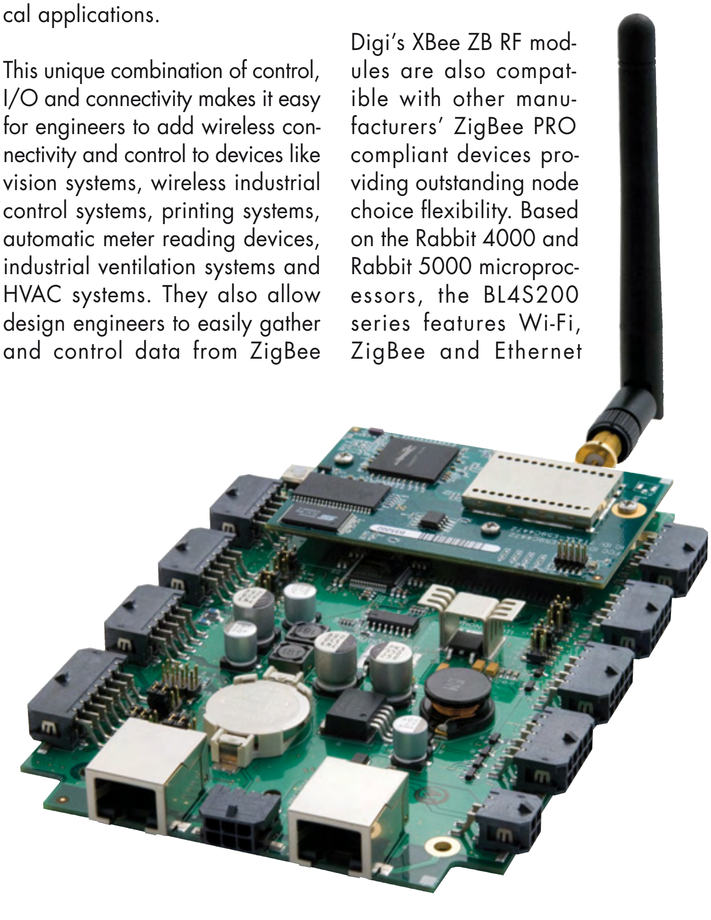
Rabbit introduced the BL4S100 and BL4S200, single board computers (SBCs) that feature either Wi-Fi or ZigBee® connectivity, a microprocessor, memory and abundant I/O that allow easy deployment of wireless nodes for industrial, commercial and medical applications.

This unique combination of control, I/O and connectivity makes it easy for engineers to add wireless connectivity and control to devices like vision systems, wireless industrial control systems, printing systems, automatic meter reading devices, industrial ventilation systems and HVAC systems. They also allow design engineers to easily gather and control data from ZigBee nodes and uplink the data to a server via Ethernet providing an easy to deploy ZigBee-to-Ethernet gateway.

The BL4S100 series offers both ZigBee and Ethernet connectivity and provides embedded design engineers a simple and straightforward approach for machine control and data acquisition without the burden of cables and wiring harnesses. It allows design engineers to deploy ZigBee nodes at various control points and connect those nodes wirelessly to the BL4S100 board. The BL4S100 SBC can then gather and collate the data from the ZigBee nodes and uplink it to a server via Ethernet.

Based on the Rabbit® 4000 microprocessor, the BL4S100 series features analog inputs, general purpose I/O, serial ports, and ZigBee and 10 Base-T Ethernet connectivity to provide design resources for commercial and industrial applications.

Digi's XBee ZB RF modules are also compatible with other manufacturers' ZigBee PRO compliant devices providing outstanding node choice flexibility. Based on the Rabbit 4000 and Rabbit 5000 microprocessors, the BL4S200 series features Wi-Fi, ZigBee and Ethernet networking options.



It also includes a broad range of configurable I/O resources, I/O expansion using Rabbit's RIO chip, and multi-channel analog inputs and outputs.

BL4S100 and BL4S200 development kits are available that include the essential hardware and software tools necessary, including Rabbit's popular Dynamic C integrated development environment, to make embedded development easy.

www.rabbit.com/products/SingleBoards/

(081040-IV)

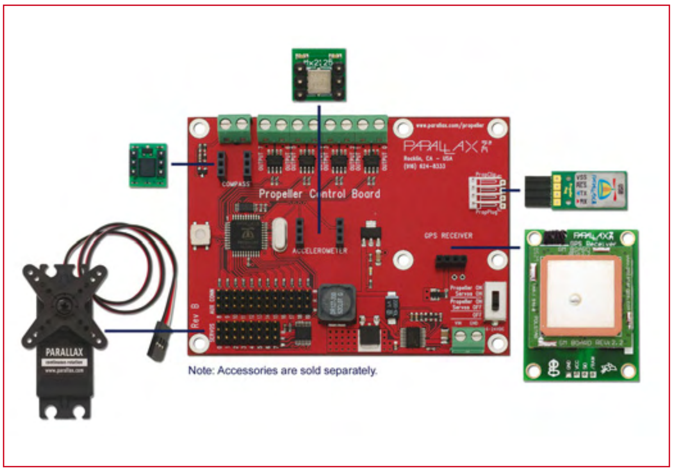
Yes we Can: www.elektor-usa.com

Elektor USA is progressing well in terms of popularity and subscribers. Near the time this edition is published, the publication will have its own website up and running at www.elektor-usa.com.
The magazine is now also available from selected bookstores allowing 'folks' to browse a copy before buying. Starting with the February 2009 issue, the magazine is stocked by selected Borders and Barnes & Noble bookshops in and around major cities in the USA. This may be followed by other bookstore 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.

www.elektor-usa.com

Parallax Propeller control board

The Parallax Propeller Control Board is a complete solution for controlling robotics platforms. This is the same control board that ships with the QuadRover robot. The Propeller Control Board is capable of interfacing directly with the Parallax GPS Module, the Hitachi HM55B Compass Module and the Hitachi H48C 3-Axis Accelerometer. This provides options for tracking position, heading and acceleration for robotic applications.
Twelve of the Propeller I/O pins are brought out for customer interfacing and control applications, 8 are dedicated to servo control and 8 are dedicated to N-Channel MOSFET outputs, providing high-current drivers for solenoids or other high-current devices. There is also a magneto-pickup compatible tachometer input which can be used for measuring the RPM of gasoline engines.
The power switch allows you to turn all power off, only the Propeller on, or the Propeller and servos on. There is also a reset button for the Propeller chip. A socketed 5 MHz crystal is included.
The Propeller Control Board costs U$ 195.99 (plus P&P).



Note: Accessories are sold separately.

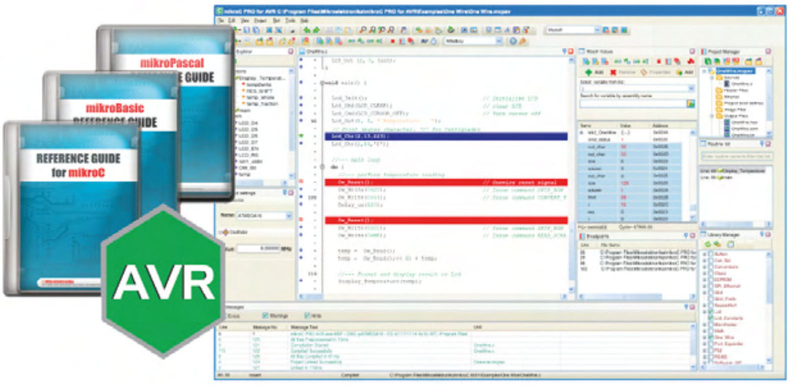
www.parallax.com (item # 28205)

mikroBasic PRO for AVR® 2008

MikroElektronika from Serbia have recently launched a new BASIC compiler for AVR® microcontrollers: mikroBasic PRO for AVR® 2008. The IDE features project-based design and supports an impressive range of AVR® microcontrollers. MikroBasic PRO for AVR® 2008 offers a set of libraries which simplify the initialization and use of AVR® MCU and its modules:

- ADC Library
- CANSPI Library
- Compact Flash Library
- EEPROM Library
- Flash Memory Library
- LCD Library
- Manchester Code Library
- Multi Media Card library
- OneWire Library
- Port Expander Library
- PS/2 Library
- PWM Library
- PWM 16 bit Library
- RS-485 Library
- Software I2C Library
- Software SPI Library
- Software UART Library
- Sound Library
- SPI Library
- SPI Ethernet Library
- TWI Library
- UART Library
- Button Library
- Conversions Library
- String Library

and more ...



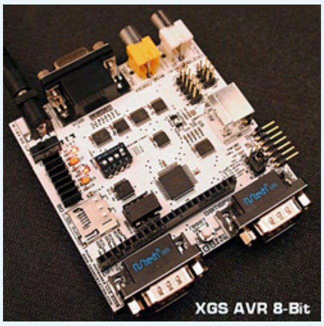
Also, the new compiler has plenty of practical examples and comprehensive documentation which allows a quick start in programming AVR®. AVR® hardware development tools that completely support mikroBasic PRO for AVR® 2008, are also available. A fully functional demonstration version (hex output is limited to 2 k of program words) is available on the MikroElektronika web site.

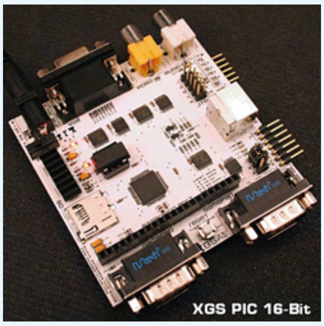
www.mikroe.com

New XGamestations XGS AVR 8-Bit & XGS PIC 16-bit

The **XGS AVR 8-Bit** from XGamestation (Nurve Networks) is based on the Atmel Mega AVR644 processor and is a highly integrated development kit for exploring the Mega AVR processors in a fun and engaging way. On the other hand, the product is designed to be a serious AVR development kit for schools, students, engineers and anyone interested in learning AVR programming. Clocked at over 28 MIPS the XGS AVR 8-Bit is a seriously powerful 8-Bit development board! The system has the following features:

- AVR644 processor with 64 k Flash / 4 k SRAM running at over 28 MIPS!
- 3.3/5 V Dual Supplies
- VGA
- NTSC/PAL with color generation helper hardware
- Micro SD card interface
- Serial port
- ISP and JTAG programming ports
- PS/2 keyboard/mouse port
- Expansion port header exporting numerous I/O, power, and signal lines for experimentation
- 3.3/5 V internally regulated supplies
- Two game controller ports (Nintendo compatible).


XGS AVR 8-Bit


XGS PIC 16-Bit

The XGS AVR 8-Bit kit gives customers a fun way to learn AVR programming in the context of graphics, audio, and simple game development. So instead of blinking LEDs and displaying digits on a 7-segment display, customers will develop graphics applications that control the VGA and NTSC screen to learn AVR C and ASM. The kit is extremely competitive and comes with the following items in the bundle:

- 350+ page printed manual covering hardware, software, and numerous programming tutorials
- AVR ISP MK II programmer + USB cable
- 9V power supply
- A/V cable
- Game controller
- DB9 PC serial port to XGS header converter
- DVD-ROM with numerous examples and complete driver library including; Graphics, Sound, Keyboard, SD card, Serial Comms, Mechatronics, and lots more

- Bonus materials on DVD-ROM include electronic copies of numerous game development and electronics gaming hardware books

The XGS AVR 8-Bit is priced at $139.99.

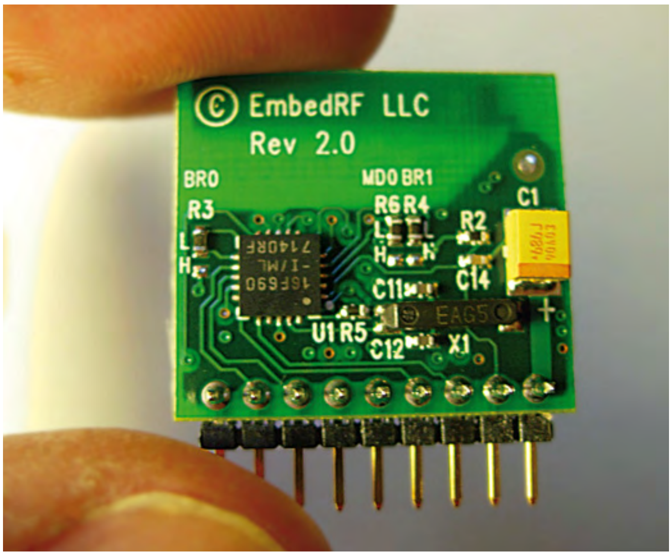
The **XGS PIC 16-Bit** is based on Microchip's new 16-bit PIC24 processor and is a highly integrated development kit for exploring Microchip's 16-Bit microcontrollers in a fun and engaging way. On the other hand, the XGS PIC 16-bit is designed to be a serious PIC24 16-bBit development kit for schools, students, engineers and anyone interested in learning PIC programming. The system has same features as the XGS AVR 8-Bit except a PIC24 16-bit next generation processor with 256 k Flash / 16 k SRAM running at over 40 MIPS!
The kit has the same contents as the XGS AVR 8-Bit, except a Pickit2 programmer + USB cable is supplied.
The XGS PIC 16-Bit is priced at $159.99.

www.xgamestation.com

Tiny Multipoint RF modules

Saelig's EmbedRF Integrated is a new RF solution for economically and rapidly adding low power wireless capabilities to your own analog or digital product, without requiring any network license fees or complex RF skills. Based on the proven EmbedRF board, EmbedRF Integrated consists of the EmbedRF Baseband Processor, a low cost Microchip PIC microcontroller preloaded with EmbedRF OS firmware and carefully designed Gerber design files for integrating an RF design directly into your own board.

The EmbedRF Baseband Processor with EmbedRF OS firmware handles the RF data transfer between two modules (base-slave or peer-to-peer). It has an intelligent search-and-link feature which enables a given EmbedRF module to initiate contact with only the EmbedRF chip in closest proximity. The EmbedRF OS was designed with power conservation in mind so that a single module can run for years off of a single Lithium coin cell. EmbedRF Integrated can be easily configured as a periodic transceiver for applications such as parameter or medical data- logging, or an on-demand transceiver for remote control applications. With 4 analog inputs, the EmbedRF Integrated design can directly interface with voltage-output sensors and transmit the information up to 50 feet (15 m).



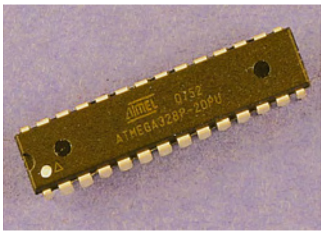
Operating with an output power up to 5 dBm at frequencies from 400 MHz to 928 MHz, EmbedRF Integrated features intelligent search and sophisticated 24-bit security capabilities. EmbedRF Integrated can work on 2.0 to 3.6 V from –40 degC to 85 degC without trimming. EmbedRF Integrated will operate in point-to-point one-way, two-way, or two-way multipoint star-connection.

Designing the EmbedRF Integrated wireless solution into your product costs under $8.00 per unit in quantity, including the Microchip controller, free EmbedRF software and APIs. Designed and made in the USA, the EmbedRF Development Kit is available now at $ 299, with individual modules priced at $ 59.

www.saelig.com

Low Cost ZX Microcontroller

Elba Corporation have started shipping the newest member of the powerful multi-tasking ZX microcontroller family intended for use by scientists, engineers, experimenters, hobbyists and other embedded systems developers. The new device, designated ZX-328n, is provided in a standard 28-pin (300 mil) DIP package. A few additional components (e.g. crystal, serial level converter) are required for device operation. All ZX-series microcontrollers are programmed in ZBasic, a subset of Microsoft's Visual Basic (VB6) with microcontroller-specific extensions and other productivity enhancements.



The ZX-328n is the newest member of a series of ZX devices that operate in 'native' mode, meaning that the user's ZBasic program is compiled to native machine code for the underlying microcontroller, in this case the Atmel ATmega328P MCU. This is in contrast to other ZX family members that use the 'virtual machine' model (also known as the interpreter model).

The new ZX-328n is largely source code compatible with the previously released ZX family members. The ZX-328n is intended for entry-level users and for simpler applications that can be implemented in the available 30 k of code space, 2 k of RAM and requiring 16 I/O lines or fewer. The single unit pricing of the ZX-328n is $ 9.95. Volume pricing is available.

Programs for the ZX-328n may be edited, compiled and downloaded using a state-of-the-art Integrated Development Environment. The ZBasic IDE provides productivity-enhancing features such as word completion, call tips, auto-indenting, syntax highlighting and undo/redo.

www.zbasic.net

Microchip: stand-alone serial SRAM devices

Microchip announces a family of 8- and 32-kByte stand-alone serial SRAM devices designed to increase a system's available RAM through adding small, inexpensive external devices. The 23A640, 23K640 (23×640), 23A256 and 23K256 (23×256) devices feature a familiar, industry standard SPI interface, providing increased design flexibility while reducing design and production costs.

Many embedded applications require volatile RAM for temporary data storage, or for use as a scratchpad, for bulk processing and for math algorithms. In many cases, this RAM is embedded within the microcontroller (MCU). In the past, the most viable way to add more RAM was to buy a larger MCU, which could add unnecessary feature overhead and increase design costs. The only alternative was to add large, parallel-access RAM devices that use up large numbers of I/O pins.

Microchip's serial SRAM devices provide a simple, inexpensive way for designers to add more RAM to their application while keeping the same MCU or, as they require fewer MCU I/O resources, even using a smaller MCU. The serial RAM devices require just four I/O pins as opposed to 16 or 24 pins for a parallel RAM. Additionally, the devices feature a bus speed of 20MHz for fast access, and low operating and standby currents to help extend battery life.

The 23A640 and 23A256 devices have an operating voltage range of 1.7 – 1.95V. The 23K640 and 23K256 devices have an operating voltage range of 2.7 – 3.6V.

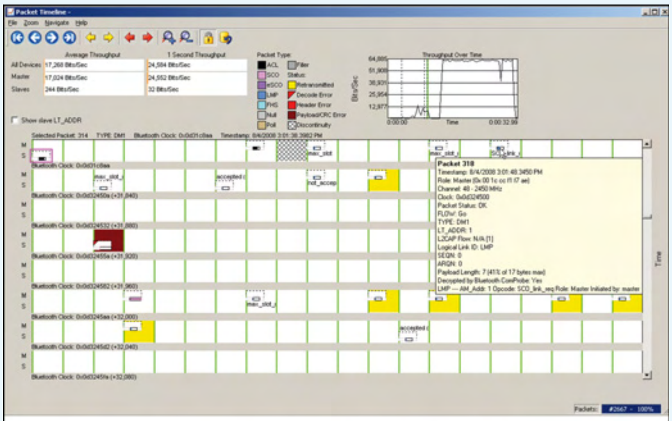
These serial SRAM devices are all available in 8-pin SOIC, PDIP and TSSOP packages. Samples are available today.

http://sample.microchip.com
www.microchipdirect.com

Bluetooth testing software

Electronic device OEMs use protocol analyzers to overcome the complexity of incorporating Bluetooth technology into existing products, greatly expanding their marketability.

Facilitating the entry of Bluetooth into the electronic device industry is the modern protocol analyzer, now updated to effortlessly examine not only the wireless interface between devices, such as a glucose monitor and a telephone modem, but also the interface between the device CPU and the Bluetooth chip inside the device. Medical device manufacturers can now release Bluetooth-enabled products with confidence that the item has been thoroughly tested and will perform to spec in the clinical environment.



	Average Throughput	1 Second Throughput
All Devices	17,268 Bits/Sec	24,584 Bits/Sec
Master	17,024 Bits/Sec	24,552 Bits/Sec
Slaves	244 Bits/Sec	32 Bits/Sec

Packet Type:

Status:

□ Show slave LT_ADDR

Selected Packet: 314 TYPE: DM1 Bluetooth Clock: 0x0d31c8aa Timestamp: 8/4/2008 3:01:38.3982 PM

Bluetooth Clock: 0x0d31c8aa

Bluetooth Clock: 0x0d32450a (+31,840)

Bluetooth Clock: 0x0d324532 (+31,880)

Packet 318
Timestamp: 8/4/2008 3:01:48.3450 PM
Role: Master (0x 00 1c cc f1 f7 ae)
Channel: 48 - 2450 MHz
Clock: 0x0d324500
Packet Status: OK
FLOW: Go
TYPE: DM1
LT_ADDR: 1
L2CAP Flow: N/A [1]
Logical Link ID: LMP
SEQN: 0
ARQN: 0
Payload Length: 7 (41% of 17 bytes max)
Decrypted by Bluetooth ComProbe: Yes
LMP — AM_Addr: 1 Opcode: SCO_link_req Role: Master Initiated by: master

Understanding Bluetooth communication is simplified by using a protocol analyzer. The Packet Timeline in Frontline's analyzer provides a graphic-based trace that shows high-level information such as data throughput and detailed packet-level information.

With over 40,000 units in use worldwide, spanning a wide variety of communication technologies, Charlottesville, Virginia-based Frontline Test Equipment, Inc. is a leading provider of PC-based protocol analyzers for special-purpose data communication networks.

Unlike protocol analyzers that can only store data for later evaluation, Frontline's analyzers feature real-time displays to further accelerate the debugging and verification process. The effect of user actions — whether 'positive path' or regressive — are observed in real time on the screen of the analyzer. The test engineer no longer has to rely on notes or memory to reconstruct what took place when attempting to fix any problems.

www.fte.com

Freescale: 'one-stop-shop' 32-bit industrial connectivity solution

Attention all fans of the Elektor DigiButler project.

To help developers successfully meet their design challenges, Freescale Semiconductor has introduced a 'one-stop-shop" industrial connectivity solution that combines its most highly integrated 32-bit ColdFire® microcontroller (MCU) family with a complimentary offering of the Freescale MQX™ real-time operating system (RTOS).

Freescale's high-performance MCF5225x MCU family offers the ultimate industrial connectivity solution with on-chip USB, Ethernet, controller area network (CAN) and encryption, along with Freescale MQX RTOS and associated tools and software stacks. The combined silicon and software solution gives developers exceptional design flexibility, connectivity options and fast time-to-market backed by a full-featured, scalable RTOS platform valued at approximately $95,000 (USD).

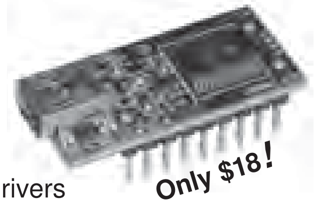
Based on the 32-bit ColdFire V2 core, the MCF5225x MCUs are well-suited for a broad range of industrial networking, building/lighting control and medical applications that require high performance and connectivity options. For factory automation systems, the MCF5225x MCU's USB port can be used to develop interfaces to standard USB barcode scanners, and the embedded Fast Ethernet controller makes it easy to control various terminals in a networked building.

By integrating USB device/host/on-the-go, Fast Ethernet and CAN on the same device, the MCF5225x family allows developers to implement the optimal connectivity protocol for their application requirements, without having to add separate communication controllers. The MCUs also include extensive integrated serial communications capabilities, such as a serial peripheral interface (SPI), two inter-integrated circuit (I2C) buses and three universal asynchronous receiver transmitters (UARTs).

The MCF5225x MCUs feature an on-chip cryptographic accelerator unit and random number generator designed to help safeguard valuable data and intellectual property during transmissions across industrial and building control networks. The encryption module supports DES, 3DES, AES, MD5 and SHA-1 algorithms.

Evaluation and demonstration systems for the MCF5225x family are available now. The M52259EVB evaluation board is a full-featured development system available at a suggested resale price of $ 299. The cost-effective M52259DEMOKIT demonstration board is available at a suggested resale promotional price of $ 49.

www.freescale.com/files/pr/coldfire.html
www.freescale.com/files/pr/mqx.html

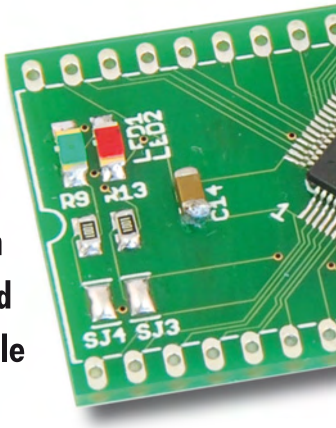

Only $18!

Microcontroller Systems Engineering
45 projects for PIC, AVR and ARM
Bert van Dam
elektor
NEW!

R32C and its Relativ

From R8C/13 to R32C/111

Marc Oliver Reinschmidt (Glyn Germany) and Bernd Westhoff (Renesas Technology Europe)



Our series on the R8C13 'Tom Thumb' was deservedly popular with Elektor readers, who will doubtless be pleased to hear this little 16-bit microcontroller has just gained a talented sibling. The R32C/111, his mathematically gifted big brother, is blessed with 32-bit architecture, Floating Point Unit arithmetic and a 50 MHz clock rate. The R8C and the R32C are both from Renesas' M16C family, making the upgrade to the more versatile 32-bit system extremely simple.

For years 32-bit microcontrollers have proliferated in applications ever more numerous. Anyone who was a serious devotee of these old 8 or 16-bit devices must by now be using a high-end machine to keep up with the demands of today's applications. It's for this reason that Renesas has expanded the well-known M16C family and brought out its new R32C 32-bit controller.

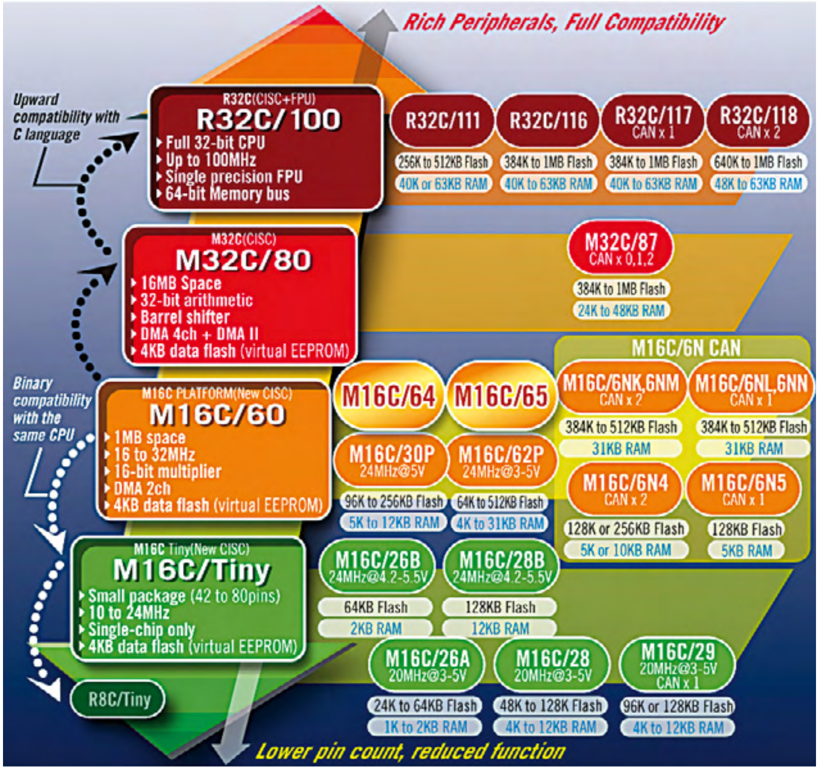


Figure 1. The M16C platform ranges from the R8C/Tiny up to the R32C/100 series with 32-bit CPUs.

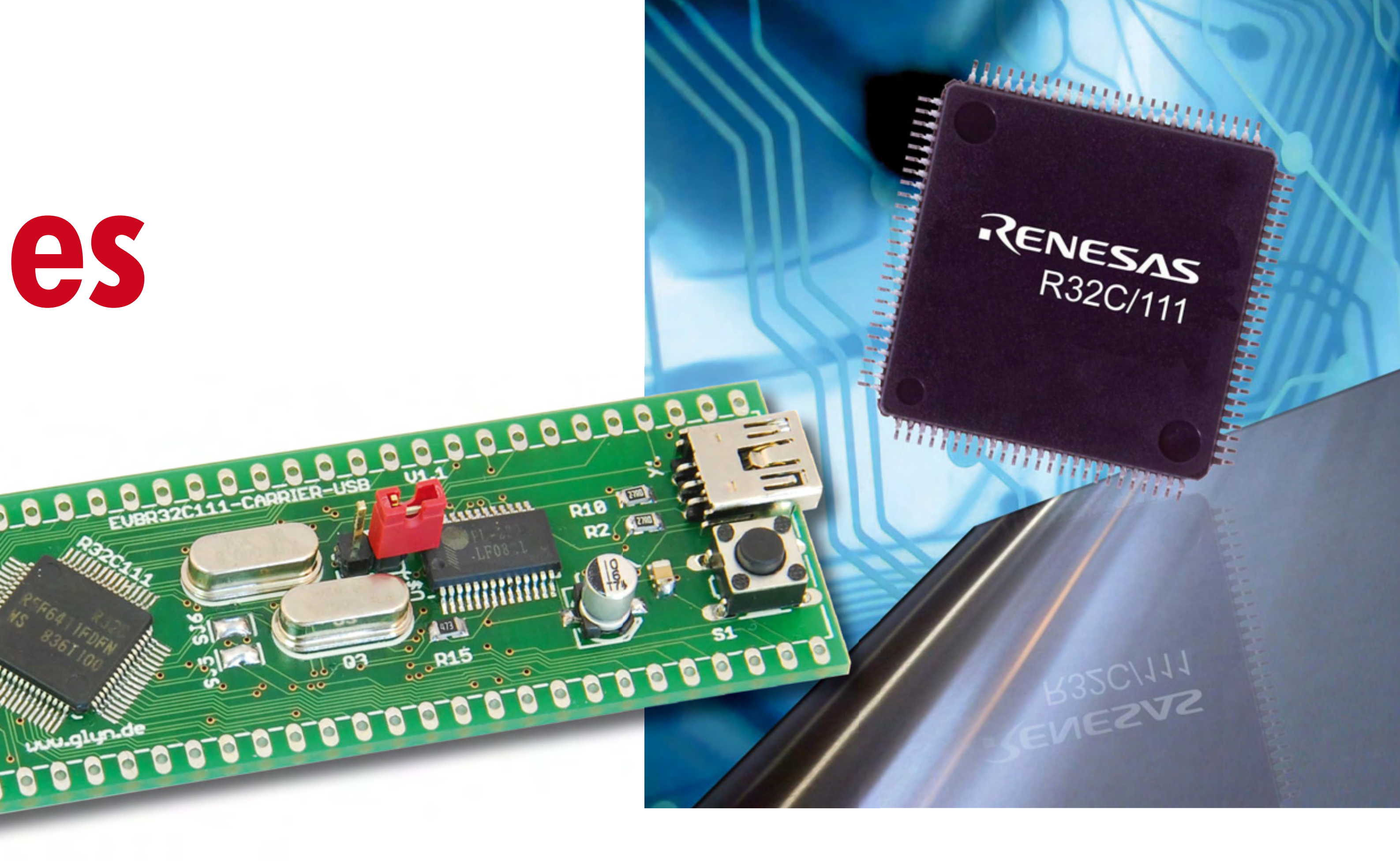
The R32C is based on the M16C (**Figure 1**), including its peripheral details and programming techniques, which means that to a large extent existing code and functions already developed can be used afresh. Only the core itself has been changed, in order to support a comprehensive 32-bit architecture. The benefits are very obvious: if you're already familiar with microcontroller structure — even the R8C for instance — you'll have no difficulties changing up to this new 32-bit derivative.

Memory magnitudes

The R32C provides internal flash memory that is available in sizes from 256 up to 1 MB according to requirements. RAM also comes in a variety of dimensions and on the R32C/111 these are even independent of the Flash memory volume.

The multiplicity of potential applications makes this kind of flexibility vital. For example if you are designing a display for large text arrays in multiple languages, you will require plenty of flash memory but not much RAM. Designing a communication module would call for exactly the opposite. Here a relatively large amount of RAM is required to handle communi-

es



cation stacks, whilst a smaller quota of flash memory is called for. Renesas has christened these combination options, shown in **Figure 2,** with the name *FlatRAM™.*

A further advantage of this flexibility is seen at development phase. It's now a reality for designers to select a chip with the maximum amount of memory available and further down the line refine the specification, replacing it with a device using the minimum necessary RAM and flash resources. This helps you manage the purchase budget far better, putting a smile on the face of all the bean counters.

Heart of the system

Figure 3 is a block diagram of the R32C/111. The core — the new 32-bit CISC MCU — is based on the proven Von Neumann architecture. It includes two register banks with a total of sixteen 32-bit universal address registers and four 32-bit frame/static base registers. The majority of the 108 CPU commands require only one to three CPU clock cycles for execution, which demonstrates the high performance of the CISC core. Further features enhance the processing speed for arithmetic calculations in particular:

- Hardware multiplier: 32 bit × 32 bit = 64 bit
- Repeat Multiply and Accumulate (RMPA) with 32 bit × 32 bit + 64 bits → 64 bits
- Floating Point Unit (FPU) with simple accuracy to the IEE 754 Standard
- 32-bit Barrel Shifter
- 64-Byte Instruction Queue

The maximum clock rate of the R32C/111 device amounts to 50 MHz, along with a minimum command cycle time of 20 ns. Taking into account the overall performance, this controller achieves 42 Dhrystone MIPS. But it can manage even better: by reducing the most significant and most frequently used op-codes to just one byte long, the program code can be optimized more tightly, so that the processing speed can be increased further, with less need for flash memory.

Maths genius

Let's now take a closer look at the arithmetic qualities of the controller. A floating point unit has been integrated in the CPU of the R32C. Known as an FPU for short, this well-known feature speeds up calculations that used to occupy so much time and produces a speed improvement factor of around 50. Floating point calculations we could mention include accumulating, multiplying, dividing, subtracting, con-

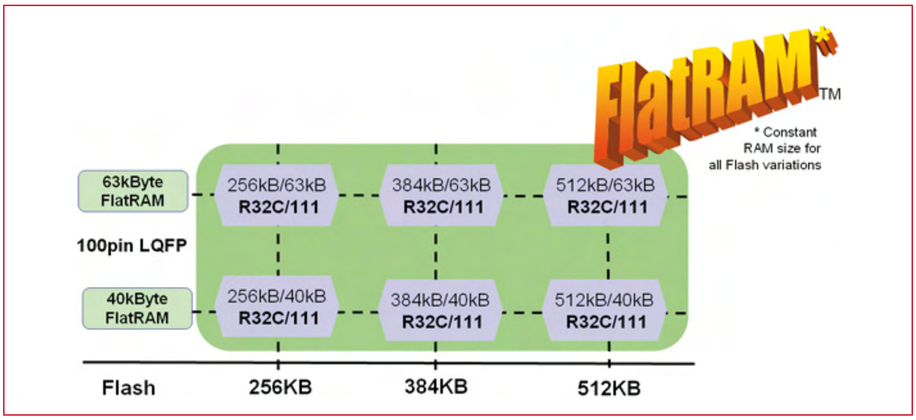


Figure 2. With the R32C111 Renesas offers a variety of flash memory options for a given RAM array.

verting, comparing and rounding. If the FPU is not needed, however, it can be turned off easily as a compiler option in the development environment.

In addition the R32C also provides a 32-bit multiplier, a 32-bit Barrel Shifter and a Repeat Multiply/Accumulate function very handy for filter calculations. This comprehensive arithmetic capability also makes the controller suitable for many applications that require DSP functionality.

A further significant feature for current applications is the rapid reaction to Interrupts external to the system. To this end the register sets in the R32C core are duplicated. When an Interrupt is called switching between the register banks can be effected rapidly without time-consuming Push and Pop instructions with access open to the stack. This switching is achieved simply by setting a Flag (see **Figure 4**), so that the Interrupt reaction time can be crunched down to just 0.1 μs.

Another peep under the black ‘engine cover’ of the R32C reveals some more details of how it all ticks. Internally the CPU is connected to a 64 bit-wide memory bus, in order to ensure the fastest possible data exchange. If you examine the 16-bit peripheral bus, you will spot components already proven in the M16C family to match up to a 32 MHz data rate. The external 16/8-bit bus also reaches 32 MHz and by using the four Chip Select lines, it can address up to 64 MB externally and this with individual Wait States per selected domain.

Control of the internal buses is handled by the Bus Interface Unit (BUI).

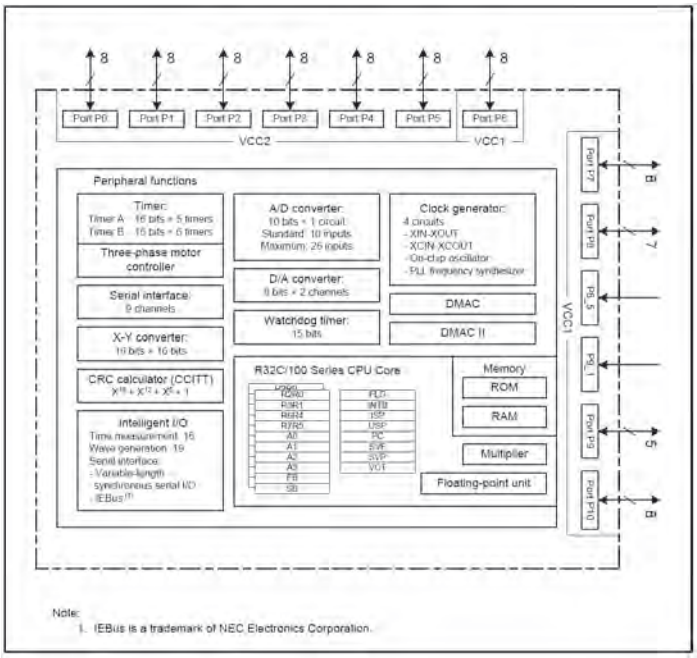


Figure 3. Block diagram of the R32C/111.

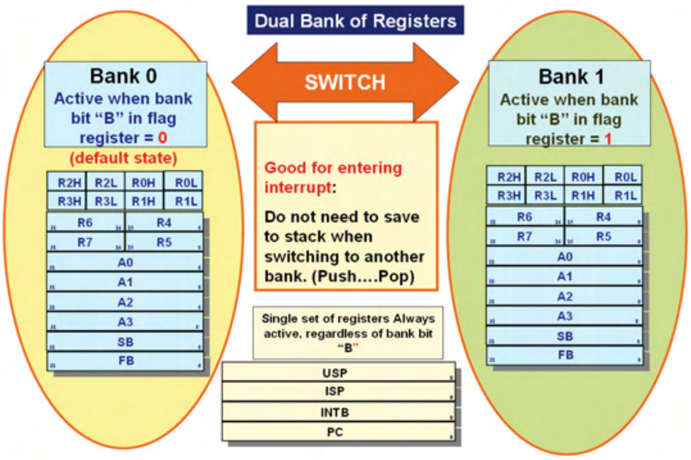

Dual Bank of Registers
SWITCH
Bank 0
Active when bank bit “B” in flag register = 0
(default state)
Bank 1
Active when bank bit “B” in flag register = 1
R2H
R2L
R0H
R0L
R3H
R3L
R1H
R1L
R6
R4
R7
R5
A0
A1
A2
A3
SB
FB
Good for entering interrupt:
Do not need to save to stack when switching to another bank. (Push….Pop)
Single set of registers Always active, regardless of bank bit “B”
USP
ISP
INTB
PC

Figure 4. The register sets in the R32C core are duplicated in order to shorten the Interrupt reaction time to just 0.1 μs.

On the periphery

“Seen one, seen them all” — or so it might appear. Peripheral functions are based on the trusted structure of the M16C family, enabling you to re-use code that you have already written and profit from the collective know-how assembled up to now.

When it comes to interference protection and emissions the R32C is well equipped. Reduced electromagnetic interference (EMI) products, high electromagnetic susceptibility (EMS) protection and modest power dissipation simplify circuit layout at the design and planning stage. As with all previous controllers of the M16C series, the R32C is also pin-compatible with its predecessors such as the M16C62P. Accordingly you can use it to replace controllers already in use without problem, if their performance no longer meets your requirements.

Now here’s a round-up of the most important peripheral elements. The R32C provides eleven 16-bit timers, asynchronous and synchronous interfaces, a four-channel DMAC, a 10-bit A/D converter with up to 34 input channels and an 8-bit D/A converter. Companion features include three intelligent I/O units that can be programmed for an extremely varied range of purposes, such as Input Capture/Output Compare function, an additional synchronous serial interface with variable character length and digital filters.
Within the R32C family you have also recourse to controllers that pro-

vide a CAN interface (or even two). In this way the development process can be optimised within a processor family according to cost and power requirements.

The controller also distinguishes itself by its low current consumption of only 28 mA at 3 V and 50 MHz clock frequency. If the clock rate is dropped, the current through the internal 125 kHz oscillator or an external 32 kHz crystal can be reduced to below 8 μA. The R32C operates on a supply voltage of either 3 V or 5 V, with no restrictions on the clock rate.

Application

The R32C, like the other controllers of the M16C family, is designed for applications in the consumer and industrial fields. Its versatile functionality makes it suitable for use in industrial automation, security equipment, gateway functions, motor control or retail automation. A number of emulator systems are available to developers. Basic debugging functions can be carried out without additional hardware, as on the R8C13, using the serial interface. We'll describe this process in greater detail in the next issue of Elektor. If you prefer a more comfortable debugging environment or need more functions you can employ the E8a or the E30a Debugger.

For your software development environment the HEW4 (High-performance Embedded Workshop) is at your disposal. This software package contains an Editor, C-Compiler, Debugger and Project Manager in a single combo and is ready for you to download *gratis* on the Renesas website. After installation all features of the Compiler can be used immediately. After a trial phase of 60 days it becomes limited to 64 kB of code. Alternatively an unrestricted KPIT GNU C Compiler is in preparation and this allows debugging with the HEW. More on the theme of software and downloads in the next issue.

(080082-I)

The Authors

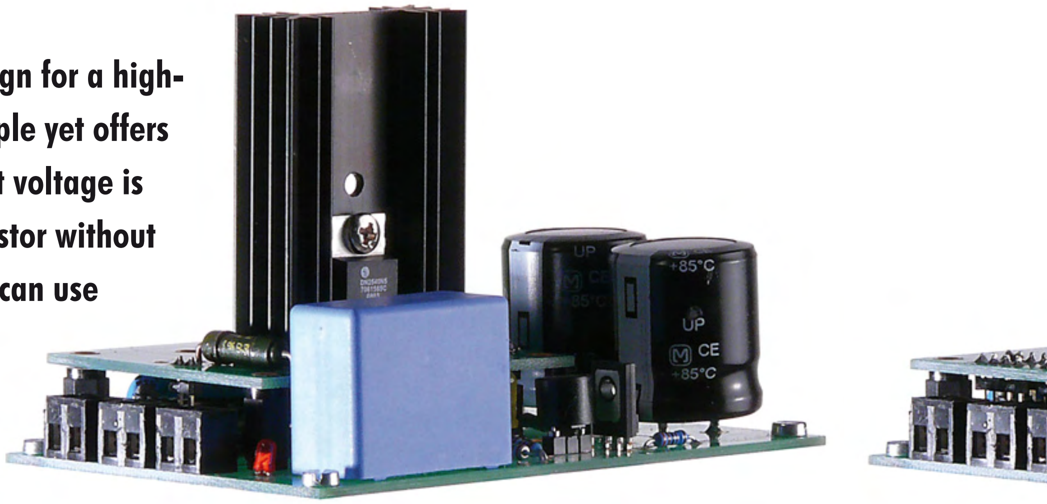
Marc Oliver Reinschmidt is an applications engineer with Glyn Germany, where he holds special responsibility for the M16C and R32C microcontroller range. His next exercise for Elektor is a small sample application using the R32C carrier board, which we will make available in the Elektor Shop. Also in preparation is a joint collaboration with Elektor Labs for an application board for projects using OLED displays…

Bernd Westhoff at Renesas Technology Europe looks after product marketing of the M16C, M32C and R32C MCU series to consumer, industrial and distributor markets.

T-Reg A high-voltage r for tube amps

Jan Didden (The Netherlands)

This article describes a novel design for a high-voltage regulator that is very simple yet offers excellent performance. The output voltage is fully adjustable with a single resistor without any changes in performance. You can use several types of tubes or MOSFETs for the pass device in the circuit.



Quick Project Specs

- **Universal design, output voltage easily adjustable with one resistor**
- **Low output impedance and low hum and noise over the complete audio band**
- **Output voltage adjustable between 0 and 500 V, up to 1 A output current**
- **Usable for tube preamplifiers and (single ended) power amps**
- **Build-in delay function (30 s) for output voltage after power-on**
- **Three plug-in options available for the pass device:**
  with 6528 valve (max. 600 mA);
  with EL84/6BQ5 or EL86/6CW5 valve (max. 50 mA),
  or with DN2540 MOSFET (max. 1 A)

Tube amplifiers can benefit from regulated and stabilized power supplies, especially sensitive preamplifier stages, but also (single-ended) power amps. The assumption is that a DC power source is, well, just that: direct voltage, i.e. with no hum, noise and ripple from the signal frequency.
But there's always ripple and noise from the power lines through the rectifier and smoothing circuits. Even if you get fancy with C-L-C-*etcetera* circuits, it's never completely clean. And the load current, which varies with signal frequency, flows through the internal resistance of the supply and Mr. Ohm showed us that [current × resistance] causes a signal-related voltage on the supply.

Every amplifier has a property called Power Supply Rejection Ratio (PSRR). That is a measure of how much of the ripple and hum on the supply bleeds through to the output. It varies with amplifier topology, but it's always there. Especially single-ended circuits are poor performers in this respect.

So, a regulated supply that minimizes ripple, noise and hum on the HV supply line is quite useful for clean, undistorted sound. Traditional high-voltage regulators add several extra tubes, complexity and power losses, while good performance is not easy to obtain.

This article describes a novel design that is very simple, yet offers excellent performance. Output voltage is fully adjustable with a single resistor without any changes in performance.

Concept

The concept of my circuit is shown in **Figure 1** (how I got the idea is described in the **inset**). You're looking at a high-voltage regulator, but without relatively rare and expensive high-voltage devices and parts. That's why there is a separate low-voltage supply, floating on top of the high-voltage output. This low-voltage supply, powered from the rectified pass tube heater, runs the reference and control circuits which can be low-voltage stuff.
This type of regulator always consists of the same parts. There is always a

egulator

reference voltage and an error amplifier that compares the reference voltage to (a sample of) the output voltage. The error amplifier then drives the pass element to keep the output constant.

In my circuit, the reference voltage comes from resistor R3, driven from the floating constant-current source T1. Transistors T2 and T3 form the error amplifier. It is a differential amplifier but with complementary devices rather than a difference amp with two like devices. But it works the same: the reference is input at T3, and the output voltage at T2. If the output voltage starts to fall below the set value, T2 will start to conduct harder and the drive voltage for the pass device across R5 will increase. This in turn raises the output until it is again at the set value. When V_{out} becomes too high, the drive is reduced, lowering V_{out} down to the set value. Since V_{out} is equal to V_{ref} except for a 2 V_{b-e} offset, in principle you can set the output voltage to anything you want by just selecting R3. This is a great advantage to more conventional setups. Traditionally, you would have a reference voltage that

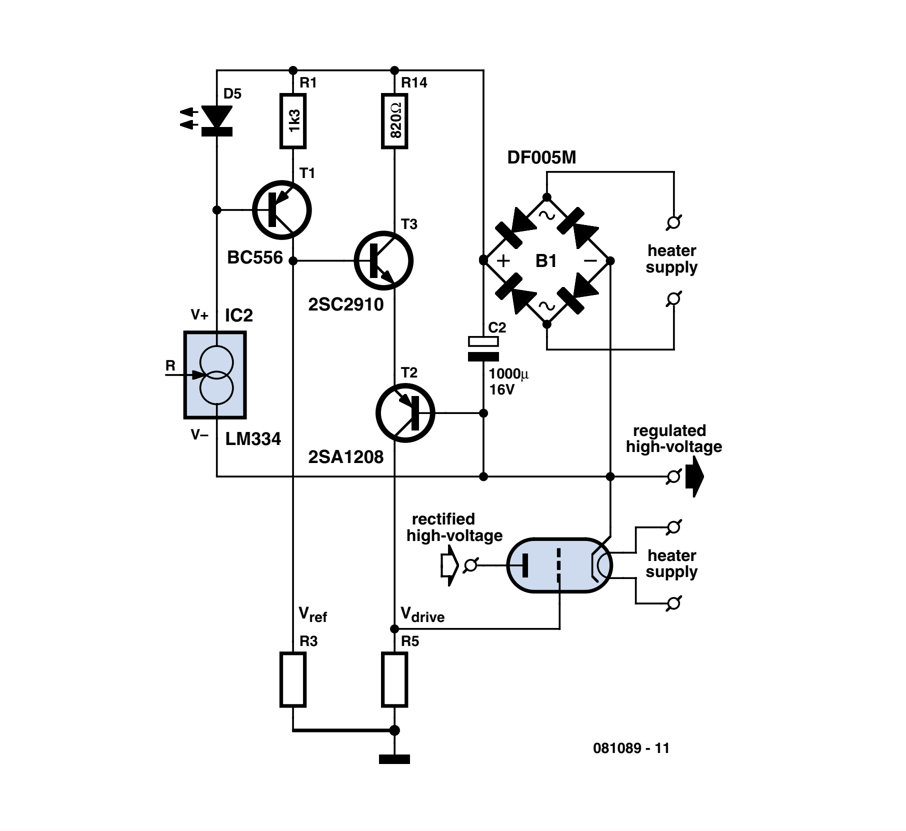


Figure 1. T-Reg conceptual diagram.

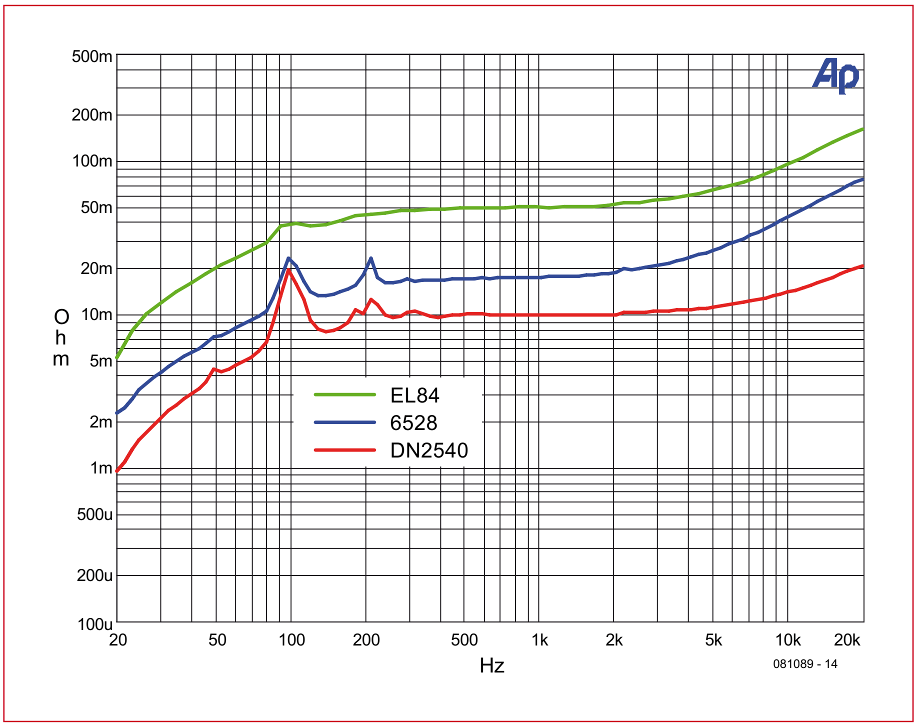


Figure 2. T-Reg output impedance plots.

is some fraction of V_{out}, and then you divide V_{out} down to V_{ref} before feeding it into the error amplifier. If you want to change V_{out} considerably you would need to change the division ratio.

The disadvantage is that this also changes the gain of the control loop. This in turn means that the performance and stability change with output voltage. By using a reference voltage equal to V_{out}, as I have done, the stability and performance of the circuit does not change with output voltage.

The error amplifier is a simple circuit rather than a high-performance opamp. Opamp-based very high gain circuits come with their own issues regarding stability and compensation, for instance. In fact the circuit is already pretty high performance, as shown in **Figures 2** and **3**. One reason for the good performance is that the load resistor for the error amp is the resistor from V_{drive} to ground, R5. Using a tube as a pass device, this resistor forms the grid resistor, which should be around 500 kΩ, depending on the device. So, even a very small error voltage between the base and emitter of T2 will lead to a relatively large 'correction' current through R5, and thus to a quite large correction voltage at V_{drive}. This loop gain is pretty large notwithstanding the simple circuit.

Practical circuit

This being a tube regulator, there are a few other things to take care of. One is the delayed application of the anode voltage, not just for the pass device but also for the amp to be powered. The full circuit (without the pass device) is shown in **Figure 4**. IC1 is a standard 555 (CMOS) timer that pulls down the LED in IC3 some time after the supply is switched on. The delay is set by R8 and C3 and with the given values is about 30 seconds. Once the LED in IC3 is on, the opto-triac will fire and switch on thyristor TH1, which will apply the rectified high-voltage to the pass tube.

I used the heater (filament) voltage to power the reference and control circuits through rectifier B1 and capacitor C2.

An important factor is the 'purity' of the reference voltage. If there is any hum or noise on the reference, it will be duplicated 'big time' by the error amp on the output. Current source T1 gets its reference V_{BE} through LED D5. With R1 at 1.3 kΩ, the reference current is almost 1.00 mA which makes the selection of the reference resistor R3 easy: just 1 kΩ for every volt of HV output you want. LED D5 acts as a voltage source, and to improve further on that, it is biased by constant-current regulator IC2, an LM334.

There are some other parts on the schematic for protection purposes. Fuse F1 protects the high voltage. Next, there's protection for the transistors. For example, if the high voltage is switched on by TH1, the reference voltage at the

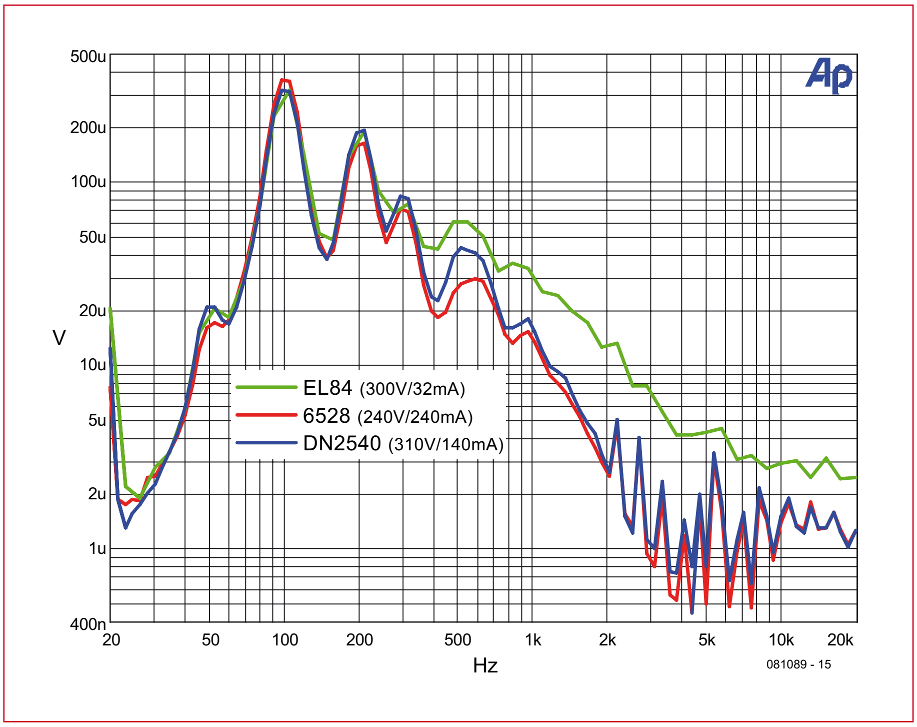


Figure 3. T-Reg output hum & noise plots.

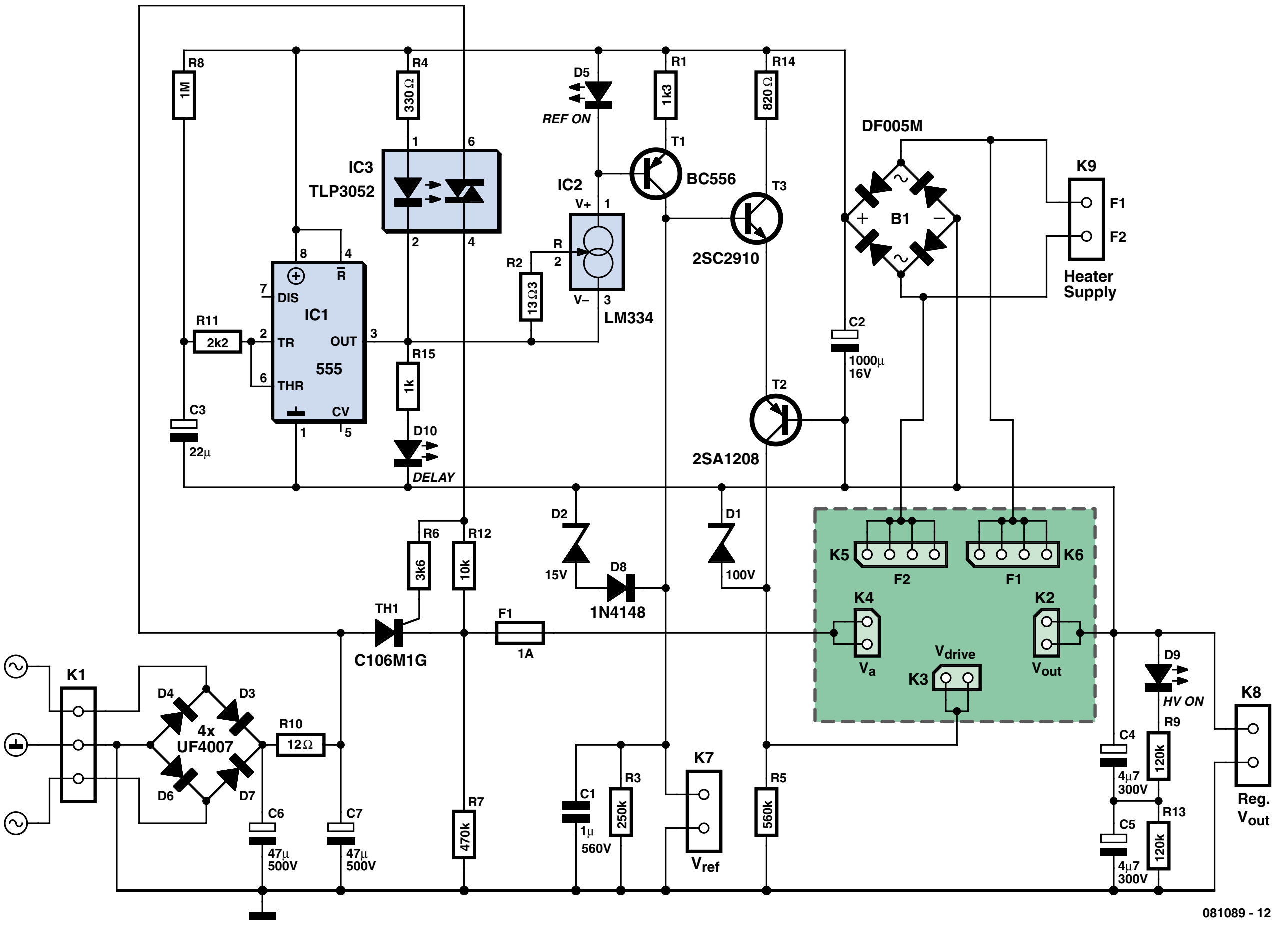


Figure 4. Complete circuit diagram of the main board.

Obsolete?

The idea for this project came to mind when I was working on a defective 1970's Lambda 250 V lab power supply. After spending a few hours on it, I found the problem: a defective voltage regulator IC, a 14-pin Lambda type FBT-00031. Tech support at Lambda kindly provided a single spec sheet for the chip. It took me several more weeks until someone on the Internet identified the chip as a re-branded Motorola MC1466 (figure 1). Obsolete, of course!

I found two NOS sources: one Far East at the tune of $ 280 each with a minimum order of 4 (!), the other in nearby Germany for 6 euros each. I ordered a couple, replaced the chip and the supply worked perfectly. I found the conceptual approach of this chip, with a floating error amp and reference setting with a single resistor so intriguing though, that I decided to see if I could apply it to a high-voltage valve regulator. This article is the result.

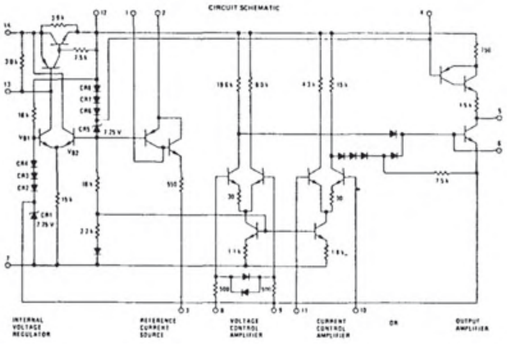

CIRCUIT SCHEMATIC
INTERNAL VOLTAGE REGULATOR
REFERENCE CURRENT SOURCE
VOLTAGE CONTROL AMPLIFIER
CURRENT CONTROL AMPLIFIER
OUTPUT AMPLIFIER

Figure 1. MC1466 simplified circuit.

So far so good. Then I received an email-copy of the June 2008 issue of LT Magazine. It featured the new LT3080 Linear Regulator (figure 2) that looks conceptually very much like an MC1466 that was declared obsolete more than a decade ago. What's more, this chip was selected by EDN as Innovation of the Year, while LT's Chief Technical Officer Bob Dobkin was named Innovator of the Year. I guess one should be very careful with what's declared obsolete...

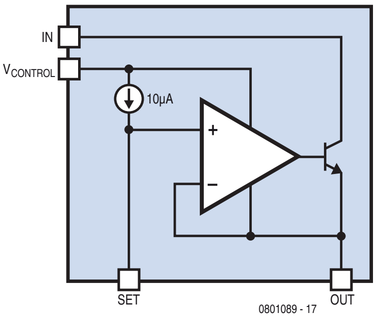


Figure 2. LT3080 simplified circuit diagram.

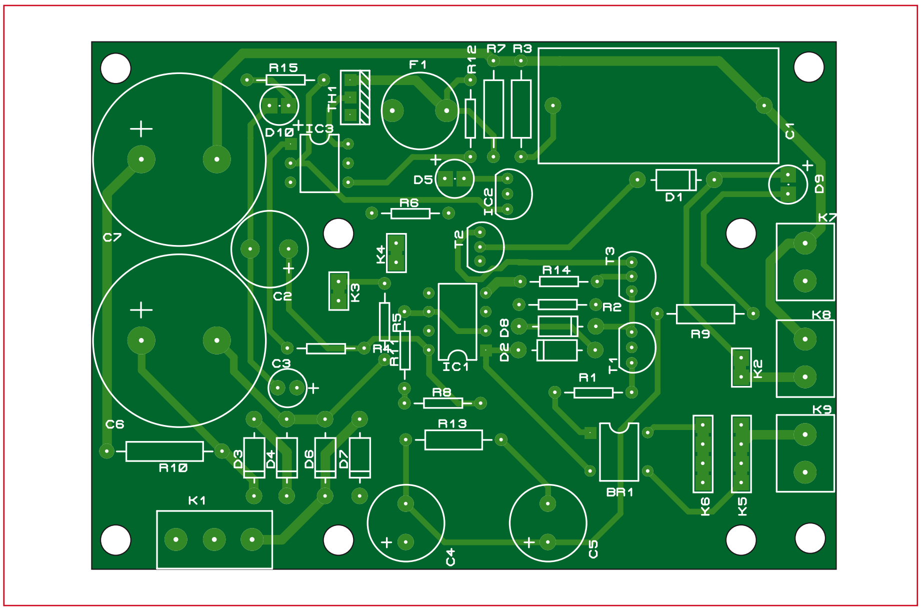


Figure 5. PCB stuffing plan for the motherboard. The copper track layout is a free download from the Elektor website.

COMPONENT LIST
main board

Resistors
R1 = 1kΩ3
R2 = 13Ω3
R3 = 250kΩ (see text)
R4 = 330Ω
R5 = 560kΩ
R6 = 3kΩ6
R7 = 470kΩ
R8 = 1MΩ
R9,R13 = 120kΩ
R10 = 12Ω (see text)
R11 = 2kΩ2
R12 = 10kΩ
R14 = 820Ω
R15 = 1kΩ

Capacitors
C1 = 1μF 560V MKT
C2 = 1000 μF 16V
C3 = 22 μF 15V
C4,C5 = 4μF7 300V
C6,C7 = 47μF 500V

Semiconductors
IC1 = LMC555
IC2 = LM334
TH1 = C106M1G (thyristor for 600V 2.5A)
IC3 = TLP3052
T1 = BC556
T2 = 2SA1208
T3 = 2SC2910
D1 = 100V 5W zener diode
D2 = 15V 0.5W zener diode
D8 = 1N4148
D3,D4,D6,D7 = UF4007 (fast/soft recovery rectifier 1000V 1A)
D5 = LED, yellow, 5mm (.2”) pitch
D9 = LED, red, 5mm (.2”) pitch
D10 = LED green, 5mm (.2”) pitch
B1 = DF005M (bridge rectifier 50V/1A)

Miscellaneous
F1 = 1A fuse, round, plug-in
K1 = 3-way PCB screw terminal block, lead pitch 5mm (.2”)
K2,K3,K4 = 2-way SIL socket strip
K5,K6 = 4-way SIL socket strip
K7,K8,K9 = 2-way PCB screw terminal block, lead pitch 5 mm (.2”)
DIL-06 IC socket
DIL-08 IC socket
4 standoffs, h=12mm (.5”)
4 machine screws for standoffs
PCB, ref. 081089-1 from www.thepcbshop.com

collector of T1 needs time to ramp up by charging C1. D2/D8 limits V_{CE} for T1 to a safe value during start-up. Protection for T2 is also required for the same reasons: V_{CE} of T2 is limited to 100 V by D1. This means that V_{drive} can be up to 100 V negative with respect to V_{out} which is the cathode voltage of the pass tube. This value should be sufficient even with low-*mu* triodes and low load current. R14 limits the maximum current through the small-signal transistors during start-up and transient conditions.

Configurations

Since this supply is pretty universal as regards its output voltage, I also tried to make it flexible as to the current it can supply. The rectifier, reference and control circuits fit on a ‘motherboard’ type of PC board (**Figure 5**). Then there is a small plug-in board that carries the pass device and some resistors. This is plugged on to the motherboard. This way, you can use the regulator with different pass devices depending on your load current. Currently, I have three plug-in -boards (see head illustration): for power amps, the plug-in board with a 6528 dual triode tube can deliver up to 600 mA (with a suitable power transformer, of course).

A type 6080 tube would also fit on this board. Due to its lower *mu*, it has a slightly higher Z_{out} and slightly higher hum and noise, but if you happen to have one of those lying around it can be used with good results. The plug-in fitted in the picture is a board with an EL84/EL86 (6BQ5/6CW5) type triode-connected small pentode for preamps and similar loads, up to 50 mA. Lastly, for those who want a ‘green’ version and save 10 watts or so on the pass tube heater, there is a board with a depletion-mode MOSFET type DN2540. This plug-in-board can accommodate two DN2540s and their heat sinks for an output load of more than 1 A, depending on the dissipation from the V_{DS} drop.

The circuits for the plug-in boards are shown in **Figure 6**. Since the 6528 (6080) is a dual triode, there are small value cathode resistors to equalise the currents at higher loads. The same is used on the DN2540 board in case you use two devices for increased load capacity (you can short out the source resistor if you use only a single DN2540). The plug-in -boards also have the grid and gate stopper resistors to aid stability. The DN2540 board has an additional 100 V/5 W zener diode (D5/D6) to protect the MOSFET during switch-on and load transients as well as V_{GS} protection zener diodes. It will also accept a pair of standard heat sinks. The boards are connected with standard 0.1-inch (2.5 mm) pitch PCB connectors and headers cut to size. Note that when you use the MOSFET pass device you don’t need the heater connectors between the two boards. M3 machine screws and 12 mm stand-offs secure the plug-in to the motherboard.

Performance

Figures 2 and 3 show the output impedance and output hum and noise of the various versions over the audio band. Hum and noise under load is typically below 500 μV_{RMS} in a 20 kHz bandwidth for all versions. The DN2540 is the best performer because of its high transconductance, and it has another advantage: a lower drop-out voltage. The DN2540 works fine with just 10 or 15 V V_{DS} (depending on the input ripple), while the tube versions require several tens of volts between anode and cathode to operate cleanly. This means that for a given transformer the DN2540 gives you 10 to 15 V more DC output and proportionally less dissipation. But there's more to it than just measurements. You may prefer a tube regulator in a tube amplifier for perfectly sound reasons. The choice is yours!

Figure 7 shows the component stuffing plans for the plug-ins.

Customisation

If you need an output voltage in excess of 500 V, you can use higher rated diodes for D3, 4, 6, 7 and higher voltage caps for C6, C7. Note that the diodes should be rated for at least twice the rectified peak voltage, preferably more. You can use a transformer with a single secondary connected to K1 pins 1 and 3, or a centre- tapped secondary with the tap at K1 pin 2. In the latter case you can omit diodes D4 and D6, of course.

The board has provisions for a small resistor between the two reservoir capacitors (R10) to take the high-frequency edge off the ripple voltage. The 12 Ω resistance is probably a bit too high for load currents above 100 mA, so you should experiment with it. Lowering the ripple on the rectified voltage will however not bring a proportional improvement on V_{out} because then the ripple on V_{ref} will start to dominate. The upshot is that the performance is pretty well balanced.

You set V_{ref} by selecting R3. The reference current being very nearly 1 mA, V_{ref} (in volts) = R3 (in kilo-ohms). You can also use an external potentiometer at K7 to set the output voltage, but make sure that it is rated for 500 V. The minimum V_{out} depends on the negative grid/gate drive voltage needed by the pass device; if you make V_{ref} = 0 by shorting R3, the DN2540 gets pretty close to zero as well, but a 6528 will not go lower than a few volts or even tens of volts (depending on the load) because of the necessary negative grid bias.

Construction

This supply contains several points carrying lethal voltages and currents, including the heat sinks for the DN2540s! You are well advised to observe electrical safety precautions and if necessary seek the assistance of an old hand at tube electronics. Double check that the high-voltage input is disconnected or turned off before doing anything on the board. Carefully discharge the high-voltage capacitors with a 1 kΩ resistor. To avoid body currents make it a practice to keep one hand in your pocket when touching live boards with a meter probe or any-

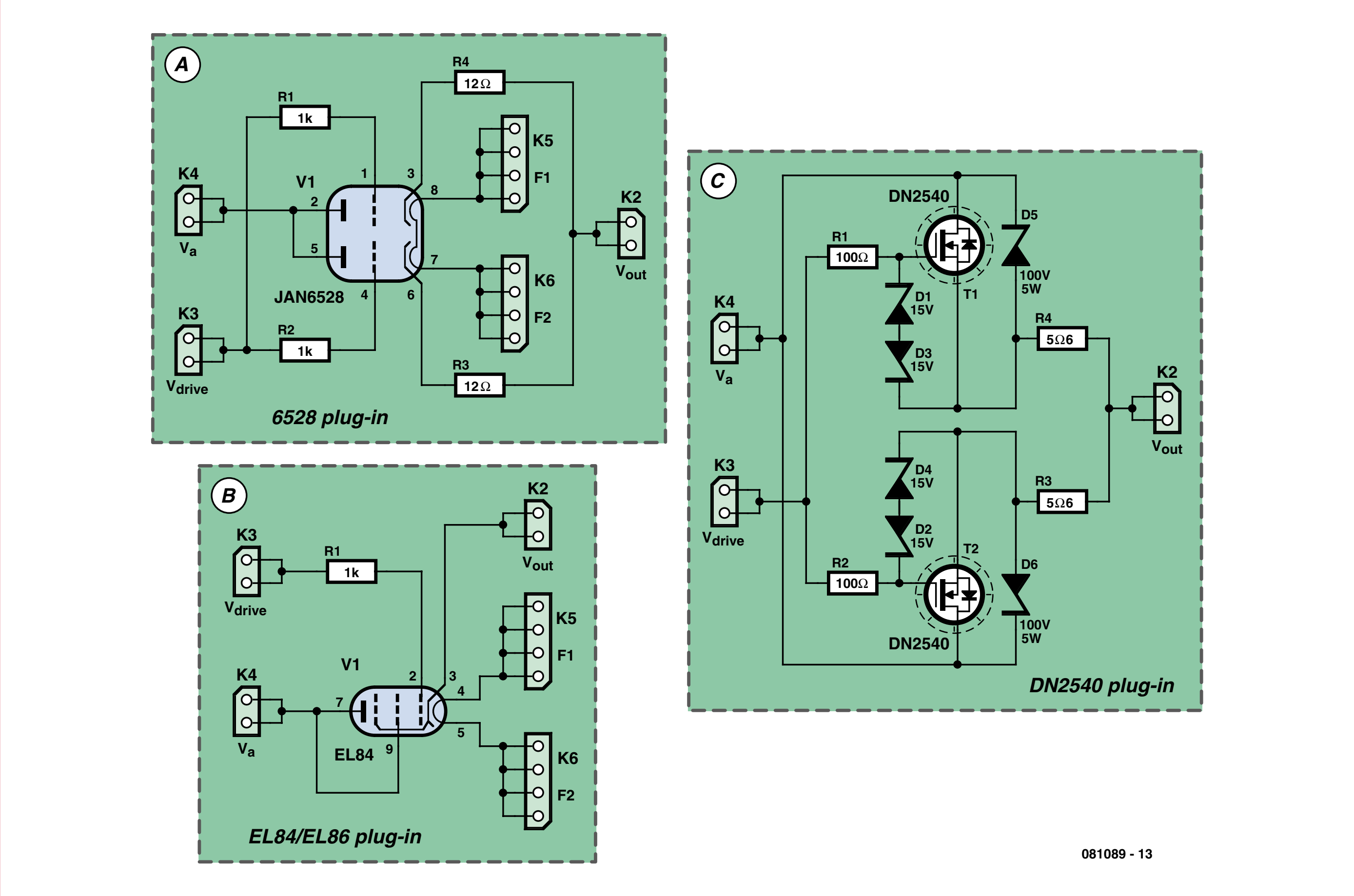


Figure 6. Three different plug-in board/pass device circuits.

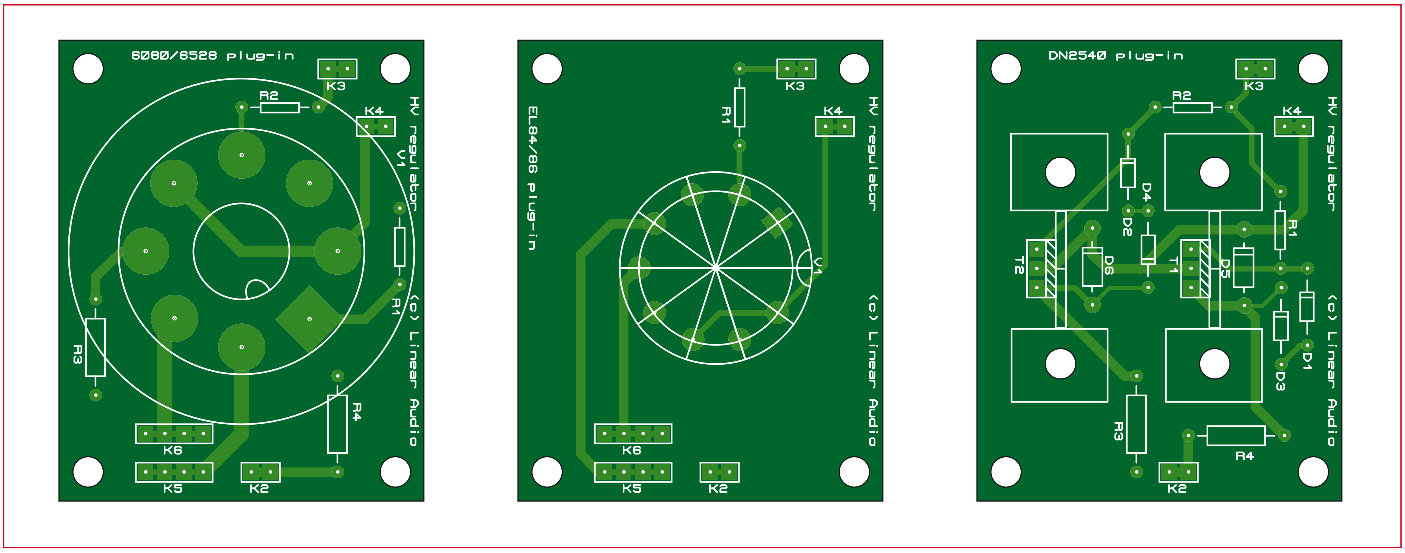


Figure 7. PCB stuffing plans for the plug-ins. The copper track layouts are free downloads from the Elektor website.

COMPONENT LIST
6528 plug-in

R1,R2 = 1kΩ 0.5W, 1%
R3,R4 = 12Ω 2W, 1%
V1 = 6528 or 6080
K2,K3,K4 = 2-way pinheader
K5,K6 = 4-way pinheader
PCB-mount Octal socket
PCB, ref. 081089-2 from
www.thepcbshop.com

COMPONENT LIST
EL84/6BQ5 plug-in

R1 = 1kΩ 0.5W, 1%
V1 = EL84 (6BQ5) or EL86 (6CW5)
K2,K3,K4 = 2-way pinheader
K5,K6 = 4-way pinheader
PCB-mount Noval socket
PCB, ref. 081089-3 from
www.thepcbshop.com

COMPONENT LIST
DN2540 plug-in

R1,R2 = 100Ω 0.5W, 1%
R3,R4 = 5Ω6 2W, 1%
D1-D4 = 15V 0.5W zener diode
D5,D6 = 100V 5W zener diode
T1,T2 = DN2540
K2,K3,K4 = 2-way pinheader
Heat sink(s) 4 K/W, e.g. Wakefield
637-20ABPE
PCB, ref. 081089-4 from
www.thepcbshop.com

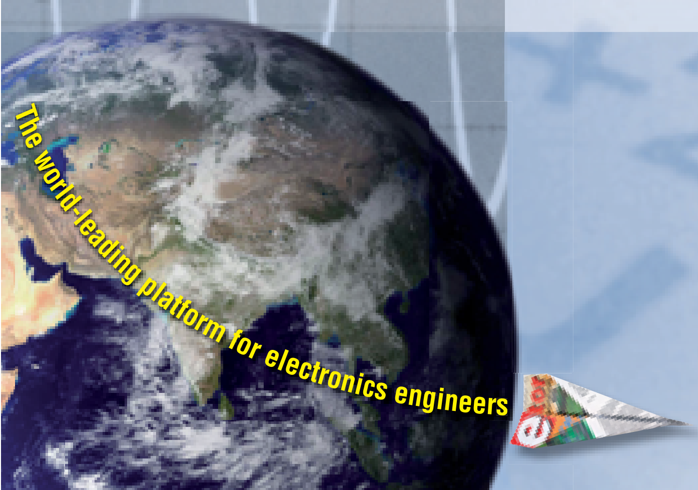
thing at all. Even when switched off, the high-voltage capacitors may still have a dangerous charge.

Caution: the heater supply for the pass tube is also used for the reference and control circuits, and floats on the regulated high-voltage. Do **not** use this heater winding for any other tubes in the equipment. If you use the DN2540 plug-in board, you still need a separate floating supply for the reference and control circuits. You could also use a small separate 5 or 6 V_{AC} transformer, provided it has enough secondary insulation to allow it to float to up to the maximum V_{out}.

The best is to start with the small parts on the motherboard, leaving the larger caps for last. Take care to mount the pins and headers straight upright so they will fit easily. The transistors are below the plug-in board and because of the height limitation, should be folded horizontally backward on the board. The same should be done with the output electrolytics C4 and C5. You can test the motherboard without a plug-in, to verify that the delay and thyristor circuits work. LED D10 should light when switched on, while D5 should come on after the delay time. (D9 will not come on unless you have a pass device plugged in). The input voltage at K4 should also come on after the delay. Next, populate one of the plug-in boards, mount it on the mother board and check out the whole assembly.

You can contact me via my website www.linearaudio.nl; I will also post any additions and tips there. Happy building and happy listening!

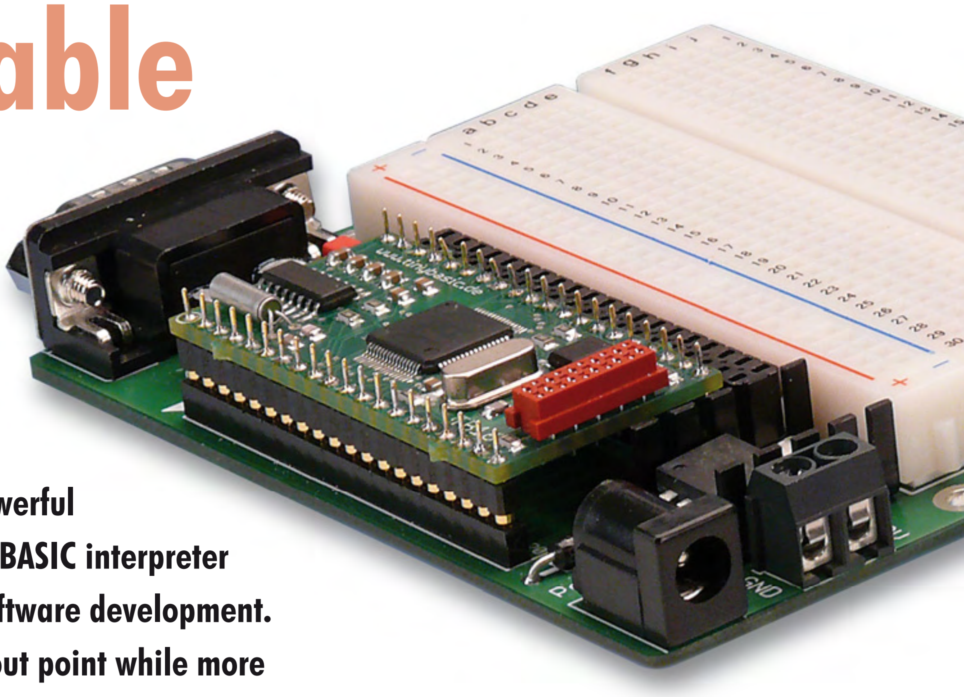
(081089-I)


The world-leading platform for electronics engineers

M16C TinyBrick

Programmable in BASIC

Dr. Uwe Altenburg (Germany)



A TinyBrick is a small self-contained microcontroller module fitted with a powerful Renesas 16-bit M16C microcontroller. A BASIC interpreter is installed in the module to simplify software development. Beginners will find it an ideal starting out point while more experienced users will appreciate its power and convenience. An evaluation board is introduced here; together with a TinyBrick we build an intruder alarm that sends SMS texts.

Quick project specs

TinyBrick processor board:

- Renesas M16C29 microcontroller
- 128+4 kB Flash, 12 kB RAM
- 1x RS485/RS232 (TTL)
- 1x RS232 (V24 swing)
- 1x ISP interface
- 8x PWM (16 bit)
- 8x ADC (10 bit)
- 2x INT, 1 x SPI, 1 x I²C

Kit of parts available, including SMD-prestuffed board

Microcontroller preloaded with BASIC interpreter

PCB available for evaluation board

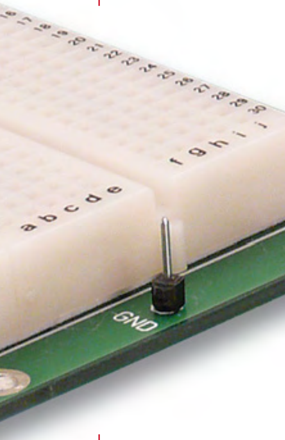
It was way back in the nineteen eighties when Intel programmed a control-oriented version of a BASIC interpreter into its 8052 microcontroller. Elektor was the first to do practical things with it. The chip was called 8052-AH-BASIC and the interpreter known as BASIC-52 became legendary as a result of Elektor publications. Many independent companies quickly recognized the chip's potential, among them Micromint who incorporated it together with the necessary RAM, communication interfaces and supplementary logic into their Domino modules [1]. By the middle of the nineties the company Parallax [2] had introduced the BASIC Stamp module which uses either the PIC microcontroller or Parallax SX chip.

While the 8052 BASIC concept was originally aimed at the professional user, the BASIC Stamp (despite its limited memory) quickly found many friends within the semi-professional and enthusiasts community. The success of these modules encouraged many other companies to produce equivalent devices. One popular example is the C-Control range of controllers produced by Conrad Electronics [3] in Germany.

The TinyBrick16 module used here can be seen as an addition to this list but has the advantage of a powerful 16-bit microcontroller. The M16C is a bigger brother of the popular R8C [4] which should be well known to regular Elektor readers. A TinyBasic interpreter [5] is preinstalled in the controller making a simple and compact environment ideal for developing control solutions to problems commonly encountered in the fields of measurement, control, regulation and robotics. Regular Elektor readers will already be aware of TinyBasic which was used in the TinyDisplay project featured in the May 2008 edition [6]. In the same way that the TinyDisplay PCB for that project was made available from the Elektor Shop we are also offering the TinyBrick module for this project which will be supplied with the controller and all other SMD components already mounted. Other components such as the connector and crystal are supplied in the kit.

Circuitry

The TinyBrick circuit diagram, PCB and pin layout is shown in **Figures 1, 2** and **3**. At the heart of the module is the M16C29 (M30291FC) microcontrol-



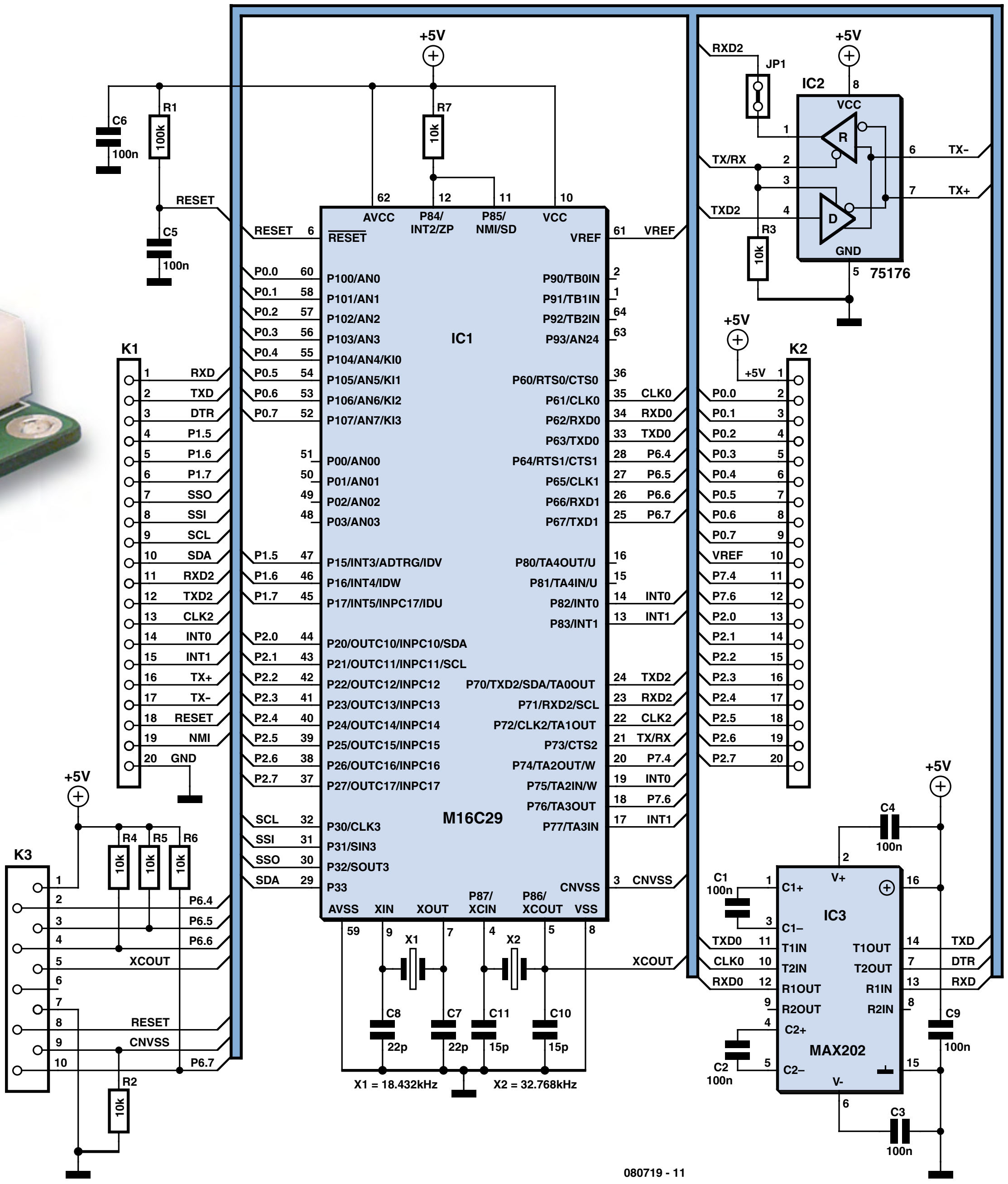


Figure 1. The TinyBrick module circuit diagram.

ler from Renesas. The M16C29 is available in either an 80 or 64-pin LQFP package. The TinyBrick PCB space constraints allow only the 64 pin variant to be used. Any external circuitry required by the M16C is kept to an absolute minimum. R1 and C5 form the reset network while C6 is the supply decoupling capacitor. An 18.432 MHz crystal provides the system clock; this frequency can be easily divided down internally to generate the standard baud rates for serial port communication (300 to 115,200 Baud). The chip can be clocked at 20 MHz maximum, so the crystal used does not represent too much of a reduction in processing power. A 32.768 kHz watch crystal is also fitted and can be used by a counter to generate an exact one second clock which is useful for real-time clock applications. Both crystals are fitted with appropriate load capacitors. The watch crystal oscillating frequency is particularly dependant on the value of its load capacitors (C10 and C11) and can deviate a few ppm from its nominal value. To produce a more accurate clock the seconds count can be periodically corrected in software by synchronizing to a reference radio clock source. The TinyBasic interpreter supports this function.

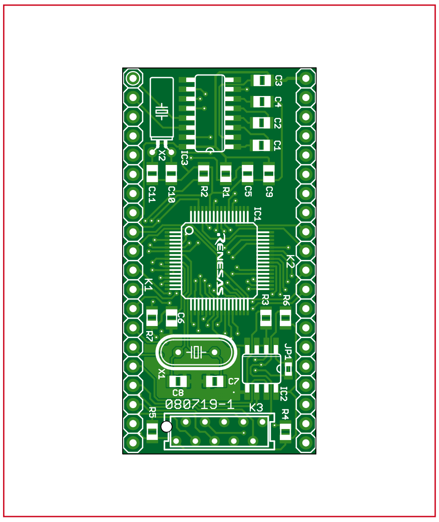


Figure 2. The TinyBrick module PCB.

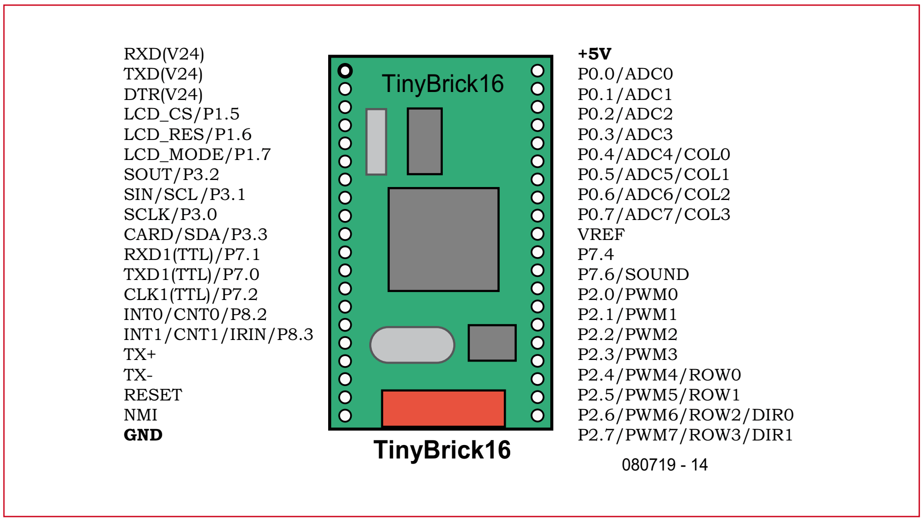


Figure 3. The TinyBrick pinouts.

COMPONENT LIST

TinyBrick module

Resistors
R2-R7 = 10kΩ
R1 = 100kΩ

Capacitors
C10,C11 = 15pF
C7,C8 = 22pF
C1-C6,C9 = 100nF

Semiconductors
IC1 = M30291FC-U5
IC2 = SN75176
IC3 = MAX202CSE

Miscellaneous
X1 = 18.432 MHz quartz crystal
X2 = 32.768 kHz quartz crystal
K1,K2 = 20-way SIL pinheader with turned pins, 2.54mm / 0.1” lead pitch
K3 = 10-way Micromatch socket
JP1 = jumper

PCB # **080719-1** [8]

PCB, ready stuffed with microcontroller* and SMD components; all other parts included. Order code **080719-91** [8]

* programmed with bootloader and BASIC interpreter.

Interfacing

The M16C29 contains three universal asynchronous interfaces (UARTs). The first of these serial interfaces using pins P6.1 to P6.3 are connected to a MAX202 (IC3) and then on to pins 1 to 3 of the pin-strip connector K1 to provide V24 serial communication signal levels. This port is used to download BASIC applications. BASIC routines can also use this port for example to send/receive messages or interface to a serial display or modem. For this last application the DTR signal is also available along with TXD and RXD.

The second interface uses port pins P6.4 to P6.7 together with control signals ($\overline{\text{RESET}}$, CNVSS) on the 10-pin Micro-Match connector. This port is used for In System Programming (ISP). When an E8 emulator together with a HEW development system (from Renesas) is used it is possible to download, run and debug C programs. Even without an E8 an additional programming adapter has been developed for fans of C and will be presented soon in Elektor — we have already reserved the url: [7].

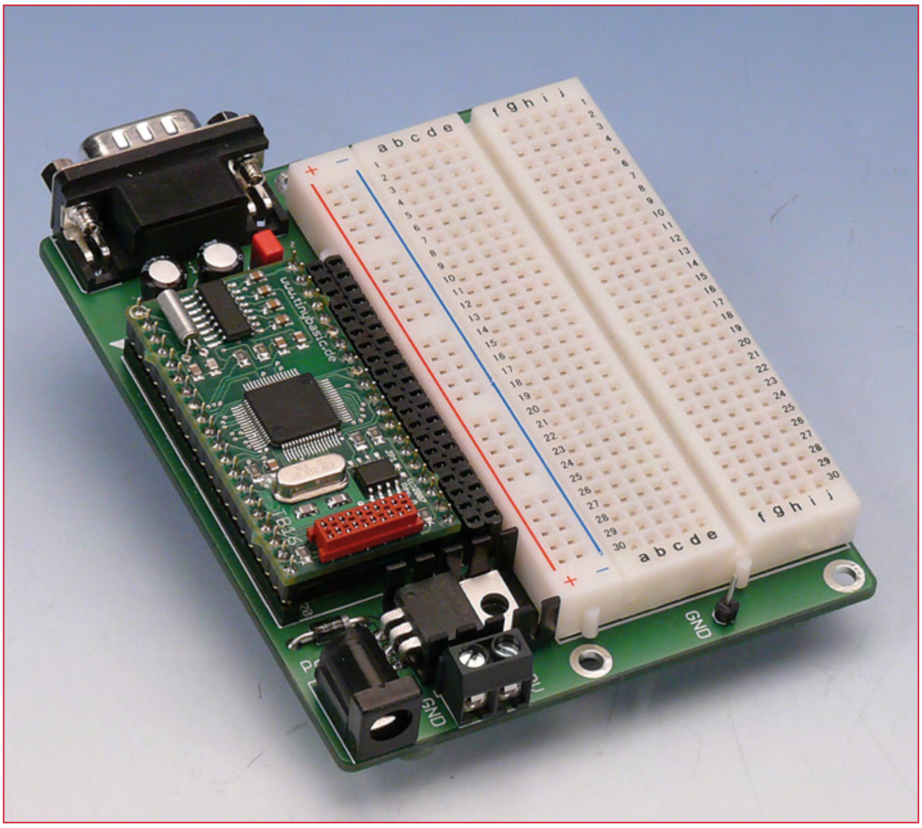


Figure 4. The TinyBrick evaluation board.

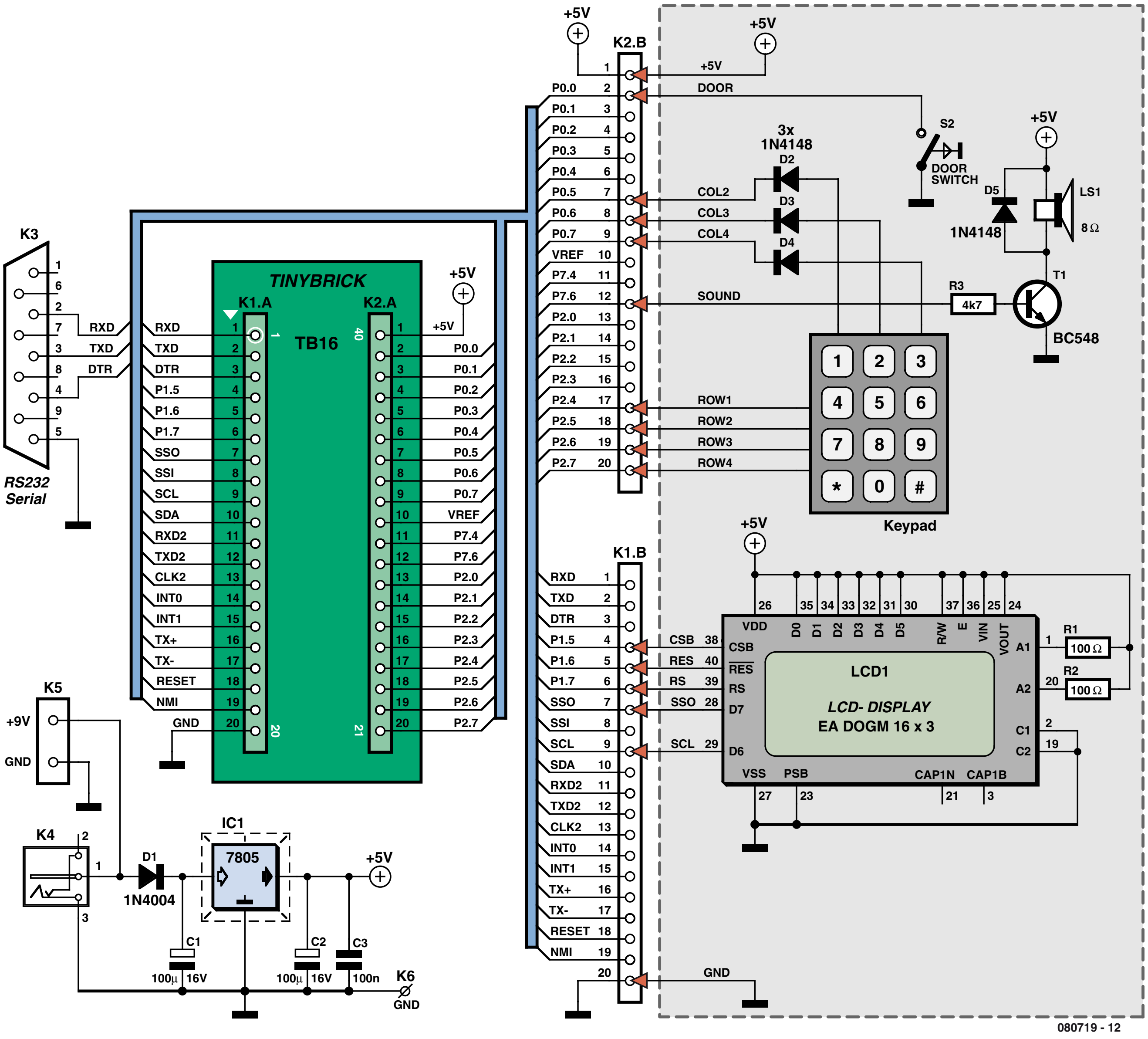


Figure 5. Circuit of the TinyBrick evaluation board (The example application is shown in grey underlay).

The third interface has two functions. Firstly it sends and receives signals directly on the pin strip connector K1 at TTL signal levels which is commonly used by equipment such as GPS receivers or radio time signal receivers. The signals are also connected to a SN75176 differential line driver (IC2) to provide TX+ and TX– for an RS485 interface on K1. This allows several TinyBricks to be bussed together to form a network, a useful feature for applications such as home automation. TinyBasic includes support for a simple network protocol.

Evaluation Board

The minimum circuitry required to run a TinyBrick module is a regulated 5 V power supply (mains adapter or bench power supply) and a serial interface cable. The connections can be made by plugging the module into a prototyping pin board or using a piece of perforated board. A simpler solution is to order the TinyBrick evaluation PCB available from the Elektor Shop; this produces a much neater job. The PCB requires very few components to be fitted and no special soldering skills are required. The evaluation board is shown in **Figure 4**, it provides connectors for an external power supply and a serial communications cable.

The serial port uses a standard 9-pin sub-D connector. This simplifies the task of attaching peripherals such as

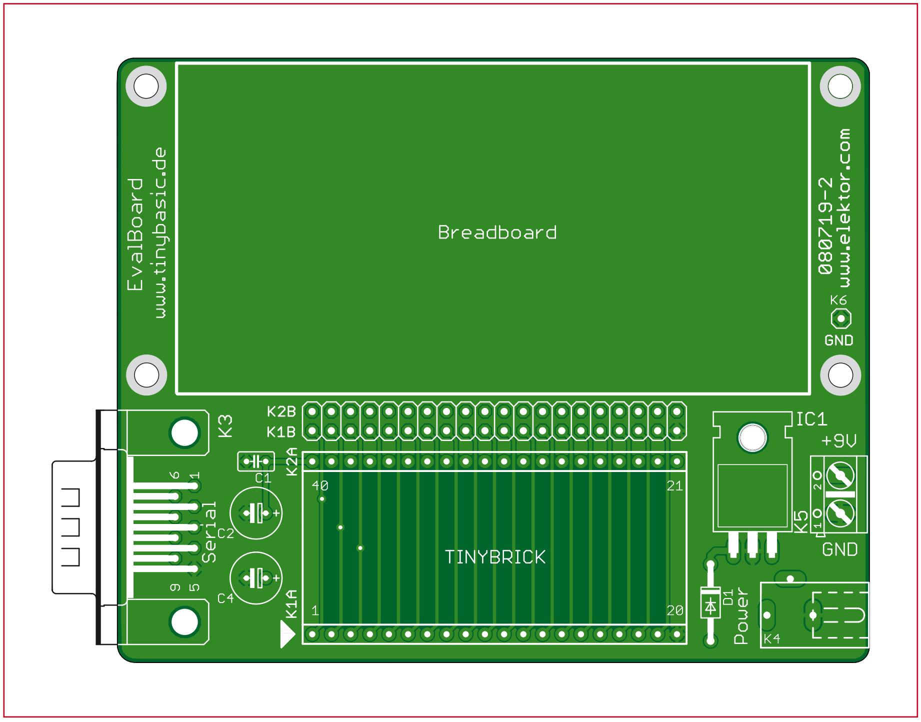


Figure 6. TinyBrick evaluation board PCB.

COMPONENT LIST

Evaluation board

Capacitors

C1 = 100nF
C2,C4 = 100µF 16V

Semiconductors

IC1 = LM2940-5.0
D1 = 1N4004

Miscellaneous

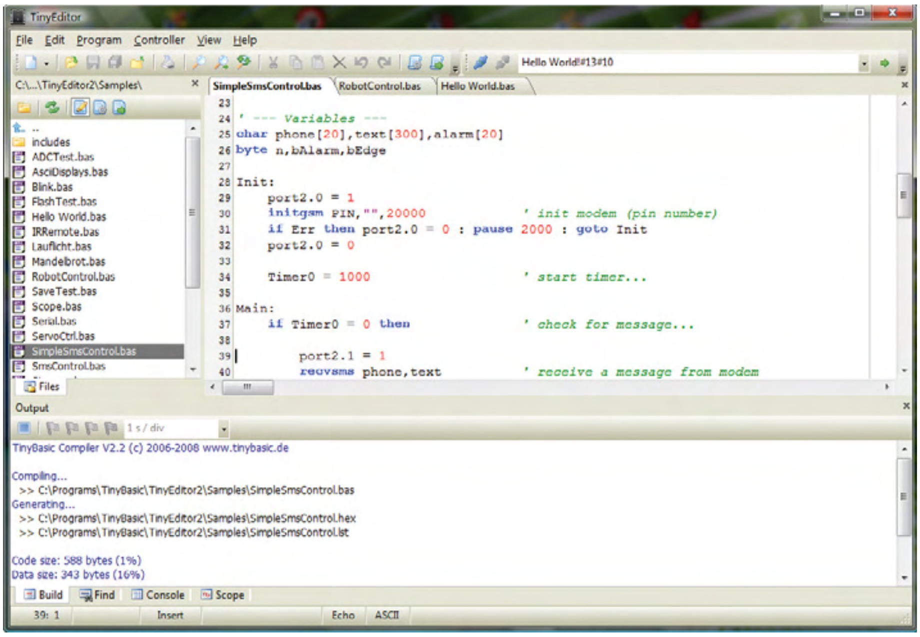
K1, K2 = 20-way SIL socket strip, 2.54mm / 0.1” lead pitch
K3 = 9-way sub-D socket (female)
K4 = PCB socket for power line adaptor
K5 = 2-way PCB screw terminal block, lead pitch 5mm (.2”)
K6 = solder pins
2 x 20-way socket for Tinybrick module (2.54mm / 0.1” lead pitch)
1x protoboard (ca. 84 x 54 mm / 3.3 x 2.1”)
1x heatsink type SA220
PCB # **080719-2** [8]

modems or displays to the evaluation board. A null-modem cable is required to download programs but if you do not have access to a PC with a serial port it will be necessary to make a ‘null-modem to USB’ cable (use a USB to TTL cable together with some RS232 level shifters).

Construction details of such a cable are available on the Elektor website [8].

Software

As mentioned earlier TinyBricks are primarily programmed in TinyBasic [5], which is one of the easier languages for beginners to pick up. Experienced developers will also find this system has much to offer allowing small tasks to be completed in a very short time. The M16C29 is equipped with 128 kB Flash, 12 kB RAM and 4 kB of data flash memory, 64 kB is reserved for the boot loader and TinyBasic interpreter. The current version of TinyBasic uses around 40 kB so there is ample space left for future expansion. 32 kB is available for use by the Basic program. The program is stored in the form of tokens so this should be more than enough. The remaining 32 kB is free to be used as necessary, for example to store data. Lastly there is 4 kB of data flash memory in which initialisation values and other parameters are stored. The data Flash memory can be erased and reprogrammed up to 10,000 times while the normal flash memory is good for up to around 1,000 erase/reprogram cycles. The Bootloader is called immediately after a system reset. If no download is detected within the first 2 s the TinyBasic interpreter begins executing the most recently stored BASIC program. The bootloader can also be used to load the resident TinyBasic interpreter; this ensures that any future versions of the interpreter with newer features can still be used by the current TinyBrick. While the TinyBrick is executing a BASIC program a system reset can be initiated by sending a reset sequence over the first serial interface port. Once this is received a software reset is generated and the bootloader begins execution again. This allows a program download to be initiated at any point.



```
' --- Variables ---
char phone[20],text[300],alarm[20]
byte n,bAlarm,bEdge

Init:
    port2.0 = 1
    initgsm PIN,"",20000              ' init modem (pin number)
    if Err then port2.0 = 0 : pause 2000 : goto Init
    port2.0 = 0

    Timer0 = 1000                     ' start timer...

Main:
    if Timer0 = 0 then                ' check for message...

        port2.1 = 1
        recvsms phone,text            ' receive a message from modem
```

```
TinyBasic Compiler V2.2 (c) 2006-2008 www.tinybasic.de

Compiling...
 >> C:\Programs\TinyBasic\TinyEditor2\Samples\SimpleSmsControl.bas
Generating...
 >> C:\Programs\TinyBasic\TinyEditor2\Samples\SimpleSmsControl.hex
 >> C:\Programs\TinyBasic\TinyEditor2\Samples\SimpleSmsControl.lst

Code size: 588 bytes (1%)
Data size: 343 bytes (16%)
```

Figure 7. TinyEditor used to generate and download the TinyBasic program.

A freeware editor is useful for the development of TinyBasic programs (**Figure 7**) [5]. It contains all the tools necessary to enable programs to be written, compiled and downloaded to the TinyBrick. The compilation process detects and indicates the line number of any syntax errors. In addition it reduces the size of the finished code quite appreciably.

TinyBasic the basics

Many system variables are defined in TinyBasic. Using the variables port0 to port9 the status of the ports can be read and changed. It is often necessary to access just a single port pin. The line:

```
Port2.0 = not Port2.0
```

inverts the state of pin P2.0 each time it is executed. Variables should always be assigned meaningful names to help make the program listing more readable:

```
#define Motor   Port2.0

#define ON      1
```

allows us to turn on a motor connected to port pin P2.0 using the command:

```
Motor = ON
```

When the processor is reset it restarts with all of its port pins configured as (high impedance) inputs. This ensures that there are no signal conflicts with any externally connected hardware. Using the ‘setport’ instruction any of the port pins can be defined as an output. Pull-up resistors can also be connected to inputs pins under software command. As an example:

```
setport 2,%00000001,%11110000
```

will configure pins P2.4 to P2.7 as inputs with pull-ups, pins P2.1 to P2.3 as inputs without any pull-ups and pin P2.0 as an output.

One special feature of TinyBricks is its direct support for different types of text and graphic displays. The Display On Glass Module (DOG-M) from the company Electronic Assembly [9] offers a number of advantages over other displays, for example its use of an SPI interface means that it only uses five of the controllers port pins. Interfacing a graphic display module has already been covered in more detail in the previous article [6].

Only one command is required to indicate to the BASIC interpreter which type of display is fitted. This information determines which output command is used to write to the display i.e. move or print. In this example a three line text display with 16 characters per line is defined:

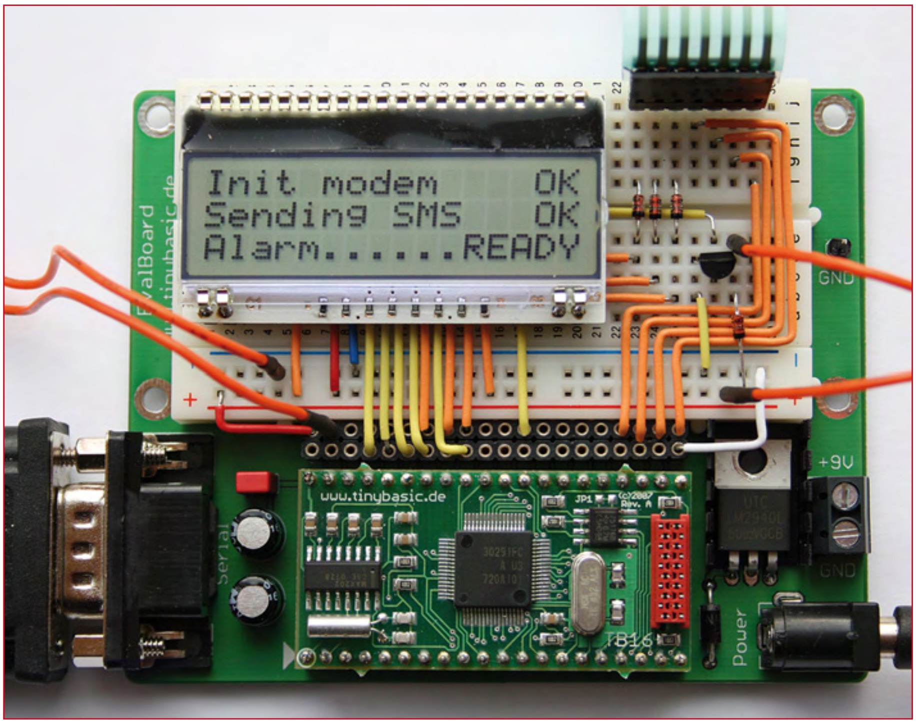


Figure 8. The small intruder alarm built using the evaluation board.

```
setdisplay LCD_DOGM16x3
```

This display incorporates its own character set so it is only necessary to send the code for each character to the display. When a graphic display is fitted it is necessary to send the pixel pattern for the character to the display. A graphic display also allows lines, circles and other symbols to be represented.

Another basic concept in TinyBasic is the difference between the five input and five output channels. Channels #0 and #1 are assigned to the first two serial interfaces, while channel #2 writes to the display and reads from the attached keypad. Channels #3 and #4 allow two independent files to be referenced on a SD card.

The following two-line example demonstrates how the channels are used to send the text string ‘Hello World!’ first to the serial interface and then to the display:

```
print #0, “Hello World!”   ‘
```

sends the text to the first serial interface, while

```
print #2, “Hello World!” ‘
```

sends the text to the display.

There are many different text features supported in the language such as `using()`, `chr()` and `spc()` which are useful for text formatting.

A first project

As an example application we will use the evaluation board fitted with a TinyBrick to build an intruder alarm (**Figure 8**). The proposed alarm is quite simple but should be sufficient to provide some security to the contents of, say, a garden shed. The alarm becomes armed once the door is closed and will be activated when it is opened. A limited time window allows the correct PIN code to be entered to disable the alarm before the break-in is detected. The alarm is fitted with a display and sends an SMS text to a predefined telephone number.

The grey underlay on the circuit diagram in **Figure 5** shows the components used. Characters are displayed on a three line ‘DOG’ display module from Electronic Assembly. A 3×4 foil matrix type keypad is used to enter the PIN code and a small microswitch is fitted behind the door strike plate to detect the door opening. The alarm sound can be produced by a 12 V siren but this is quite deafening in a small space and not really necessary because an cellphone SMS text is also sent at the same time, a small loudspeaker should therefore be sufficient.

To send the SMS texts over the cellphone (GSM) radio network it is necessary to use a GSM modem with a serial interface such as the basic TC35i from Siemens. The company Telit [9] also offer a more integrated solution with its GM862-QUAD modem with multiband GSM and GPRS capability. These stand-alone GSM modems all turn out to be relatively expensive. A better solution is to use an old mobile phone. Newer models are usually unsuitable because they are not fitted with a serial interface port. Some of the older types of phone also have the problem that the serial port is not available when the charger is attached and vice-versa.

The trusty old Nokia 6310i mobile phone is ideal for this application because it has a serial interface and a separate charging socket. The phone also supports ‘Text Mode’ for sending SMS texts. It is also necessary to use the Nokia DLR-3P data cable to connect to the phone. This is powered from the DTR signal so it is necessary for the program to switch this line to +12 V.

The complete source code for the project can be freely downloaded from the Elektor website [8]. A fragment of the code for the intruder alarm is shown in **Listing 1**. The program has been kept deliberately simple; for example the phone number to which the text is sent is defined in source code and can be changed by the text editor but there is no reason why the program cannot be modified so that one or more numbers entered from the keypad can be stored for example in the data Flash memory of the microcontroller.

I am sure you have thought of other applications where it would be useful to send or receive SMS texts on your mobile. The unit could for example be incorporated into a heating system controller to call the heating engineer when a fault occurs. A routine could also be written in TinyBasic to receive and decode SMS text messages and use the TinyBrick I/O capability to control processes appropriately.

(080719-I)

Listing 1. The intruder Alarm source code (fragment)

```
…

‛ --- Main loop ---
do
‛ The door was just locked...
   if (State = NONE) and (Door = LOCKED) then
        cls
        print #2,chr(0)                       ‛ print special char
        move 5,1
        print #2,”ARMED”
        State = ARMED
        pause 1000
   endif

   ‛ The door was just opened...
        if (State = ARMED) and (Door = OPENED) then
               cls
               print #2,”Enter Pin Code:”
        move 6,1
        PinOk  = 0
        PinNum = 0
        Timer  = T30SEC
        State  = WAITING
   endif

…


   ‛ Time elapsed...
   if (State = WAITING) and (Timer = ELAPSED) then
        cls
        print #2,”Init modem    “,
        initgsm 8034
   ‛ init gsm modem with pin code
        if Err = 0 then print #2,”OK”,

        print #2,”Sending SMS   “,
        sendsms “0174xxxxxxx”,”Burglary!”
   ‛ send sms with message text
        find “OK”,15000

        if Err = 0 then
               print #2,”OK”,
               print #2,”Alarm......”,

               for nAlarm = 0 to 50          ‛ alarm siren
                      for nFreq = 3000 to 2500 step -50
                             sound nFreq
                      next
               next

               print #2,”READY”,
        endif

        State = NONE
   endif

loop
```

Internet Links

[1] www.micromint.com

[2] www.parallax.com

[3] www.c-control.de

[4] www.elektor.com/r8c

[5] www.tinybasic.de

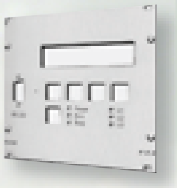
[6] www.elektor.com/070827

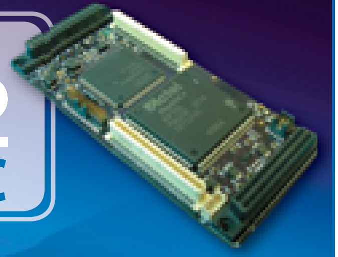
[7] www.elektor-usa.com/080422 (to be published)

[8] www.elektor-usa.com/080719

[9] www.lcd-module.de

[10] www.telit.com





Sinewaves in Progra

Direct digital synthesis using an Altera CPLD

Jean Christophe Humez (France)

Generating a stable, high-frequency signal from a crystal is simple, but you can't always get just the frequency you want. Of course, you can always order custom-made crystals, but not everyone can afford this and it's not necessarily terribly practical. A more flexible solution is direct digital synthesis (DDS). Ready-made DDS devices do exist, but are complicated to use. Here's a compromise between the flexibility of a DDS device and the inflexibility of a crystal: programmable logic.

Technical specifications

- DDS frequency synthesizer (Direct Digital Synthesizer)
- Sinewave output by virtue of the principle and the filtering
- Output frequency from 610 Hz to 28 MHz with a resolution of 610 Hz
- Possibility of performing FSK (Frequency Shift Keying) frequency modulation
- EPM7064SLC44-10 (or LC44-7) programmable logic IC from Altera
- 70 MHz operating frequency
- R-2R network DAC
- Output filtering using an LC circuit
- 5 V supply

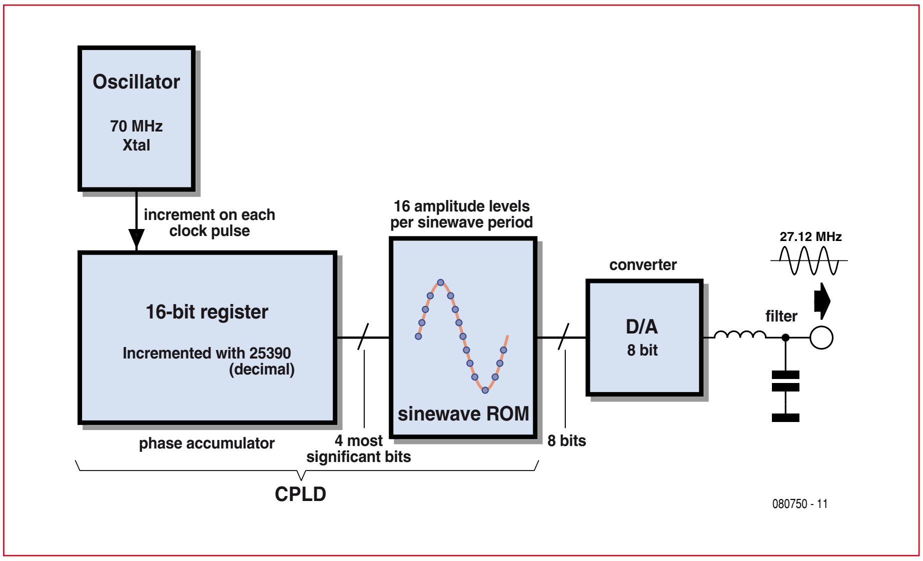


Figure 1. Block diagram of the DDS generator.

Programmable frequency generators have been published before in Elektor [1][2], but they used a special IC based on a phase locked loop (PLL).

Here we're going to apply a different principle to arrive at the same result. We're talking about circuits that synthesize sinusoidal signals directly (DDS — Direct Digital Synthesis), which allow us to generate specific frequencies with great precision. They are a digital solution in competition with solutions using PLLs. Integrated circuits do exist to achieve this function, but there aren't all that many of them around.

The AD98xx series from Analog Devices is a family of DDS devices with varying degrees of speed and accuracy, but these ICs are hard to implement, especially if you only need to generate a single frequency. They have to be configured using internal registers, and so need a microcontroller to work alongside. Also, they come in extremely small packages (28-pin SSOP for the AD9851), and fitting them is not always within the capacity of amateurs. In this article, we're suggesting a simple and effective solution based on the principle of DDS, but without the complexity of a DDS device, that lets us produce frequencies that are useful for radio applications like the 27 MHz Citizens' Band (CB). Our solution uses a type EPM7064SLC44-10 complex program-

mmable Logic

mable logic device (CPLD) from Altera. It goes without saying that this solution can be used with other FPGA programmable ICs, or even microcontrollers for generating lower frequencies.

Block diagram

The heart of a DDS system is a register called a phase register or phase accumulator. This register is incremented with a constant at the rate of a stable clock. When the phase register overflows, one period of the output signal is completed. The overflow is so arranged as to reset the register to a lower value and a new output signal period starts. In this way, the phase register behaves cyclically, and it is this cycle that generates the frequency of the output signal. This technique makes it possible to generate any frequency, with a resolution that depends on the size of the phase register.

A certain number of bits in the phase register are then used to index a table that contains the waveform to be generated — usually a sinewave. The larger this table is, the more accurate the output waveform will be.

In our system (see **Figure 1**) the phase register is 16 bits and is incremented on each rising edge of the 70 MHz master clock with a value of 25,390. In this way, the output frequency is 27.120 MHz. This increment is easy to calculate (with a 16-bit phase register = 2^{16} = 65,536):

$$increment = 65536 \times \frac{27.120\,MHz}{70\,MHz} = 25390$$

The four MSBs of the phase register address a memory containing 16 8-bit samples of one cycle of a sine-wave (Table 1). An 8-bit R-2R ladder network performs the digital-to-analog conversion. The signal is then smoothed by an LC filter, centered on 27 MHz, so as to obtain a clean sinewave.

Circuit in detail

Figure 2 shows the circuit of our oscillator in detail. A binary input E_

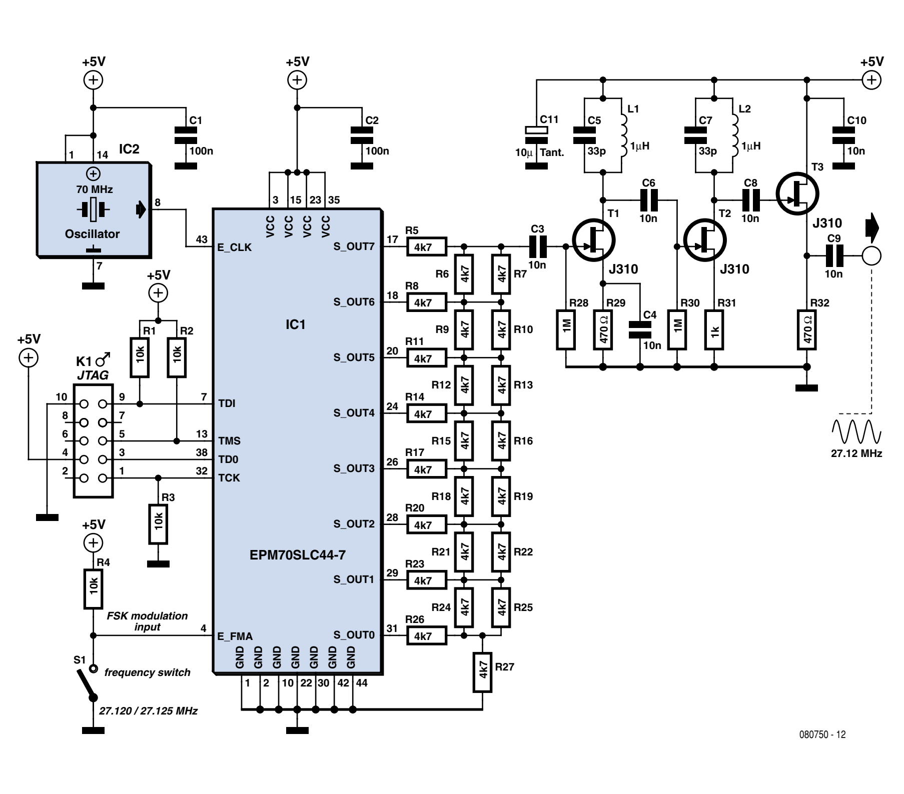


Figure 2. Circuit diagram of the CPLD based DDS oscillator.

Table 1. Contents of the sinewave 'ROM'

Input	Sine function in degrees	Output converted to period [0 – 255]
0000	sin(0)	127
0001	sin(22.5)	176
0010	sin(45)	217
0011	sin(67.5)	245
0100	sin(90)	255
0101	sin(112.5)	245
0110	sin(135)	217
0111	sin(157.5)	176
1000	sin(180)	127
1001	sin(202.5)	78
1010	sin(225)	37
1011	sin(247.5)	9
1100	sin(270)	0
1101	sin(292.5)	9
1110	sin(315)	37
1111	sin(337.5)	78

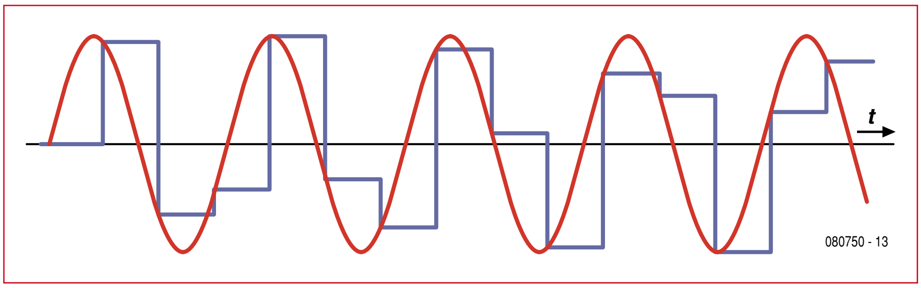


Figure 3. Example of the digital sinewave before (blue) and after (red) filtering.

FMA lets us choose between two frequencies, and so allows binary FSK (Frequency Shift Keying) frequency modulation.

The digital sinewave leaves the CPLD on an 8-bit port and a DAC is needed to convert it into an analog sinewave. To avoid having to use a special device, we are using a resistor network, referred to as ‘R-2R’ because of its structure. Each of the eight outputs generates a voltage which contributes to the total output voltage.

The DAC output signal doesn’t look much like a sinewave (see **Figure 3**) and a filter is needed to improve it. As we have on average only 2.6 samples per cycle, lots of unwanted harmonic frequencies are generated and need to be eliminated. This task is entrusted to two LC filters in series.

The filtering makes use of J310 JFET transistors, well-known in HF electronics. The LC filters are centered on 27 MHz, calculated using Thomson’s formula:

$$frequency = 1/(2\pi\sqrt{(LC)})$$

where L is the inductance and C the capacitance. With 33 pF for C and 1 μH for L, we get a frequency of 27 MHz.

The VHDL code

The CPLD is programmed in VHDL. The program breaks down into two processes.

Process 1: the phase accumulator

This counter operates with a recurrence of 27.120 MHz or 27.125 MHz.

Listing 1: VHDL program

The leading – indicates a comment

```
-- Direct Digital Synthesiser December 26, 2008
-- JC HUMEZ

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY sine_dds IS

  PORT – - definition of the CPLD external signals
  (
    -- CPLD inputs
    E_CLK : IN STD_LOGIC; -- 70MHz master clock
    E_FMA : IN STD_LOGIC; -- (frequency select FMA=0=>27.120MHz or FMA=1=>27.125MHz)
                          -- (to perform 0 or 1 FSK modulation)
    -- CPLD outputs
    S_OUT : out std_logic_vector(7 downto 0) – sinewave amplitude coded in 8 bits
  );

END sine_dds;

counter ARCHITECTURE OF sine_dds IS
  -- definition of internal CPLD signals
  SIGNAL SIG_CPT : std_logic_vector(15 downto 0); -- phase accumulator
  SIGNAL SIG_ESIN : std_logic_vector(3 downto 0); -- 4 MSBs of phase accumulator
  SIGNAL SIG_SSIN : out std_logic_vector(7 downto 0) – sinewave amplitude

BEGIN -- start of architecture
  --//-- start of phase accumulator process
  PROCESS (E_FMA,E_CLK)
  BEGIN
    IF (E_CLK’EVENT AND E_CLK = ‘1’) THEN
      IF E_FMA = ‘0’ THEN
        --here FMA0 27.120MHz -- formula = 65536/70MHz*27.120MHz = 25390
        SIG_CPT <= SIG_CPT + 25390;
      ELSE
        --here FMA1 27.125MHz -- formula = 65536/70MHz*27.125MHz = 25395
        SIG_CPT <= SIG_CPT + 25395;
      END IF;
```

This is a 16-bit register called SIG_CPT and hence its maximum value is 2^{16} = 65,536. This process is run on each rising edge of the 70 Hz signal, defined by the condition:

```
E_CLK’EVENT AND E_CLK = ‘1’
```

To achieve our repetition rate of 27.120 MHz, we need to add the whole number 25,390 on each rising edge of the 70 MHz:

```
SIG_CPT <= SIG_CPT + 25390
```

Next, the variable SIG_ESIN lets us isolate the four MSBs of SIG_CPT:

```
SIG_ESIN(3 downto 0) <= SIG_CPT(15 downto 12)
```

Process 2: the digital sinewave
The second process involves a small memory containing the eight bits of the digital sinewave, and it too runs at each rising edge of the 70 MHz. This process causes each of the 16 values of SIG_ESIN to correspond to one sample of the digital sinewave SIG_SSIN. This function is achieved using the instruction

```
CASE
…
WHEN
```

The digital sinewave samples are calculated like this:

$$Sample = 127.5 \times \sin(N\pi / 8) + 127.5$$

where *N* is from 0 to 15. **Table 1** gives the 16 values stored in the ‘ROM’.

Construction

A prototype was made on a piece of prototyping board with an earth plane. There are no adjustments. In **Figure 4** you can see the 70 MHz oscillator, the CPLD chip, the three JFET transistors with the inductors, the JTAG programming connector and a 5 V regulator for the power. On the back of the board (**Figure 4b**), the R-2R network is wired using SMD 0805 resistors.

Depending on the state (0 or 1) of the E_FMA input, we get one or the other of the two frequencies. This E_FMA input is intended for binary frequency modulation (FSK).

Figures 5a and **5b** show the circuit’s two output signals as displayed on a digital oscilloscope**.**

Programming other frequencies

The highest frequency we will be able to obtain is 40% of the value of

```
    END IF;

  END PROCESS;
  --//-- end of phase accumulator process

SIG_ESIN(3 downto 0) <= SIG_CPT(15 downto 12); -- only the 4 MSBs are retained

--//-- start of memory process
PROCESS (E_CLK)
BEGIN
  IF (E_CLK’EVENT AND E_CLK = ‘1’) THEN – clock rising edge
    CASE SIG_ESIN IS
      WHEN “0000” => SIG_SSIN <= “01111111”; --127
      WHEN “0001” => SIG_SSIN <= “10110000”; --176
      WHEN “0010” => SIG_SSIN <= “11011001”; --217
      WHEN “0011” => SIG_SSIN <= “11110101”; --245
      WHEN “0100” => SIG_SSIN <= “11111111”; --255
      WHEN “0101” => SIG_SSIN <= “11110101”; --245
      WHEN “0110” => SIG_SSIN <= “11011001”; --217
      WHEN “0111” => SIG_SSIN <= “10110000”; --176

      WHEN “1000” => SIG_SSIN <= “01111111”; --127
      WHEN “1001” => SIG_SSIN <= “01001110”; --78
      WHEN “1010” => SIG_SSIN <= “00100101”; --37
      WHEN “1011” => SIG_SSIN <= “00001001”; --9
      WHEN “1100” => SIG_SSIN <= “00000000”; --0
      WHEN “1101” => SIG_SSIN <= “00001001”; --9
      WHEN “1110” => SIG_SSIN <= “00100101”; --37
      WHEN “1111” => SIG_SSIN <= “01001110”; --78

      WHEN OTHERS => SIG_SSIN <= “00000000”;

    END CASE;
  END IF;

  END PROCESS;
  --//-- end of memory process

S_OUT <= SIG_SSIN; -- sinewave output amplitude coded in 8 bits

END counter; -- end of architecture
```

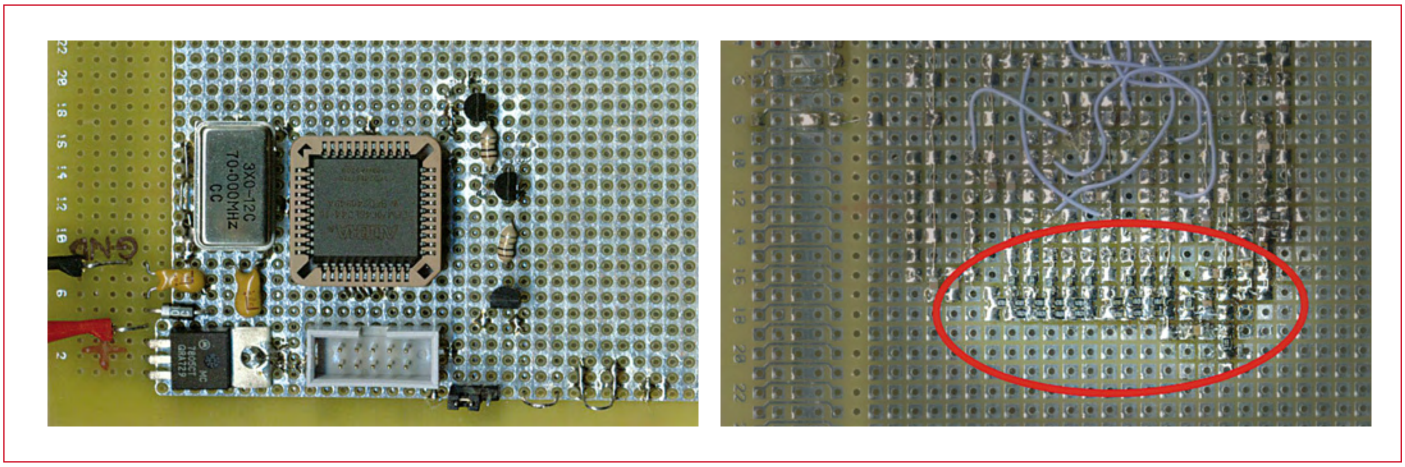


Figure 4. Prototype of the circuit on prototyping board (left). The R-2R network is fitted on the soldering side of the board (right).

the input frequency. In our case, 40% of 70 MHz comes out at 28 MHz. Any given frequency can be calculated like this:

$$increment = 65536 \times \frac{desired\ frequency}{70\,MHz}$$

Which, for a frequency of 12 MHz for example, gives an increment of 11,235 (rounded).

The filter used here does not allow operation over a wide range of frequencies. To adapt this circuit to a frequency of 12 MHz, the corresponding LC filter will be:

$$12\,MHz = 1/(2\pi\sqrt{(LC)})$$

Which is (almost) true when, for example, C = 180 pF and L = 1 μH.

Conclusion

The circuit shown here is no new invention, but a little survey on the Internet has shown that this idea of constructing it using a programmable device is not very widespread. Yet this is a simple solution that can cover RF frequencies, as here up to 28 MHz, using what is by no means the fastest or best-performing programmable device.

It's not worth struggling to find the DDS device you need and then coding the software to configure it. It's much easier to build it using more readily available devices. The literature on the subject is often long-winded and employs lots of signal analysis calculations that were not helpful in understanding this article.

Here, the CPLD device is only 48% full. This leaves you scope to code another DDS generator, or else improve this one (by increasing the number of bits in the phase register, for example, or enlarging the digital sinewave table), or to output two sinewave signals with different phases, e.g. two sinewaves in quadrature.

(080750-I)

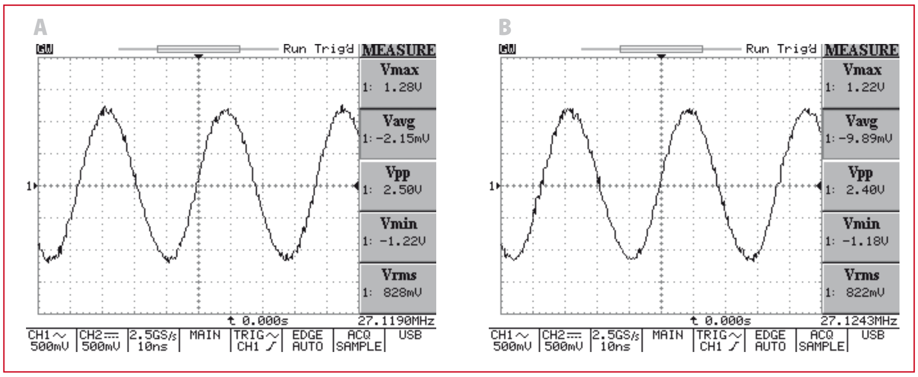


Figure 5. Output signal after filtering: 27.120 MHz (A) and 27.125 MHz (B). The frequency is shown in the bottom right-hand corner.

References

[1] USB Cyberclocks, Elektor November 2004.

[2] Serial programmable crystal oscillator, Elektor February 2005.

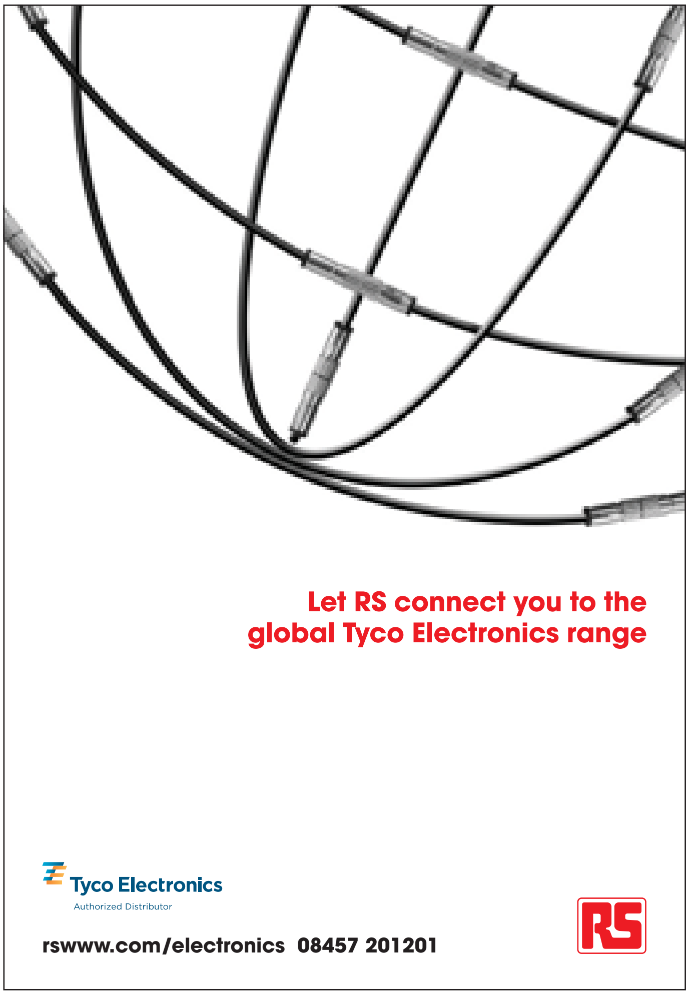
For further reading

Data sheets for DDS devices, e.g. the AD9851 from Analog Devices.

The author

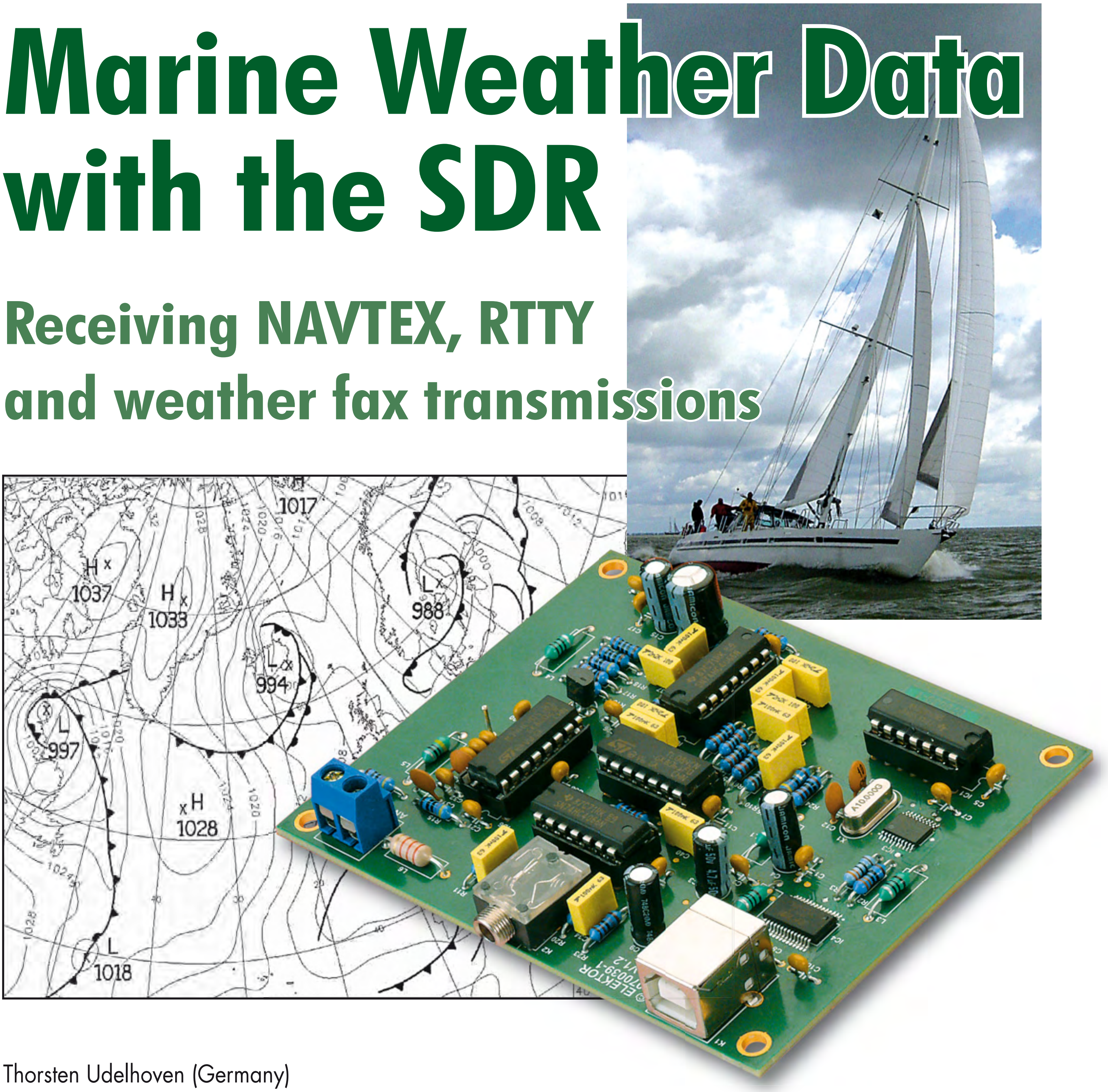
Jean Christophe Humez
is an electronics engineer, a graduate of the ENSEEIHT higher education college in Toulouse, France.

He spends his spare time taking aerial photographs from kites in the Kite Club of France. This activity, now over 120 years old, involves remotely releasing the shutter of a camera attached to the string of a kite. It's an activity that lends itself readily to all sorts of electronic solutions for controlling the camera. The camera is aimed and fired via a radio link.



Marine Weather Data with the SDR

Receiving NAVTEX, RTTY and weather fax transmissions



Thorsten Udelhoven (Germany)

As an enthusiastic yachtsman, the author of this article is primarily interested in using the SDR receiver described in the May 2007 issue of Elektor for receiving marine weather data. However, this requires enhancements to the basic design to accommodate the special operating modes of marine weather data broadcasts. In addition to the tuning and demodulation software, you need specific weather data decoding software and a second sound card.

This article is based on the author's experience and is intended to help other persons interested in this subject get started with receiving weather data using the Elektor SDR receiver [1]. A certain amount of handiwork and experimenting was necessary between unpacking the SDR module from the box and the first weather report displayed on the screen of the author's notebook computer. However, since then the receiver has accompanied the author onboard and fulfilled its duties faultlessly.

Weather data broadcasting

Long-range weather data transmitters are available for aviation and for maritime navigation. For aviation applications, they are short-wave SSB voice transmitters, such as Shannon Volmet at 5505 kHz or RAF Volmet at 5450 kHz, while for maritime navigation they are short-wave or long-wave transmitters that broadcast text messages using radio teletype (RTTY) and NAVTEX (see inset) or weather maps using radio facsimile (weather fax, also known as WEFAX).

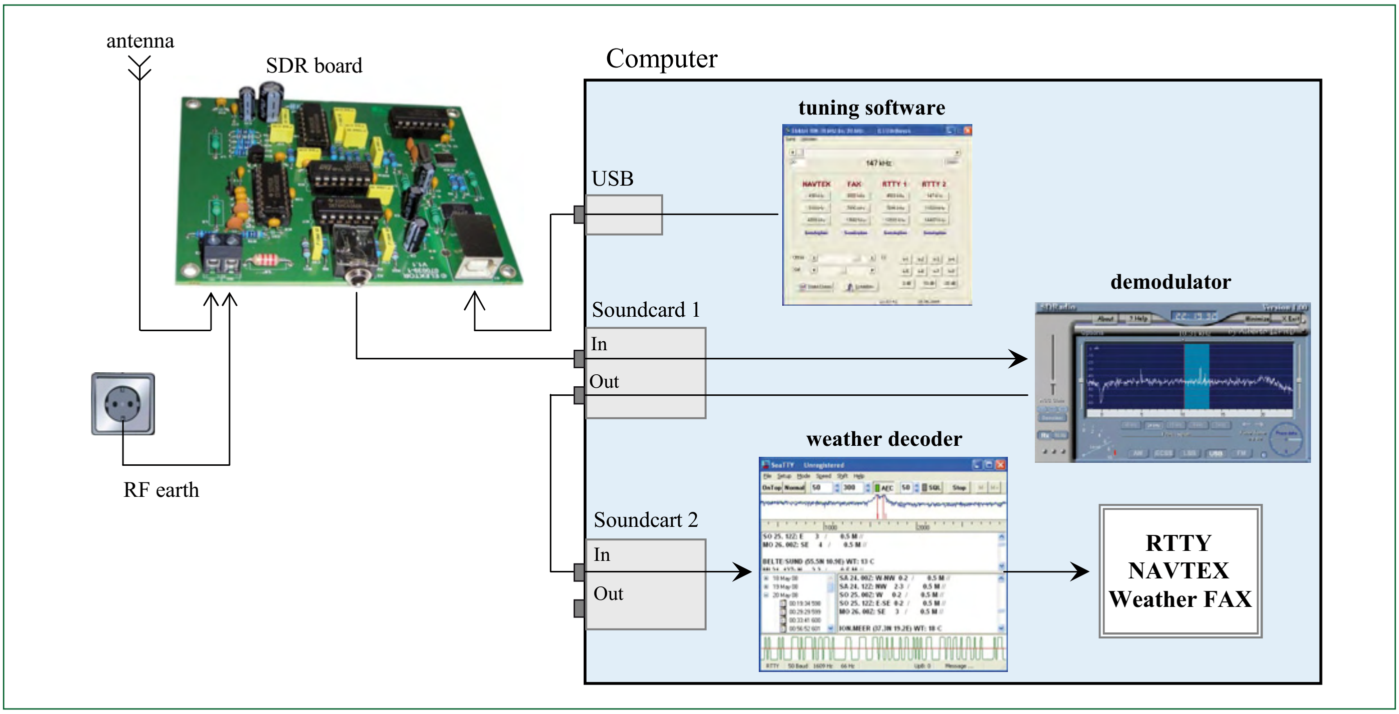


Figure 1. Elektor SDR configuration for receiving marine weather transmissions.

These transmitters are operated by national meteorological services, such as the Met Office in the UK and the NWS/NOAA in the US, according to fixed broadcast schedules that specify the transmission frequencies and times [2a][2b]. A worldwide compendium of marine weather broadcasting stations is available on the website of the US National Weather Service [3].

SDR enhancements

Figure 1 shows the SDR configuration for receiving marine weather data. No changes to the SDR board are necessary. However, in addition to the standard configuration for receiving radio broadcasts with the SDR, you need a second sound card and special weather data decoding software. The sound card is necessary because the baseband signal output from the demodulation program (such as Dream or SDRadio) must be fed back into the PC for processing by the weather data decoding software. The weather data decoding software provides the functions of an RTTY, NAVTEX, or fax decoder.

Second sound card

There are two options here: you can use a second sound card, or you can use Virtual Audio Cable.

If your PC does not already have a second sound card, a mini-USB sound card is an attractive option. The author used a Speed Link model [4], which works under Windows XP or later as a plug & play device that is ready to use right after it is plugged into a spare USB port. The technical demands on the second sound card are relatively modest, in contrast to the first sound card, which is used to feed in the in-phase and quadrature signals and must therefore have stereo input, an anti-aliasing filter, and the highest possible sampling rate.

Virtual Audio Cable [5] is a program that sells for around € 35 (US$ 50) and can be used to transfer an audio stream from one application to another. This avoids the need for a second sound card. If you prefer this option, you should try the demo version before making your final decision in order to ensure that the software works properly with your specific hardware configuration. The most economical solution is the USB sound card, which should work properly with all computers running Windows XP or later.

SDR tuning software

Version 2 of the SDR tuning software [6] written by Burkhard Kainka (Elektor SDRadio2), which was described in the September 2007 issue of Elektor Electronics, has a tuning range of 30 kHz to 30 MHz with a step size of 1 kHz (**Figure 2**). This range and resolution are perfectly adequate for this application. Nevertheless, the author wrote a special version of the tuning software for receiving weather data, so that the most important frequencies and DWD transmission schedules could be selected or displayed directly. It is based on the source text of Elektor SDRadio2, and it is included in the downloads available for this article [7] on the Elektor website. In this version of the program, the frequencies are arranged in columns labeled NAVTEX, FAX, RTTY1, and RTTY2 in increasing order of frequency. If the frequency at the top of the list does not yield adequate signal strength, you can select one of the other frequencies in the list. In principle, the further away the transmitting station is, the higher the selected frequency should be.

Demodulation software

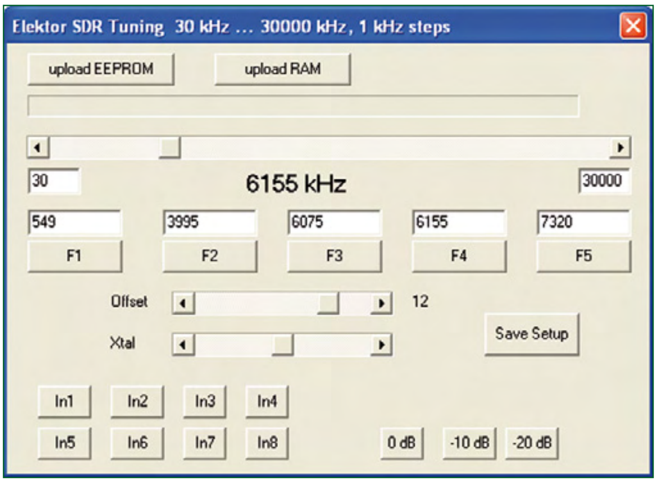
A variety of demodulator programs can be used with the Elektor SDR module. The two programs used by the author – Dream [8] and SDRadio [9] – are very popular. Single-sideband (SSB) demodulation is used to receive weather data. For this purpose, Dream is set to 'USB' (upper sideband) as shown in **Figure 3**. To keep the signal level as steady as possible, you can select the 'Fast' option under 'AGC' (automatic gain control). The 'Noise Reduction' option appears to distort the signal and is not recommended by the author. Dream includes an audio recording function ('Save Audio'), which is useful for storing the received signals.

The subjective sound quality of the SDRadio demodulation software is somewhat better due to its good noise suppression. It is also operated in USB mode for demodulation (**Figure 4**).

Now the trick is to find the right transmitting station and configure the signal bandwidth properly. Recorded examples of individual signals, which are also available on the Elektor website, are very helpful for this purpose. NAVTEX, RTTY, and weather fax signals have their own characteristic sounds, and after a bit of practice you will be able to identify them clearly.

Weather data decoder software

The author uses SeaTTY [10] for weather data decoding.

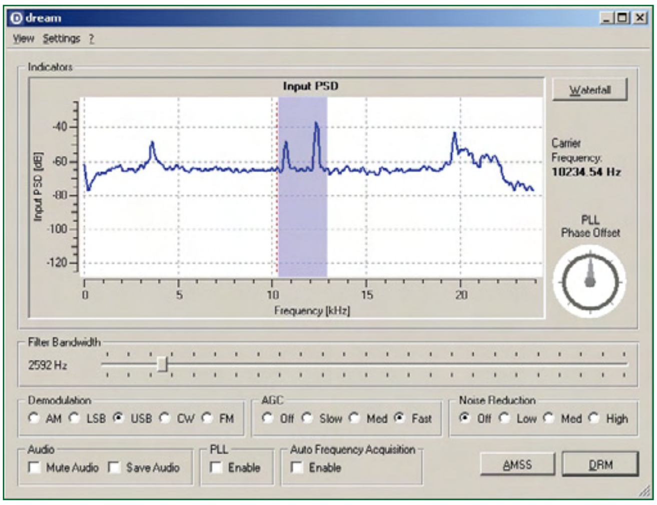

Elektor SDR Tuning 30 kHz ... 30000 kHz, 1 kHz steps

30 6155 kHz 30000

549 3995 6075 6155 7320

Offset 12

Figure 2. Elektor SDR tuning software with a broad tuning range.


Carrier Frequency: 10234.54 Hz

Filter Bandwidth 2592 Hz

Demodulation ○ AM ○ LSB ● USB ○ CW ○ FM

AGC ○ Off ○ Slow ○ Med ● Fast

Noise Reduction ● Off ○ Low ○ Med ○ High

Audio □ Mute Audio □ Save Audio

PLL □ Enable

Auto Frequency Acquisition □ Enable

Figure 3. Configuration settings for Dream.

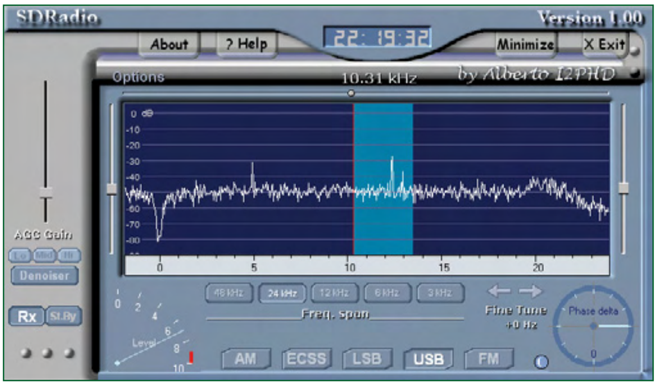


Figure 4. Another good option: SDRadio.

This program (**Figure 5**) has proven to be a good choice for this purpose because it provides all the necessary functions in a readily understandable form and is by far the easiest to use.

A few operational settings are necessary. Use the 'Setup' menu to select the appropriate sound card, such as the mini USB sound card. You can select the decoding mode in the 'Mode' menu. Use 'NAVTEX' or 'RTTY' for weather messages in text form, or 'HF-Fax' for weather fax data. The 'Speed' and 'Shift' parameters must be configured according to the specifications of the weather data transmitting station. They are stated in the header of the transmission schedule. For example, if a frequency shift of ±225 Hz is specified, SeaTTY must be configured with a 'Shift' value of 450 Hz.

The spectrum of the signal currently being received is shown in the upper part of the SeaTTY window. If SeaTTY cannot find the two peaks on its own, you can use the mouse to drag the two red lines to the right positions. Decoding then starts immediately, and the text appears in the box below the spectrum display. All data is logged automatically, and it can be retrieved and displayed whenever desired. This program costs around $ 50, and it also runs under Windows Vista.

An alternative to SeaTTY is JVComm [11], which is also widely used. It requires the same settings [12] as SeaTTY, but it is distinctly more expensive.

A freeware option is MMTTY [13], but it is limited to RTTY reception. A German help file is available online [14]. A freeware program for NAVTEX is available at [15].

Antennas

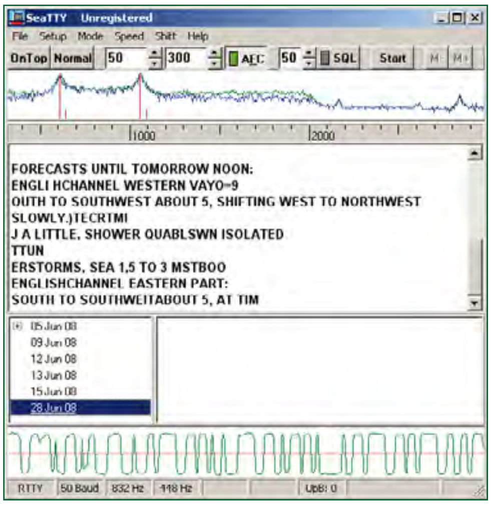
As the saying goes, the best preamp for a receiver is a good antenna. Two types of antenna are suitable for receiving marine weather radio signals: long-wire antennas and ferrite antennas.

When a sailor hears the words ‘long-wire antenna’, the first thing that comes to mind is the aft stay. The best reception is obtained with an insulated stay, with the signal being tapped off using a balun. A balun is a wideband, high-frequency transformer, and here it converts the characteristic impedance of the long-wire antenna to the impedance of the coaxial cable over the range of 100 kHz to 40 MHz. If the coaxial cable were connected directly to the aft stay, only around 10% of the signal would reach the receiver, with the rest being lost in reflections. A fully encapsulated version that can withstand sea water is available from boating accessories specialists.

For charter sailors, a traditional horizontal long-wire antenna is far less expensive and easier to install. A 10-metre length of 0.6-mm enameled copper wire is fully adequate, and it can be connected to the SDR board either directly or via a length of coaxial cable. You can string the antenna in the cabin along the four edges of the ceiling. The best way to secure it is with adhesive tape applied to wooden parts, since adhesion on other surfaces is quite poor due to the high humidity.

To significantly improve reception, you should also use a ground if at all possible. The ground is just as important as the actual antenna, since it acts as a counterpoise to the long-wire antenna. The quality of reception is unpredictable without a proper ground.

There are various ways to create an HF ground on a sailboat. If you want to know more about this subject, with some efforts the basics can be extracted from [16].


SeaTTY Unregistered
File Setup Mode Speed Shift Help
OnTop Normal 50 300 AFC 50 SQL Start M- M+
1000
2000
05 Jun 08
09 Jun 08
12 Jun 08
13 Jun 08
15 Jun 08
28 Jun 08
RTTY 50 Baud 832 Hz 418 Hz UpB: 0

```
FORECASTS UNTIL TOMORROW NOON:
ENGLI HCHANNEL WESTERN VAYO=9
OUTH TO SOUTHWEST ABOUT 5, SHIFTING WEST TO NORTHWEST
SLOWLY.)TECRTMI
J A LITTLE, SHOWER QUABLSWN ISOLATED
TTUN
ERSTORMS, SEA 1,5 TO 3 MSTBOO
ENGLISHCHANNEL EASTERN PART:
SOUTH TO SOUTHWEITABOUT 5, AT TIM
```

Figure 5. SeaTTY is a proven program for weather data decoding.

The metallic parts of a sailboat are usually bonded electrically in the interest of lightning protection. All branch circuits, the keel, and usually the negative battery terminal of the onboard electrical system and the protective earth lead of the dockside power system are connected to a common earth lead that runs from fore to aft. For this reason, the simplest solution is to use the earth contact of an onboard electrical socket as the HF ground terminal. It should be connected to the SDR board by an insulated wire with adequate cross section (at least 2.5 mm^2). Here it is important that the boat earth is absolutely potential-free in order to avoid damaging the receiver. You should first check with a multimeter to ensure that there is no stray voltage present at the earth terminal.

The advantages of a ferrite antenna relative to a long-wire antenna are its compact dimensions and operation without a ground connection. Antenna construction is relatively easy, and it is described on the Elexs website [17]. Satisfactory results can be obtained by winding 90 turns of 0.2-mm enameled copper wire on a ferrite rod. However, in practical tests the author always obtained better results with a long-wire antenna.

(080685-1)

Links:

[1] www.elektor.com/070039

[2a] http://www.metoffice.gov.uk/weather/marine/guide/radio.html

[2b] http://www.weather.gov/nwr/listcov.htm

[3] www.nws.noaa.gov/om/marine/rfax.pdf

[4] www.speed-link.com/?p=2&cat=15&pid=2810&paus=1

[5] www.ntonyx.com/vac.htm

[6] www.elektor.com/070389

[7] www.elektor-usa.com/080685

[8] http://devel.der-schall.de/downloads.php

[9] www.sdradio.org

[10] http://www.dxsoft.com/en/products/seatty/

[11] www.jvcomm.de

[12] http://dj4ig.de/wetter.htm

[13] http://mmhamsoft.amateur-radio.ca/mmtty/index.html

[14] www.janson-soft.de/seminare/dh7uaf/mmttydeuhelp.pdf

[15] www.frisnit.com/navtex/index.php

[16] www.sy-merger.de/AFU/web-content/HF-Erde_V1_0.pdf

[17] www.elexs.de/iq9.htm

NAVTEX

The Navigational Warnings by Telex (NAVTEX) system is used worldwide for the automatic transmission of regional maritime safety information (MSI) using radio teletype with error correction (SITOR-B). The main transmission frequency for broadcasts in English is 518 kHz, with a second frequency at 490 kHz available for local languages. The transmitter range is around 500 to 600 km. NAVTEX transmitters in the topics use an additional frequency at 4209.5 kHz.

NAVTEX is used primarily in the North Sea region, the Mediterranean region, the coastal regions of Japan, and along all coasts of the North American continent. The transmission times are coordinated (time sharing) to avoid interference between transmitters operating on the same frequency. In addition, the transmitter power is adapted to the signal propagation conditions, since the transmission range in this frequency band (lower edge of the medium-wave band) can be considerable, especially at night.

Another Brisk Day Toda

Temperature measurement with the ATM18 and a 1-wire bus

Wolfgang Rudolph, Burkhard Kainka and Udo Jürß (Germany)

We all know what 'cold' means, but cold is not a physical quantity. If we define it as the absence of heat, we're heading in the right direction in terms of physics. Like darkness, cold is not 'real', but in subjective human terms we can certainly talk about cold, dark nights. A more objective approach is to measure the temperature, which is what we have in mind this time with our ATM18 system.

Where does the concept of temperature come from? The path from the subjective sense of temperature to an objective, measurable physical quantity was not easy. The vibration of atoms and molecules, which becomes stronger as the temperature increases, exceeded our mental grasp for a long time.

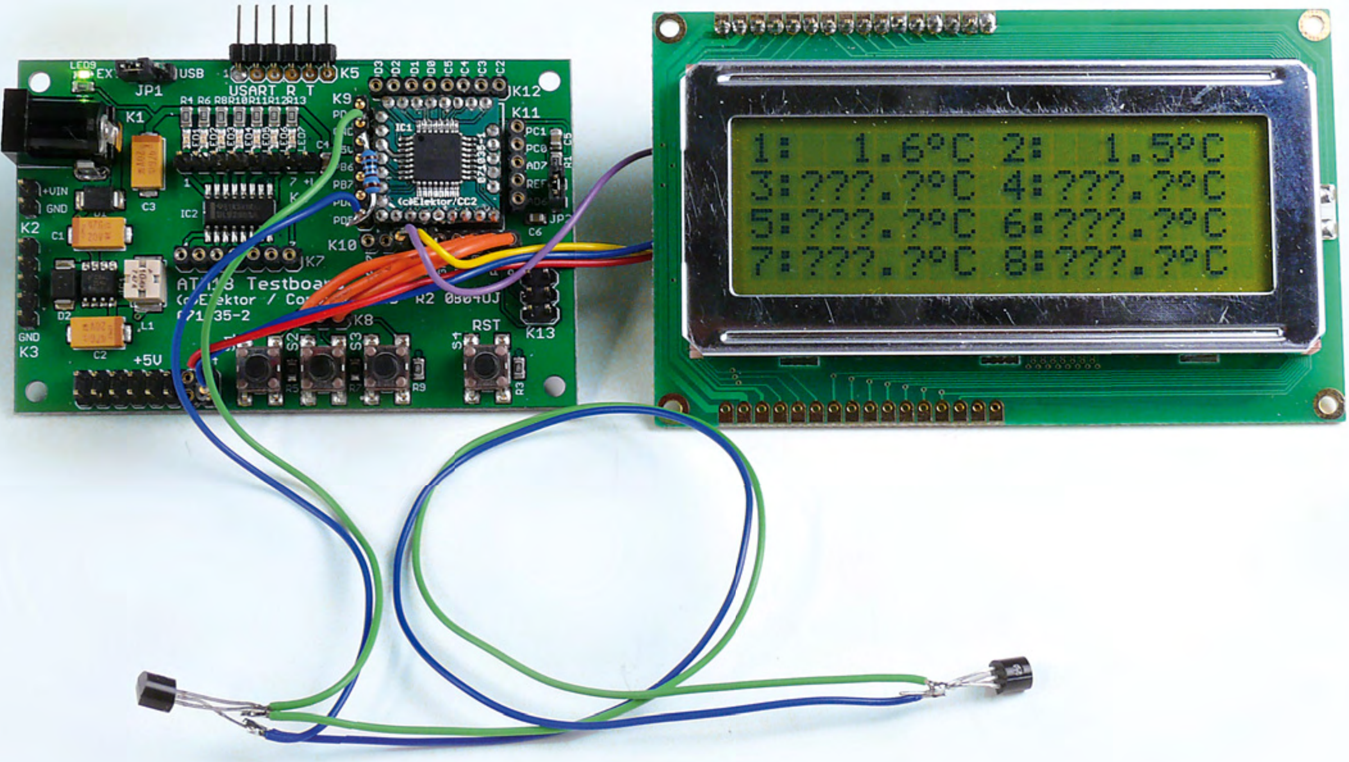
However, around 1700 (the exact date is not known) Isaac Newton had already discovered the principle of heat. He devised a temperature scale with its zero point at the temperature of freezing water. One degree on the Newton scale corresponds to 3.03 degrees Celsius. Other temperature scales followed quickly, all with different sizes of degrees. The Rømer scale (1 °Rø = 1.9 °C) was defined in 1701, followed by the Fahrenheit scale (1 °F = 0.56 °C) in 1714 and the Réaumur scale in 1730 (1 °Re = 1.25 °C). The Celsius scale was defined in 1742, and the Kelvin scale, in 1848. The degree spacing of the Kelvin scale is the same as that of the Celsius scale.

The various temperature scales differ not only in the size of their degrees, but also in their reference points. The Fahrenheit scale, which still predominates in the USA, is especially strange: the zero point (0 °F = –18.3 °C) is defined by a mixture of ice, water and sea salt, while the body temperature of a healthy person (35.6 °C) serves as the upper reference point. To make things even more complicated, there are only 96 graduations between the two reference points instead of 100. This means that a person with a body temperature of 100 °F has a slight fever (37.8 °C).
When the metric system of units (the SI system, which stands for Système International d'Unités) was introduced in 1960, it included only one unit for temperature – the Kelvin – with degrees Celsius as a derived unit. The degrees of both scales are the same, but the zero point of the Kelvin scale is absolute zero, which corresponds to –273.15 °C. Naturally, this means that 0 °C = 273.15 K. This interesting topic is discussed in great detail in a very informative article at [1].
Only quantities that can be measured can be put to use for other purposes. In the old days, when we (as electronics enthusisasts or professionals) wanted to know how warm or cold it was, we had a relatively limited selection of inexpensive temperature sensors to choose from. The most commonly used sensors were thermistors, which are resistors whose resistance depends on their temperature – but not linearly. You could connect a milliampère meter in series with a thermistor and read the temperature from a homemade scale. For any further processing, the non-linear characteristic had to be linearised. Instead of making complex calculations, people often used tables of resistance versus temperature for this purpose. It was sometimes necessary to generate your own table.
However, those days are long gone. Today sensors with integrated signal conditioning, which output the measured temperature in digital form and do not require any additional circuitry, are available at low prices. The sensors used in the project described in this article even draw their operating power from the data line, which means they can be used with what is called a 'one-wire network'.

The one-wire hoax

The one-wire technology was developed by Dallas Semiconductor, which now belongs to Maxim. This network technology uses a twisted-pair 'bus' for all of its communication. Here 'one wire' is clever marketing ploy that is only true if you don't count the ground wire. Naturally, the bus needs a ground line as well as a data line in order to work properly. What makes this technology attractive is that you can connect many devices in parallel to this two-wire bus.

The data line must be pulled to +5 V by a pull-up resistor. This represents the 'high' level. To write a logic 1, the bus is pulled to the Low level (ground) for less than 15 μs. To transmit a logic zero, the bus is also pulled Low, but in y!



this case for at least 60 μs. To initiate a communication session, the bus master (in our case the ATmega) performs a bus reset by pulling the bus Low for longer than 480 μs. It then waits for responses from the slave devices, in our case the temperature sensors, which issue an extended Low pulse.
If the master wants to communicate with a particular slave, it sends the address of the slave device and initiates a handshake. After the slave has responded, the master sends commands (which may be device-specific) and waits for responses. Every component with a 1-wire port has a unique 64-bit serial number. These numbers are assigned to the components when they are manufactured, and they cannot be changed.

There are various types of sensors on the market. We chose the DS1820 and its successor, the DS18S20, which is functionally equivalent and pin-compatible with the DS1820 and also has the type designation ‘DS1820’ printed on the package. The specifications of this device are listed in **Table 1**, and its block diagram is shown in **Figure 1.** The pin assignments of the two package options are shown in **Figure 2**.
The maximum achievable bus length depends on the value of the pull-up resistor connected between the data line (DQ) and the positive supply voltage V_{CC} (+5 V). Based on practical experience, up to 80 m (250 ft) can be regarded as reliable. Longer values are possible, but external noise on the bus increases with the length of the network. The maximum cable length can be increased by connecting a second pull-up resistor (with a value in the range of 1.5 to 10 kΩ) between the data line and $V_{CC,}$ but this slightly degrades the measurement accuracy due to increased internal heating of the sensors. If the system is operated using a three-wire bus, the supply voltage should be decoupled at each

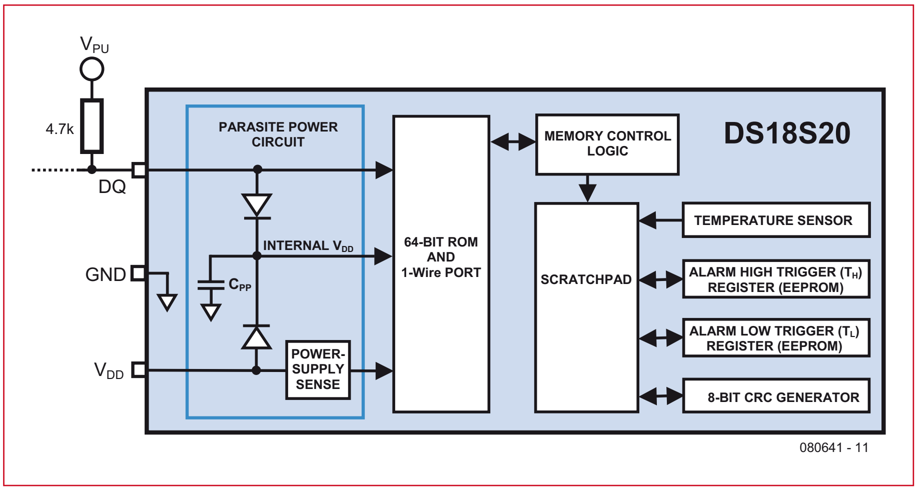


Figure 1. Block diagram from the DS18S20 data sheet.

Table 1

DS18S20 features

- 64-bit serial ID code in ROM
- 9-bit resolution (0.5 degree) from –10 to +85 °C
- Temperature measuring range –55 to +125 °C
- Package: 3-pin TO-92 or 8-pin SO
- Operating voltage: 3.0–5.5 V
- Current drain: 1 mA (standby: 750 nA)
- Temperature conversion time: 750 ms
- Drift: ±0.2 degree

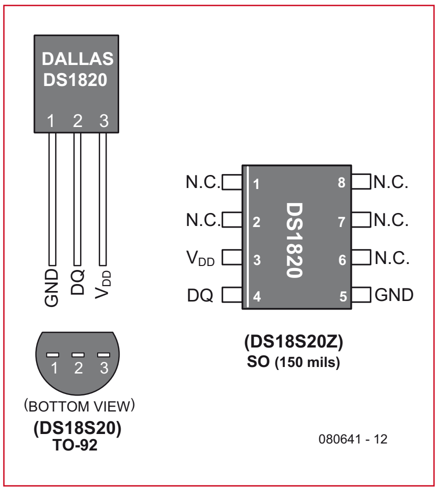


Figure 2. Pin assignments of the TO-92 and SMD versions.

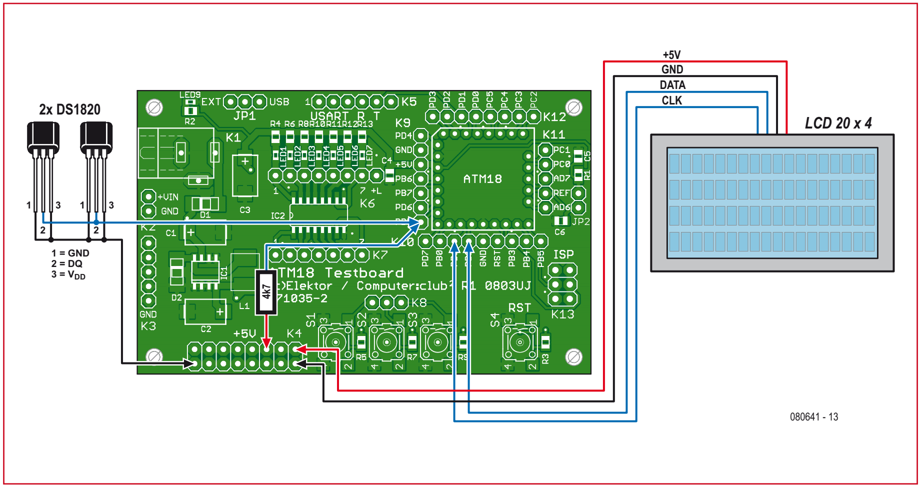


Figure 3. Wiring diagram showing the temperature sensors on the two-wire bus, the pull-up resistor and the LC display.

sensor by a 100-nF ceramic capacitor. With this arrangement, we managed to implement bus lengths up to 300 m (1000 ft) in our experiments.

Simple, fast construction

Putting a system together is very easy and only requires connecting the LC display and the twisted-pair bus line with the sensors, including the 4.7-kΩ pull-up resistor (see **Figure 3**). For reference, the wiring is also described in the source code of the ATM18_DS1820_Network project. The LCD module is connected to PB1 (clock) and PB2 (data). The V_{DD} and GND leads of the DS1820 are connected to the GND of the ATM18 test board, while the data lead is connected to PD5. This pin assignment can be altered by editing application.h. It is essential to connect the 4.7-kΩ pull-up resistor to V_{CC}. Otherwise the system won't work, even with an external power supply.

When everything has been connected properly and the right software [2] has been downloaded to the flash memory of the microcontroller, you're ready to go. The received data is transmitted via the serial interface at 38,400 baud, and the status of the network is shown on the LC display (see **Figure 4**). The BASCOM software does not drive the display, but only sends the data to the PC via the serial link.

Power supply

If you include the ground wire in the count, the one-wire bus is a two-wire bus. However, the DS1820 has three leads, and you have to decide what to do with the third lead (V_{DD}). Many Dallas chips actually need only two leads (data and ground), with the data lead also serving as the power lead. This is also true for the DS1820. However, there are certain situations in which the DS1820 sensor needs more current than it can obtain with the usual pull-up resistor value of 4.7 kΩ. The options here are to connect V_{DD} to +5 V, which means using a three-wire bus, or to make a low-impedance connection between the DQ line and V_{CC} in certain situations. If you look on the Web for sample programs for the DS1820, you may run into problems for exactly this reason. The author may have wired V_{DD} separately but not mentioned this in the software. If you use only two wires, you will be left wondering why your system doesn't work.

The two power supply options are shown on the data sheet. You can either use V_{DD} (**Figure 5**) or transmit the supply voltage via the data bus (**Figure 6**). The second option has the advantage that, as already mentioned, you only need two wires to connect the sensors to the bus. The FET that is necessary for connecting the bus to the supply voltage of the microcontroller is already present in the ATmega88. All that has to be done is to configure the appropriate I/O pin as an output and set it to the 'high' state. There are two situations where this is necessary. When a temperature measurement must be performed, the IC needs a bit of extra power for 750 ms, and when data must

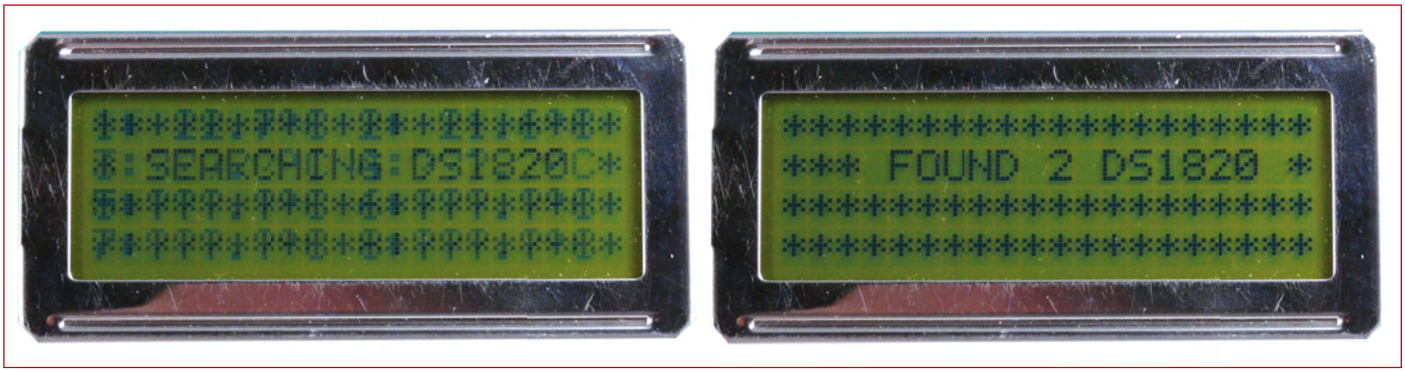


Figure 4. Find and seek.

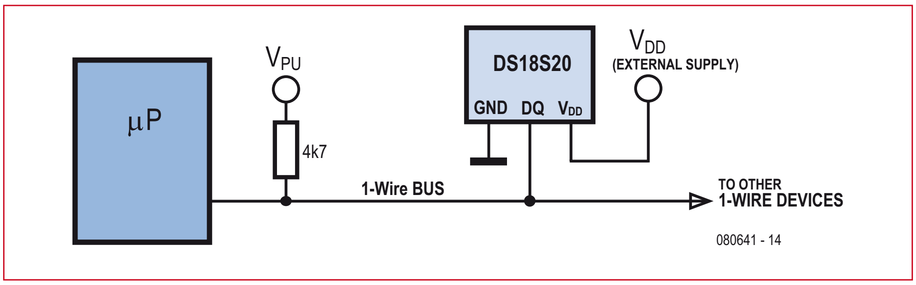


Figure 5. Power supply via V_{DD}.

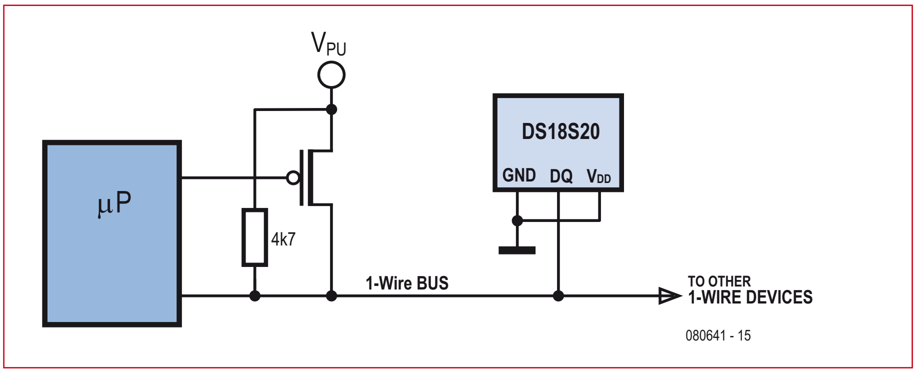


Figure 6. Power supply via DQ.

be written to the internal EEPROM, it needs more juice for 10 ms.

You can connect a single sensor or several sensors. The software for the arrangement with only one sensor is different from the software for a bus system. It is usually necessary to first identify all the sensor ICs on the one-wire bus so they can be addressed individually. For systems with only one sensor IC, Dallas has generated a simplified protocol that works without addressing. The first example in BASCOM is designed to use this simplified protocol.

One wire, one chip

BASCOM code supports the one-wire bus with special commands. This means that you don't have to do very much; all you really have to do is to decide which port pin to use for the bus. You can select any desired pin, and here we decided to use port D.5 (Config 1wire = Portd.5)

Every action is initiated by a bus reset (1wreset). After this you can write and read data. All of the details are described in the data sheet [3]. In order to communicate with a single IC, you use the SKIP ROM command [CCh]. This causes the IC to regard itself as addressed, even though its address has not been mentioned. This is similar to the situation in a tiny village with only one inhabitant: no house number is necessary.

Next comes the temperature measurement command CONVERT T [44h]. Immediately after this, the power supply must be switched on. Although the bus is already at a 'high' level in the idle state, this puts it in the low-impedance 'high' state (Ddrd.5 = 1). The maximum time necessary for making a measurement is 750 ms. To be on the safe side, you can always allocate 800 ms for this state. After this the port pin must again be configured as an input (Ddrd.5 = 0). The DS1820 now knows the temperature, but it still has to be read out. To do this, your first send a Bus Reset and then the Read Scratchpad command [BEh]. Up to eight bytes can then be read from the IC. The meaning of these bytes is described in the data sheet (see **Figure 7**). The temperature data is located in two bytes, but the data in the high byte only distinguishes between positive (00) and negative (FF) temperatures. If you want to read the temperature with higher resolution, you must evaluate even more bytes. However, this is not done in the first example.

For your initial testing in the lab, you can assume that the temperature will be positive. This means you only have to read one byte, which contains the temperature measurement in units of 0.5 degree. Simply divide the value of the byte by 2 to obtain integer degrees. **Listing 1** shows a very simple program that sends the temperature in degrees, without any digits after the decimal point, to the PC at 38,400 baud.

If you see a constant temperature of 85 degrees during your experiments, there is something wrong with the temperature conversion or the power supply, since this is the default setting after power-on before any measurement is made.

Naturally, you can do more with the temperature measurements than just display them. **Listing 2** shows a simple threshold switch with two levels. Output D2 switches when the temperature reaches 25 degrees, and output D3 switches when it reaches 30 degrees. If you can't think of any better use for this, you can always use it with two LEDs as a party gag. Have your guests hold the sensor between two fingers. With this simple instrument, you can classify them into three types: hot, lukewarm, and zombie.

Listing 1.

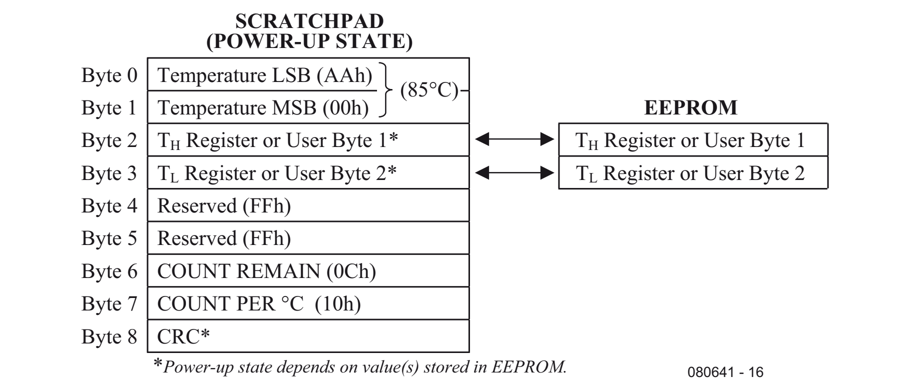
Temperature measurement with one sensor

```
‘BASCOM-AVR
‘DS18S20 1 wire PD.5


$regfile = “m88def.dat”
$crystal = 16000000
$baud = 38400

Config 1wire = Portd.5

Dim Rom1 As Integer
Dim Temp As Integer

Wait 1
Ddrd.2 = 1
Ddrb.3 = 1
Do
  1wreset
  1wwrite &HCC
  1wwrite &H44
  Ddrd.5 = 1
  Waitms 800
  Ddrd.5 = 0
  1wreset
  1wwrite &HCC
  1wwrite &HBE
  Rom1 = 1wread(1)
  1wreset
  Temp = Rom1 / 2
  Print Temp
  Waitms 100
Loop

End
```

Listing 2

Dual-threshold switch

```
If Temp > 24 Then
  Portd.2 = 1
Else
  Portd.2 = 0
End If
If Temp > 29 Then
  Portd.3 = 1
Else
  Portd.3 = 0
End If
```



SCRATCHPAD (POWER-UP STATE)

Byte 0	Temperature LSB (AAh) (85°C)
Byte 1	Temperature MSB (00h)
Byte 2	T_H Register or User Byte 1*
Byte 3	T_L Register or User Byte 2*
Byte 4	Reserved (FFh)
Byte 5	Reserved (FFh)
Byte 6	COUNT REMAIN (0Ch)
Byte 7	COUNT PER °C (10h)
Byte 8	CRC*

EEPROM

T_H Register or User Byte 1
T_L Register or User Byte 2

**Power-up state depends on value(s) stored in EEPROM.*

080641 - 16

Figure 7. ROM contents.

Several DS1820s on a single bus

Every one-wire IC has a 48-bit number, and there are no two ICs with the same number (at least not from Dallas). You obtain this number by reading and storing a total of eight bytes. You can then use it on the bus as an address for communicating selectively with a particular IC. The ICs do not even have to be the same type, since the type of each one-wire IC is included in the address (see **Figure 8**).

To start with, you can query how many devices are present on the bus (1wirecount). The Dallas bus protocol specifies that all ICs report their ID number at the beginning. BASCOM use two functions, 1wsearchfirst() and 1wsearchnext(), to query them. An array of eight bytes must be allocated for each ID code:

```
Dim Id1(8) As Byte, Dim Id2(8)
As Byte
```

The example in Listing 3 is limited to two devices on the bus. Simply to satisfy our curiosity, we displayed the two IDs on the PC. They both start with ‘10’, which is the ‘family code’ of the DS1820.

The temperature measurement routine in **Listing 4** first uses the non-addressed mode to trigger a new conversion. This means that all of the temperature measurement ICs on the bus receive the same command and make their measurements at the same time. After the 800-ms measuring interval has expired, the ICs are addressed individually and read out. Addressing is performed by sending the MATCH ROM command [55h] followed by the eight ID bytes. This can be done by sending the individual bytes in a loop, or all at once with 1wwrite Id2(1), 8. Sending an address causes a particular IC to be selected. Only this IC will respond to the readout command. You can now read out a sin-

Listing 3

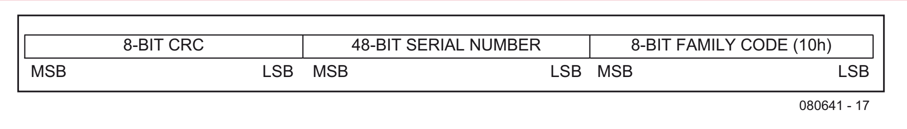
Using two sensors

```
04FF49801080033
106F0099010800B3

‘Atm18 DS1820 Bus

$regfile = “m88def.dat”
$crystal = 16000000
Baud = 38400

‘********************************
‘BASCOM-AVR
‘DS18S20 1 wire PD.5

$regfile = “m88def.dat”
$crystal = 16000000
$baud = 38400
Config 1wire = Portd.5

Dim Rom(8) As Byte
Dim Temp As Single
Dim Tempdif As Single
Dim Id1(8) As Byte
Dim Id2(8) As Byte
Dim I As Integer


Id1(1) = 1wsearchfirst()
Id2(1) = 1wsearchnext()


I = 1wirecount()
Print I

For I = 1 To 8
    Print Hex(id1(i));
Next
Print
For I = 1 To 8
    Print Hex(id2(i));
Next
Print
```

Listing 4

Measuring with two sensors

```
Wait 1
Ddrd.2 = 1
Ddrb.3 = 1
Do
  1wreset
  1wwrite &HCC

  1wwrite &H44
  Ddrd.5 = 1
  Waitms 800
  Ddrd.5 = 0

  1wreset
  1wwrite &H55

  For I = 1 To 8
    1wwrite Id1(i)
  Next I
  1wwrite &HBE
  Rom(1) = 1wread(1)
  Temp = Rom(1) / 2
  Print Temp;
  Print “   “;

  1wreset
  1wwrite &H55
  For I = 1 To 8
    1wwrite Id2(i)
  Next I


  1wwrite &HBE
  Rom(1) = 1wread(8)
  Temp = Rom(1) / 2

  Tempdif = 16 - Rom(7)
  Tempdif = Tempdif / 16
  Tempdif = 0.25 * Tempdif
  Temp = Temp + Tempdif
  Print Fusing(temp , “#.##”)
  Waitms 100
Loop
```



8-BIT CRC		48-BIT SERIAL NUMBER		8-BIT FAMILY CODE (10h)	
MSB	LSB	MSB	LSB	MSB	LSB

Figure 8. Chip address.

The ATM18 project at Computer:club²

ATM18 is a joint project of Elektor and Computer:club² (*www.cczwei.de*) in collaboration with Udo Jürsz and Burkhard Kainka. 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 27*** 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*.

gle byte, or all eight bytes at once in order to increase the temperature resolution (Rom(1) = 1wread(8)).

You can increase the temperature resolution by evaluating the seventh byte (COUNT REMAIN) of the set of read bytes. It contains a value in the range of 1 to 16, which must be interpreted as sixteenths of a degree. However, you must be careful here because the least significant bit of the low byte of the 'regular' temperature reading (0.5 degree) is the same as the most significant bit of the Count Remain register. This means that you must first round to whole degrees and then add the sixteenths count.

The best possible resolution is thus 0.06 degree, so the result is displayed with two digits after the decimal point. Naturally, the final digit should be treated with caution, since you should always make a clear distinction between accuracy and resolution. The actual accuracy is approximately 0.5 °C in the range from –55 °C to + 85 °C. However, the enhanced resolution makes it easier to recognise very small temperature changes. If you hold two sensors in close thermal contact and compare their displayed readings, you will see that the difference is usually less than 0.1 °C.

Finally, the software described here requires alterations for temperature readout in degrees Fahrenheit.

(080641-I)

[1] http://en.wikipedia.org/wiki/Temperature

[2] www.elektor-usa.com/080641

[3] http://datasheets.maxim-ic.com/en/ds/DS18S20.pdf

USB-I²C Bridge

'Easyspeak I-square-C' using 1-Wire

Fons Janssen (The Netherlands), Maxim Benelux

Modern computers no longer come with parallel or serial ports fitted. Although these ports see declining use in practice, to designers, a portless PC is a nuisance. To solve at least a part of the problem, we present an interface for the I²C bus for connection to an USB connection on your PC. It works a treat when used in the LabVIEW environment.

Quick project specs

- Simple and compact design
- Compatible with Windows XP and Vista
- Compatible with LabVIEW 7 or newer (must have .net support)
- Works on USB power
- All ICs can be sampled at Maxim IC

Sometimes you simply want your PC to communicate directly to hardware via an I²C interface. There are several options to do this, but not all of them offer the right features. For instance, there is a way to use the parallel port to emulate an I²C master (see [1]). A problem with this solution is that new PCs often lack a parallel port (a.k.a. 'LPT', 'Centronics' or 'printer'). Another problem is that contemporary Windows releases like XP and Vista do not allow easy access to the old serial or parallel ports, making them cumbersome to use if they are fitted in the first place.

Enter USB

A hopefully future-proof approach would be to use the USB port. However, this requires a complex driver and most probably a microcontroller, which requires the appropriate firmware. This is not something you can build in an afternoon. There are numerous commercial versions that don't come cheap.
The following solution can be built quickly and offers the possibility to use it in a variety of programming languages (LabVIEW, Visual Basic, etc.). This solution employs 1-Wire technology from Maxim Integrated Products (or 'Maxim' for short).

Hardware

The hardware setup is shown in **Figure 1**. The circuit connects to the PC via USB connector K1. The circuit supply voltage arriving via the USB is filtered by L1 and L2. It would probably work without these inductors, but I like to firmly rule out any potential problems. The USB data lines D+ and D– are connected to IC2; the DS2490S+, via R2 and R3. This IC contains a complete USB slave and takes care of the USB enumeration process. R1 pulls up the D+ line to indicate that IC2 is a full-speed (i.e. 12 Mbit/s) USB slave.
IC2 also contains a 1-Wire master, so that any 1-Wire slave device can be addressed from the PC. The only 1-Wire slave in the circuit is IC3 (a DS2413 from Maxim), which is an I/O Extender offering two bidirectional open-drain ports (PIOA & PIOB). R4 and R5 are

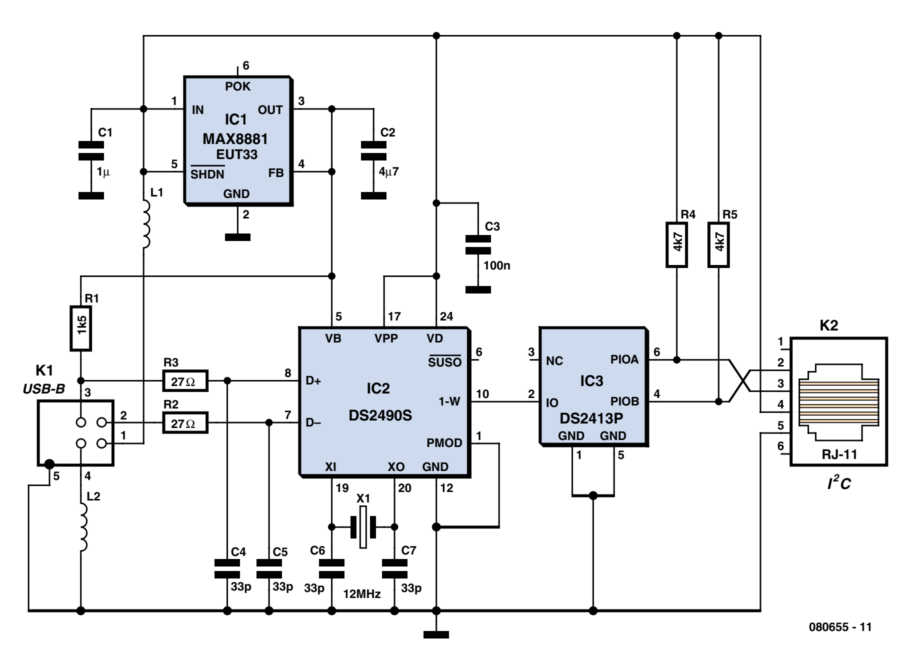
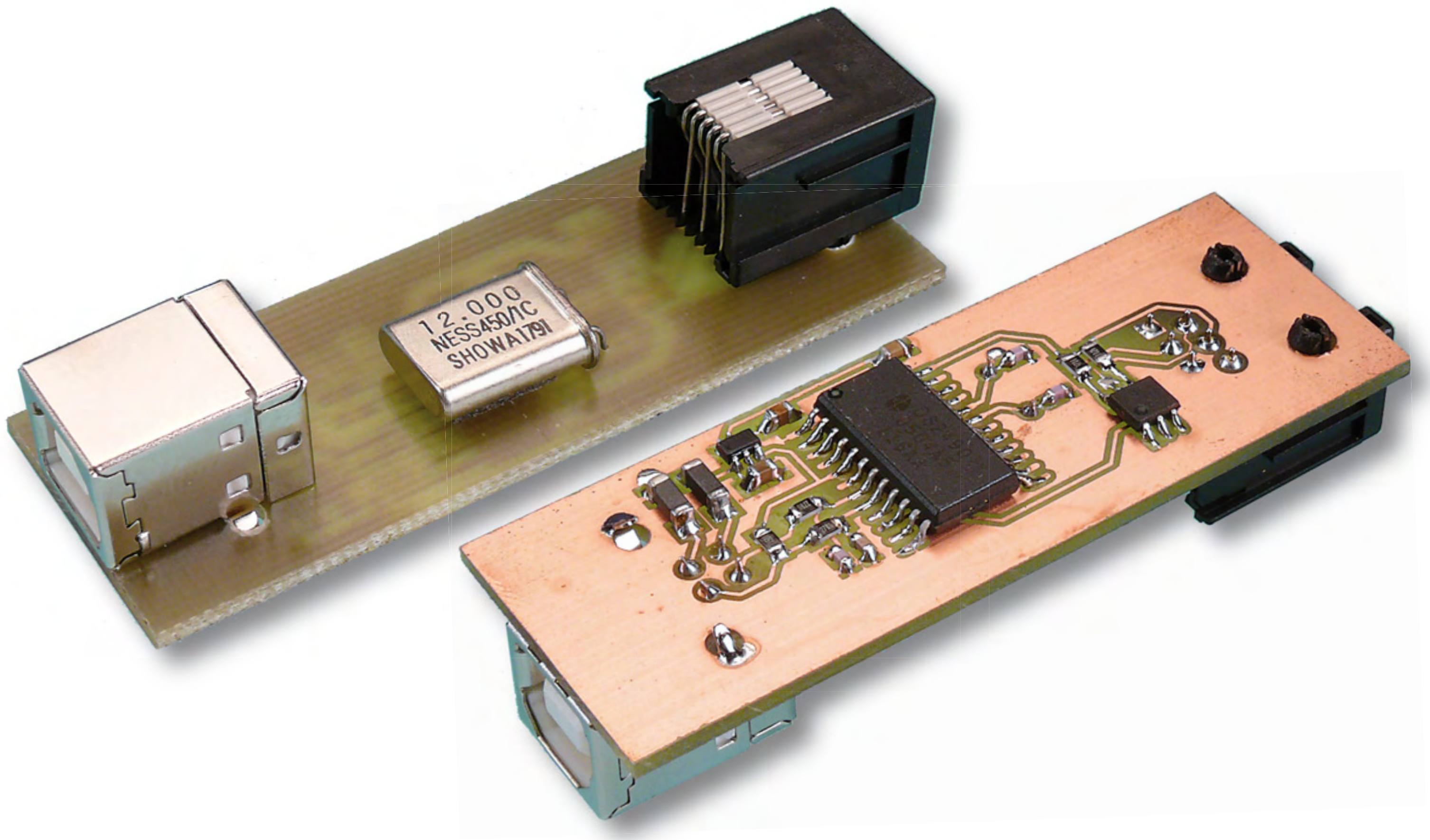


Figure 1. This circuit provides an I²C interface to a computer's USB port.



pull-up resistors for these two open-drain ports. This is exactly the right amount of hardware needed to create an I²C master. There is a PCB layout (see **Figure 2**) that can be downloaded free from the Elektor website [2] for home etching and drilling. Alternatively, readers can order a ready-made PCB from ThePCBShop.com [3].

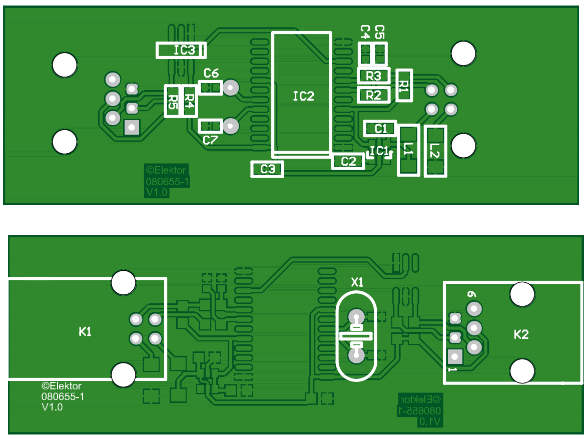


Figure 2. The component layout is very orderly. The PCB artwork can be downloaded free of charge from www.elektor-usa.com.

Software

On microcontrollers that do not have a hardware I²C master, 'bit banging' is a proven method for implementing a software I²C master. With this very same technique, a software I²C master can be created on this platform.

The software needs to control the logic levels of the two pins to generate the I²C signals. For instance, to create a so-called start condition, SCL needs to be High, while SDA drops from High to Low. To create this condition, first both SCL (PIOA) and SDA (PIOB) need to be High, after which SDA is pulled Low. So, if the software has the ability to control PIOA and PIOB, the I²C master is a matter of software.

To change the PIO outputs, the part needs to be addressed first (ROM functions). In this case, there is only one 1-Wire slave present, so the skip ROM command ('CC'; for a full list of 1-Wire commands for various devices see [4]) can be used to skip the complex addressing procedure. Then the '5A' command is sent to allow writing to the PIO output latches (the command set for the DS2413 and its PIO function flow chart can be found in its datasheet [5]).

There is an 8-bit register of which the

COMPONENT LIST

Resistors
(all 0805 shape)
R1 = 1kΩ5
R2,R3 = 27Ω
R4,R5 = 4kΩ7

Capacitors
C1 = 1µF (0805)
C2 = 4µF7 (0805)
C3 = 100nF (0805)
C4-C7 = 33pF (0603)

Inductors
L1,L2 = BLM21PG221SN1J (Murata) (220Ω at 100 MHz)

Semiconductors
IC1 = MAX8881EUT33+ (Maxim Integrated Products)
IC2 = DS2490S+ (Maxim Integrated Products)
IC3 = DS2413P+ (Maxim Integrated Products)

Miscellaneous
K1 = USB connector, PCB mount, type B
K2 = RJ-11 connector, PCB mount
X1 = 12MHz quartz crystal
PCB, ref. 080655-1, from www.ThePCB-Shop.com

two LSBs are connected to PIOA and PIOB (the other six bits are 'don't care'). To change the logic state, the new data is sent twice: once normal and once inverted. The DS2413 confirms error free reception by returning 'AA' to the master and concludes by sending the new output state as well.

Now the whole sequence can start over again, or it can be aborted by a reset from the 1-Wire master.

Table 1. Generating an I^{2}C start command

Master Mode	Data	Comments
TX	(Reset)	Reset the bus
RX	(Presence)	DS2413 is present
TX	0xCC	Skip ROM
TX	0x5A	Write PIO register
TX	0xFF	SCL=1 & SDA=1
TX	0x00	Inverted data
RX	0xAA	Data is received
RX	0x0F	Return output state
TX	0xFD	SCL=1 & SDA=0
TX	0x02	Inverted data
RX	0xAA	Data received
RX	0x2D	Return output state

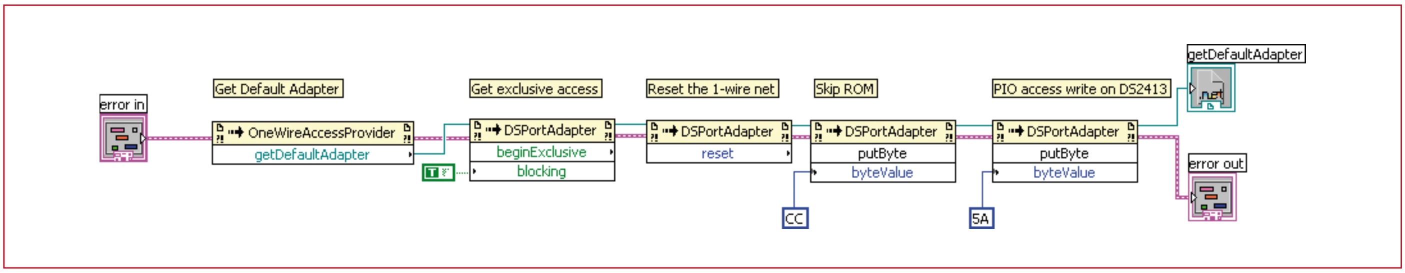


Figure 3. The Virtual Instrument called 'I^{2}C Initialize' initialises the DS2413, after which it is ready to receive I/O data.

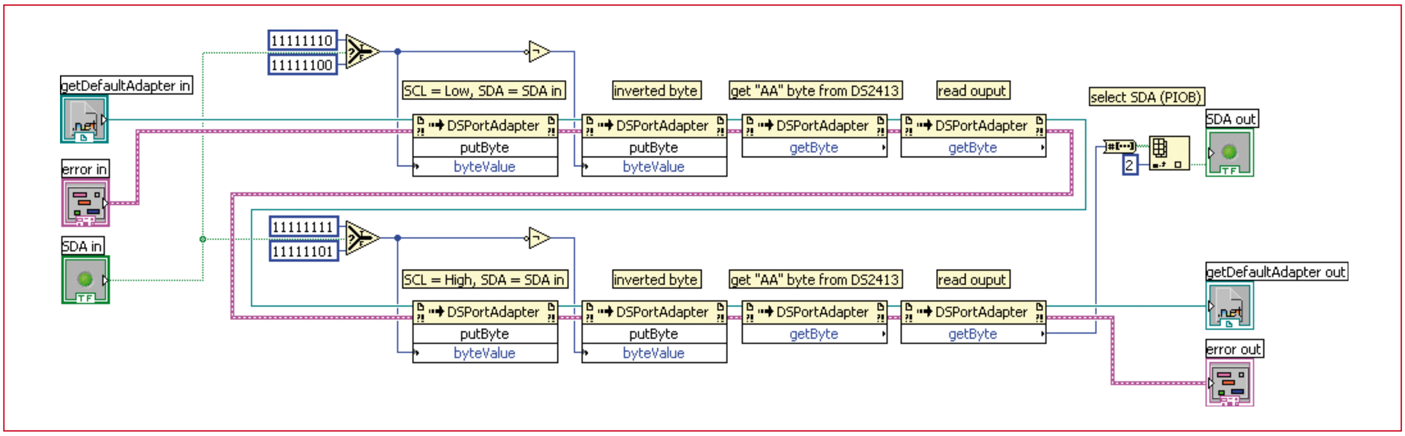


Figure 4. The VI 'I^{2}C clock' toggles SCL from Low to High. SDA can be set or reset for 'I^{2}C write' actions and read for 'I^{2}C read' actions.

Thus, a total of four bytes is transferred for each output change (if the Skip ROM and 5A command are not taken into account).
To generate the I^{2}C start command, the sequence shown in **Table 1** needs to be executed. For other events, similar sequences apply (stop, send byte, acknowledge, etc.).

.net

Maxim offers software drivers for the DS2490 (DS9490) and also a 1-Wire Software Development Kit (SDK) for Windows. This SDK offers support for Microsoft's .NET platform. As an example we will show how an I^{2}C master can be implemented in LabVIEW, using its .NET support. This example only works if the 1-wire driver is installed with .NET support. Also the Microsoft .NET framework and Visual J# redistributable package need to be installed first (see [6] for details).

Virtual Instruments

The LabVIEW Virtual Instrument (VI) 'I^{2}C Initialize' (**Figure 3** displays the block diagram) initializes the DS2490 and gives LabVIEW exclusive access to the 1-Wire net. This is to prevent other applications from accessing the 1-Wire net. After this, a 1-Wire reset is issued followed by a skip command (CC) and concluded by a PIO write command (5A).

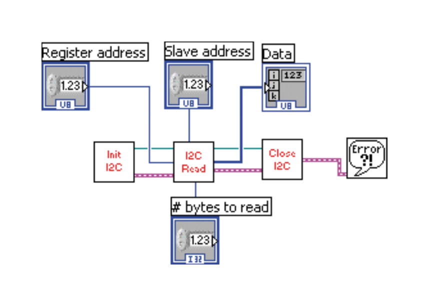


Figure 5. Example of how to read a register.

Now the DS2413 is ready to accept data for the PIOs. The VI 'I^{2}C clock' (**Figure 4**) toggles the SCL line from Low to High; SDA can be set High or Low and the VI returns the SDA state. A complete byte can be written by calling this function eight times in a row (once for every bit transmitted). If the slave is returning data, SDA is pulled High by the master. The slave can pull

SDA Low in case it wants to transmit a ‘0’ (this is okay, since PIOB is an open-drain terminal). Since the DS2413 automatically returns the updated output state, no special read action is required (see VIs ‘Send I^{2}C byte’ and ‘Get I^{2}C byte’ in the supplementary software download [2]).

The VIs ‘I^{2}C start’, ‘I^{2}C stop’, and ‘I^{2}C acknowledge’ — also in the supplementary download — use the same structure to generate the appropriate SDA and SCL signals. By combining these VIs, a complete I^{2}C read or write session can be programmed.

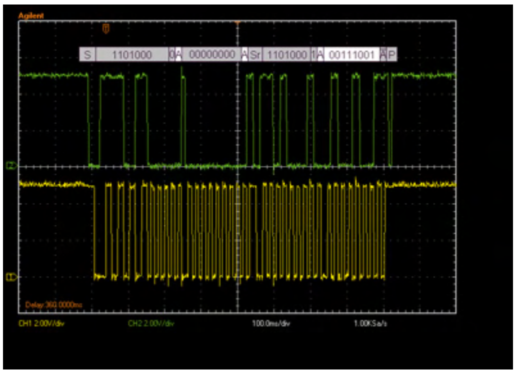


Figure 6. Read register 0 of the DS1337. S=start, A=acknowledge, Sr=repeated start, and P=Stop. Traffic from master to slave is shaded; from slave to master, white.

In **Figure 5** we see an example of a communication session, where Register 0 of a DS1337 (I^{2}C Real Time Clock) is read. The DS1337 answers with 0x39, which is the content of the register that counts the seconds. The register address is set to 0x00, the slave address to 0xD0, and the number of bytes to read is set to 1. With these settings, the scope image in **Figure 6** shows the signals generated by the circuit.

Because the overhead is quite big, the rate at which the signals are generated is severely limited (of the order of 20 bits/s). However, the solution is quite functional. The VIs that are used in the figures can be downloaded freely from Elektor’s website [2]. Samples of all the ICs mentioned in this article are available using Maxim’s sampling service for design engineers and students. It should be noted that the DS2490 is not recommended for new designs, but samples will still be available for a while.

(080655-I)

Internet Links

[1] www.maxim-ic.com/appnotes.cfm/an_pk/3230

[2] www.elektor-usa.com/080655

[3] www.thepcbshop.com

[4] http://owfs.sourceforge.net/family.html

[5] http://datasheets.maxim-ic.com/en/ds/DS2413.pdf

[6] http://pdfserv.maxim-ic.com/en/an/AN155.pdf, page 16-17.

Multiprocessing, magnetism and the milky way

3L recently announced that a new broad-band spectroscopic data acquisition system has been built and deployed for the 26-m radiotelescope of the Dominion Radio Astrophysical Observatory (DRAO) in Penticton, Canada. Designed using 3L’s Diamond multiprocessor tool-suite and based on multiprocessing hardware from Sundance, the high sensitivity system developed at DRAO has an instantaneous bandwidth of 500 MHz and a spectral resolution of 2048 channels. The phase relation of the two input signals is measured in real-time and using 3L Diamond, the instrument design was completed in only 18 months, allowing scientists to deploy the data acquisition system earlier than would have otherwise been possible.

Radio astronomers in Penticton are using the instrument to investigate the magnetic field in the interstellar medium of our Galaxy. The large bandwidth along with spectral capabilities enabled by multiprocessing is allowing them to measure the direction and field strength of the magnetic field of the Milky Way. Based on the NRC system, DRAO astronomers are now leading a project to utilize this method with other radiotelescopes around the globe including the 100-m Effelsberg telescope in Germany and the 64-m telescope at the Parkes Observatory in Australia.

The NRC design was completed using 3L’s Diamond multiprocessor tool-suite with the hardware consisting of a Sundance SMT310Q quad site carrier card populated with four daughter modules hosted on TIM (Texas Instruments Module) sites. A C series TI DSP mounted on the SMT395, two large Xilinx Virtex FPGAs on the SMT398 and a Dual Channel 1-GHz, 8-bit ADC on the SMT391. The ADC samples two input channels at 1 GHz which are then sent to the FPGAs for Fast Fourier Transform.

www.3L.com

C Sharp
Part 2
PC programs using .NET and C#

Veikko Krypczyk (Germany)

The .NET Framework relieves PC programmers of considerable effort and doesn't cost a cent. The first part of this series provided a short introduction to C#, the main .NET language. This time we present a fully developed graphics application to demonstrate the techniques of object-oriented programming.

There's one basic rule in .NET: everything's an Object! That's the fundamental difference from classic programming, in which data is separated from functions (which use and deploy data). By comparison, in object-oriented programming data and functions form self-contained units—called Objects. All Objects used in a program are arranged in Classes that specify which data can be included in an Object and which functions can be deployed in it. Let's take a simple example: a person possesses the Characteristics 'size' (a floating point number), 'place of birth' (a character string) and 'age' (an integer). We can also attribute to this person a range of activities that are called Methods here. These, for example, could include the Methods 'Sleeping', 'Eating' and 'Learning'. In our program we could define a Class by the name of 'Person'.

Listing 1. The Class 'Person'

```
class Person
{
   // Characteristics
   float Size;
   string Place of Birth;
   int Age;
   // Methods
   public void Sleeping()
   {
        // the code for Sleeping
        // thus does nothing!
   }
   public void Eating()
   {
        // relevant code
   }
   public void Learning()
   {
        // relevant code
   }
}

…

// create two objects in the Class Person
Person sample man = new Person ();
Person sample woman = new Person ();
```

A concrete Object arises when we form an Instance from a Class (by means of the keyword 'new'). Using the Class 'Person' we can create as many Instances as we wish. All these Objects then make use of the above-mentioned Characteristics and Methods. A sample implementation of a Class 'Person' in C# and the creation of two Instances are shown in **Listing 1**.

Public and private

The Characteristics and Methods of a Class can be defined in a way to make them visible for Objects of other Classes. We can also prevent this happening.

The two extreme cases here are:

- Private: the Characteristic or Method is visible only within the Class.

- Public: the Characteristic or Method is visible for all other Objects.

Taking Variables as an example, these are used only for internal calculations within a Method and are therefore defined as a private Characteristic. If reference is also to be made externally to a Characteristic of an Object, then this must be flagged as public. For instance, for the Class 'Person' it might make sense to define the Characteristic 'Age' as public, to enable other Objects to access the person's Age.

A particular advantage of object-oriented programming is its ability to 'inherit' data. A Class can be derived from another Class and in the process it then inherits all characteristics and capabilities of the parent Class. Taking an example, from the Class 'Person' we could derive the Subclasses 'Student' and 'Teacher'. Both Subclasses will then 'automatically' possess the same resources at their disposal as the Characteristic 'Age' or the Method 'Sleeping' do. Specific Characteristics can also be added individually to the derived Classes. For example we could add the Characteristic 'Grade point average' to the Class 'Student' (alone) and the Characteristic 'Salary' only to the Class 'Teacher'. It goes without saying that it is also possible to create Subclasses of .NET Base Classes, for example to create variations of top-level elements and much more. The 'inheritance' feature means that comprehensive Class hierarchies can be set up and we'll be demonstrating a tangible example in the next section.

As well as these Characteristics and Methods, a Class can also possess Class Events. These Events can be initiated while a program is running by the Objects in the Class. If the user clicks on a button for example, the event 'Click' will be triggered for the Buttons. In the definition of Classes we need to post some source code for each Event of this kind, which sets out the desired reaction. In this way these Events provide the foundations of the user interactivity required.

2-D graphical programming

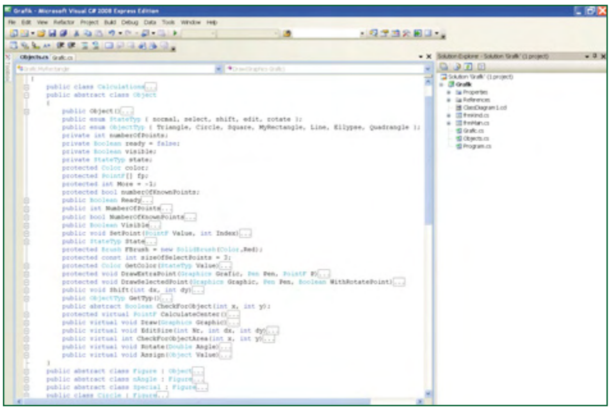
Now let's turn to an interesting practical example, specifically vector graphics programming in two dimensions. You can find the source code on this article's Web page [1]. If you have already installed Visual C# 2008 Express-Edition (see Part 1 of this series) you can get the whole project up and running with just a double mouse clicks on the file name *.csproj-Datei* in the Development Environment.
Vector graphics are assembled from individual drawing objects, which in turn are composed of a defined number of points linked together by straight lines or parts of a circle. So what do we need to do to define these drawing objects as Objects in our program? We can start by implementing a way of drawing quite simple objects, such as straight lines. Afterwards we can derive Program Objects representing more complex constructions from (drawing) objects created in a more simple fashion. This ensures the program can be expanded without difficulty, which is an important point.

Two significant elements determine program structure:

- The Class 'Grafic', responsible for graphics output and the management of the drawing objects. This Class is set out in the file *Grafic.cs*.
- Classes (hierarchy) of the drawing objects. These are packaged in the file *Objects.cs* (see **Figure 1**).

Architecture

In our application it's important to distinguish between the (virtual) drawing area for the Objects and their actual on-screen representation. The drawing operation is carried out in the background on an Object of the Class 'System.Drawing.Graphics' from the .NET-Framework. For preparing the drawing area and managing the drawing objects we implement the Class 'Grafic'. The output you see on the screen results from within a top-level element called 'PictureBox'. This is how the dimensions of the drawing surface (background bitmap) can be determined independently of the output area actually visible (which is limited by the display screen size you are using). The extract area displayed at any given time can be shifted by means of scroll bars (see **Figure 2**).



Grafik - Microsoft Visual C# 2008 Express Edition
File Edit View Refactor Project Build Debug Data Tools Window Help
Objects.cs Grafic.cs
Grafik.MyRectangle
DrawGraphics Grafic
Solution Explorer - Solution 'Grafik' (1 project)
Solution 'Grafik' (1 project)
Grafik
Properties
References
ClassDiagram1.cd
Grafic.cs
Objects.cs
Program.cs

```
{
    public class Calculations[...]
    public abstract class Object
    {
        public Object()[...]
        public enum StateTyp { normal, select, shift, edit, rotate };
        public enum ObjectTyp { Triangle, Circle, Square, MyRectangle, Line, Ellypse, Quadrangle };
        private int numberOfPoints;
        private Boolean ready = false;
        private Boolean visible;
        private StateTyp state;
        protected Color color;
        protected PointF[] fp;
        protected int More = -1;
        protected bool numberOfKnownPoints;
        public Boolean Ready[...]
        public int NumberOfPoints[...]
        public bool NumberOfKnownPoints[...]
        public Boolean Visible[...]
        public void SetPoint(PointF Value, int Index)[...]
        public StateTyp State[...]
        protected Brush FBrush = new SolidBrush(Color.Red);
        protected const int sizeOfSelectPoints = 3;
        protected Color GetColor(StateTyp Value)[...]
        protected void DrawExtraPoint(Graphics Grafic, Pen Pen, PointF P)[...]
        protected void DrawSelectedPoint(Graphics Graphic, Pen Pen, Boolean WithRotatePoint)[...]
        public void Shift(int dx, int dy)[...]
        public ObjectTyp GetTyp()[...]
        public abstract Boolean CheckForObject(int x, int y);
        protected virtual PointF CalculateCenter()[...]
        public virtual void Draw(Graphics Graphic)[...]
        public virtual void EditSize(int Nr, int dx, int dy)[...]
        public virtual int CheckForObjectArea(int x, int y)[...]
        public virtual void Rotate(Double Angle)[...]
        public virtual void Assign(Object Value)[...]
    }
    public abstract class Figure : Object[...]
    public abstract class nAngle : Figure[...]
    public abstract class Special : Figure[...]
    public class Circle : Figure[...]
```

Figure 1. In Visual Studio we can display the Characteristics and Methods of an Object in overview form.

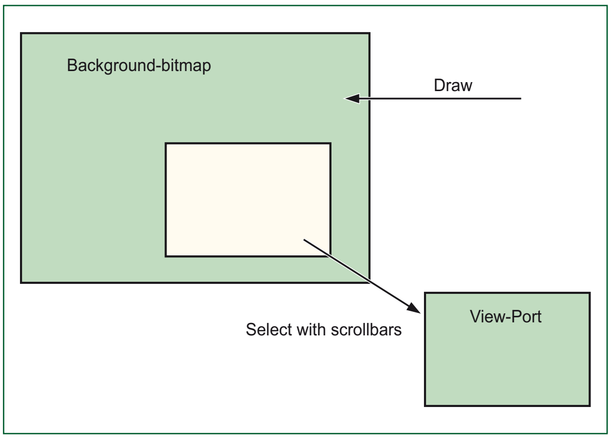


Figure 2. Our graphics program distinguishes between the virtual drawing board and the extract actually displayed.

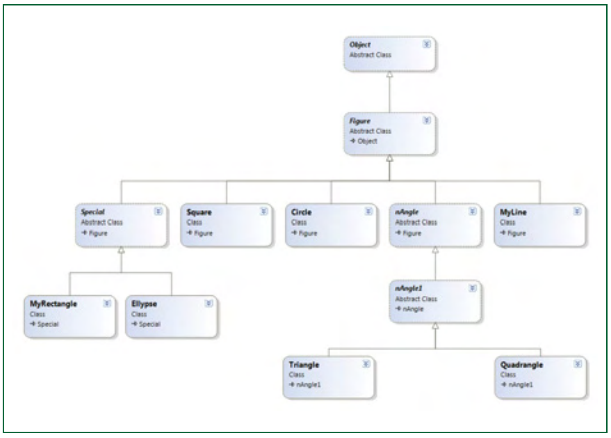


Figure 3. In the Class diagram we can see which drawing objects are derived from other Objects.

Individual drawing objects are managed in an Array that goes by the name of 'Figure'. It is defined like this:

```
public Object[] Figure;
```

As the type of the elements is defined simply as 'Object', all kinds of drawing objects can be stored in the Array.
Finally there is also provision for temporary drawing

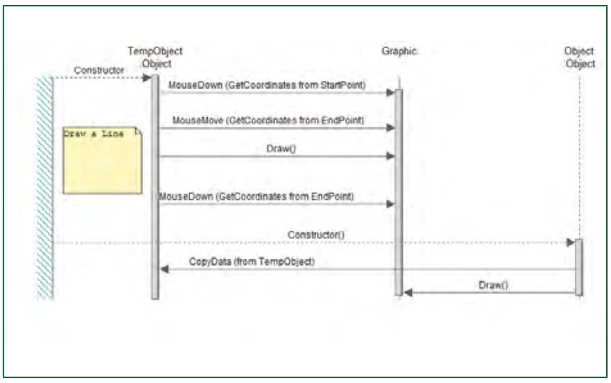

TempObject
Object
Graphic
Object
Object
Constructor
MouseDown (GetCoordinates from StartPoint)
MouseMove (GetCoordinates from EndPoint)
Draw()
Draw a Line
MouseDown (GetCoordinates from EndPoint)
Constructor()
CopyData (from TempObject)
Draw()

Figure 4. The sequential diagram indicates how the Objects in our sample application interact.

objects, which can for example be used to visualise the lateral shifting of an object. This is achieved by making a copy of the original object before this shifting takes place and altering this copy during the period of user interaction. To inform the user what moving the original object would do, one of these temporary objects is displayed directly on the relevant output area.

Base Objects

The graphics framework of .NET provides us with a series of drawing functions (Base Objects). These Base Objects create very simple geometric figures (such as Line, Rectangle and Circle). Base Objects are assembled in a Class hierarchy and **Figure 3** shows a so-called Class diagram. The top Base Object is the Class Object. This one is concerned with a so-called abstract Class, meaning that no Instances can be created from this Class. However, the Class Object already contains all of those Characteristics and Methods that are of significance for the Classes that follow on. In the construction of our Class library the principles of generalization and specialization are followed strictly. Characteristics and Methods that are valid for more several Objects are assigned to a superior Class. Specials — characteristics valid for a single drawing object only — are implemented in the corresponding Special Class.

Interactivity

As in most graphics applications here too we require interaction with the user. This includes, for example, the ability to draw or shift objects. The interactive drawing process itself is controlled by the Mouse Events 'MouseDown', 'MouseUp' and 'MouseMove'. In the so-called sequence diagram (**Figure 4**) our example shows how a line is drawn. To illustrate the drawing process a temporary object is indicated in the output area with each mouse movement, before the old object is erased. If the left mouse button is activated during the drawing process, the coordinates of the current location are stored. After this, selection of the next point of the figure can take place. Once all points in our figure have been defined in this way the temporary figure is transferred into the object store.

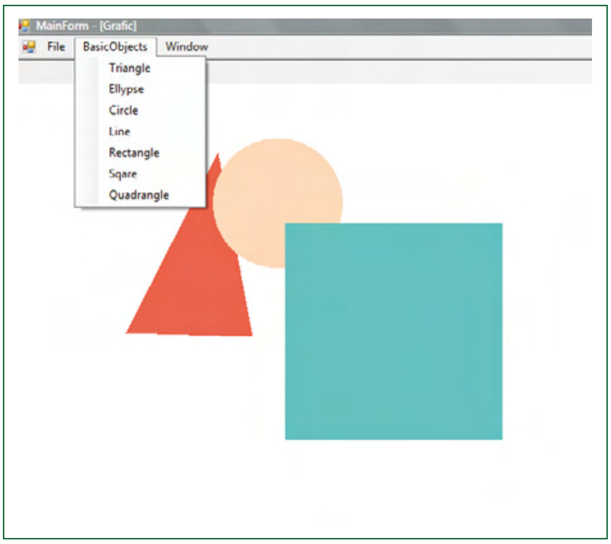


Figure 5. The graphics program in action.

Users also need to use the mouse for marking Objects. With each mouse click a test is made to check whether there is an object located at this position. If no object is found at the mouse position and a different object has already been selected, this marker is cancelled. CheckForObject (*int x, int y*) is responsible for this Method, using the coordinates of the mouse pointer as parameters. If these are found to be located inside an Object (*Rectangle, Circle*) or on an Object (*Line*), then the result is reported as 'true'. In the negative case the report back is 'false'. Objects created on one occasion should also be capable of being edited subsequently and for this we use the Method 'EditSize' of the drawing object concerned.

Expansion

Figure 5 shows our application in action. The ability to create objects in almost any form we wish on a drawing board and then move them around makes this a pretty good starting point for all kinds of graphic applications that crop up daily in electronics. What, for example, if we wanted to turn data measurements into a bar chart? No sooner said than done. The starting point of our Class Diagram is the Class 'Special' (see Class Diagram in Figure 2). **Listing 2** gives the code of this Class. Following the colon in the first line the compiler is informed that the Class Diagram inherits the Characteristics and Methods of the Class 'Special'.

Near the top of the listing an Array called 'Data' is defined and this records the test measurement data for the bar chart. The Method Diagram() is the so-called Constructor of the Class. This is always called when an Instance of the Class Diagram is produced. Following this are a couple of Method definitions with the keyword 'override'. This means that when the Method is called, it does not use the code of the inherited Method (implemented in the Class 'Figure'). Instead new code is assigned, which overwrites the inherited code. After all, it's not particularly helpful to, say, rotate a bar graph by a specific angle. The Method 'Rotate' in this new Class is therefore implemented without code; when you call it up simply nothing happens.

The Method 'Draw' produces the bar graph. The two statements *Grfc.FillRectangle* and *Grfc.DrawString* are of major significance, as they draw a bar for each value of data and write the value above it.

You can test out the whole thing by starting up the application and in the 'Extra' menu selecting the drop-down option 'Plot Data …' (see **Figure 6**). The program then instructs you to select a text file that contains a simple listing of measurement values (for example the sample file *data.txt*, to be found in the source code package [2]). If you then click on the drawing board, you can make the bar graph larger and move it around. This functionality 'inherits' the Class Diagram of the Class 'Special', which in turn is derived from the Class 'Figure'. The programmer must not implement the feature for the Class Diagram, not even once!

The source code of this exercise is at your disposal for developing your own expansion ideas, so get cracking! The code (created with Visual C# 2008, Express Edition) can be downloaded *gratis* at the web page for this article [1].

(080752)

Internet Link

[1] www.elektor-usa.com/080752

Author's email

veikko2000@yahoo.de

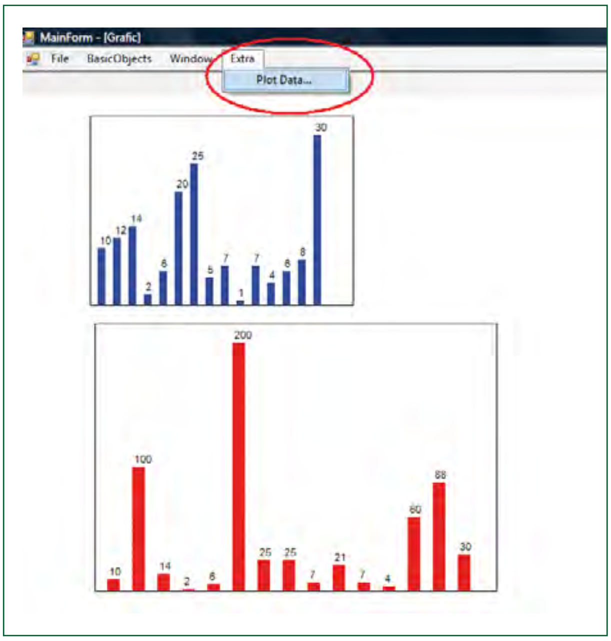


Figure 6.
The Class 'Diagram' is derived from the Class 'Figure' via the Class 'Special', enabling a bar graph to be created. The 'shift' and 'enlarge' functions are implemented automatically.

Listing 2. The Class 'Diagram' (extract)

```
public class Diagram : Special
    {
        int[] data = new int[15];
        public Diagram()
        {
            NumberOfPoints = 2;
            fp = new PointF[NumberOfPoints];
        }
•••
        public override void Rotate(Double Winkel)
        {

        }
        public override void Draw(Graphics Grfc)
        {
•••
                    Grfc.DrawPolygon(Stift, fp);
                    int maxValue = 0;
                    for (int i = 0; i < data.Length; i++)
                    {
                        if (data[i] > maxValue) maxValue = data[i];
                    }
                    int maxWidth = (int)(fp[2].X - fp[0].X);
                    int maxHeigth = (int)(fp[2].Y - fp[0].Y)-20;
                    int width = maxWidth / 31;
                    int buttom = (int)fp[2].Y;
                    int left = (int)fp[0].X + width;
                    Brush brush = new SolidBrush(Color2);
                    if (maxValue != 0)
                    {
                        for (int i = 0; i < 15; i++)
                        {
                            int currentHeigth = maxHeigth * data[i] / maxValue;
                            Rectangle rect = new Rectangle(left + 2 *
   i * width, buttom - currentHeigth, width, currentHeigth);
                            Grfc.FillRectangle(brush, rect);
                            Grfc.DrawString(data[i].ToString(), new Font("Arial", 8), new
   SolidBrush(Color.Black), new PointF(left + 2 * i * width, buttom - currentHeigth - 15));
                        }
                    }
•••
        public override void SetData(int[] Data)
        {
            Data.CopyTo(data, 0);
        }
  }
```

ECIO40 and USB

Connect a PIC to USB within 10 minutes

Bert van Dam (The Netherlands)

Analog and digital signals can be measured easily using an E-blocks ECIO module. With the right drivers it becomes a simple matter to read these values on a PC via a USB link. In this article we show how this can be implemented within 10 minutes.

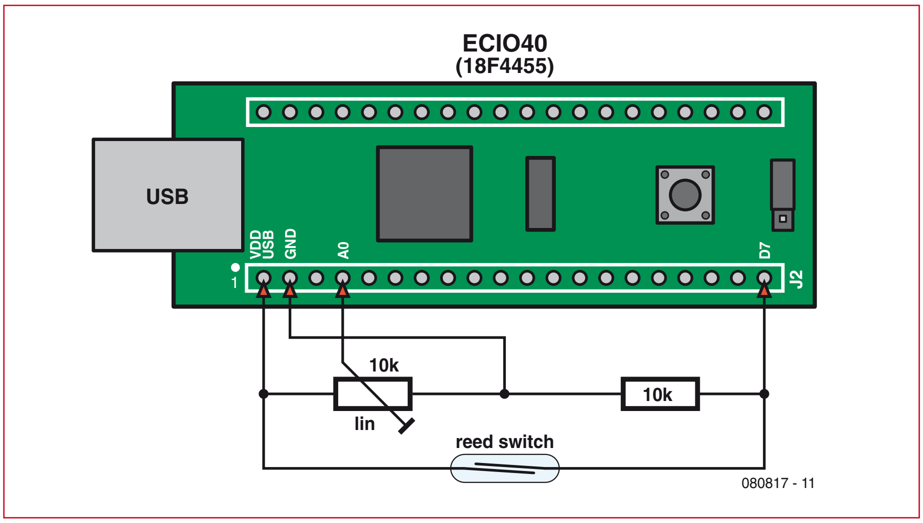


Figure 1. The circuit diagram reveals that very little hardware is required for this project.

Quick Specifications

- ECIO modules available from Elektor
- Works under Windows XP and Vista
- No power supply required; it uses the USB supply
- Connected within 10 minutes
- USB driver generated by Flowcode
- Simple communications via HyperTerminal

The ECIO40 is a modern 18F4455 PIC microcontroller mounted on a mini PCB and includes an integrated USB bootloader. You can therefore program the ECIO40 without the need for an expensive programmer. All you need is a spare USB port on your PC and a small (free) program. The ECIO40 module is available from the Elektor Shop.

A new feature is that you can now use the same USB port to communicate directly with the ECIO40. This offers a whole new range of possibilities. You can for example turn it into a mouse or joystick, or use it to make measurements and display and process them on your PC.

In this article you can find out how to measure an analogue and digital signal with the ECIO40 and then show them on your PC via a USB connection.

Hardware setup

The analogue signal is created using a potentiometer that has been wired up as a potential divider to provide a voltage between 0 to 5 volts to pin A0 of the ECIO40 (see **Figure 1**). For the switch we've used a reed switch. This is a type of switch that reacts to the presence of a magnetic field. The switch, which consists of two contacts inside a glass tube, is normally open. When you hold a magnet near it the contacts are attracted towards each other and the switch closes.

This type of switch is often used in burglar alarms. You could for example mount the switch on a doorframe and the magnet in the door itself. When the door is opened the switch opens as well, which can then set off an alarm. You can of course use an ordinary switch in this setup as well.

Apart from the ECIO40, a few components and Flowcode V3, you also need the USB pack for this project. This can be downloaded from the website of Matrix Multimedia [1]. Note that you need the licence number from your Flowcode CD for this.

Get cracking

First of all the circuit is constructed on a piece of experimenter's board. The

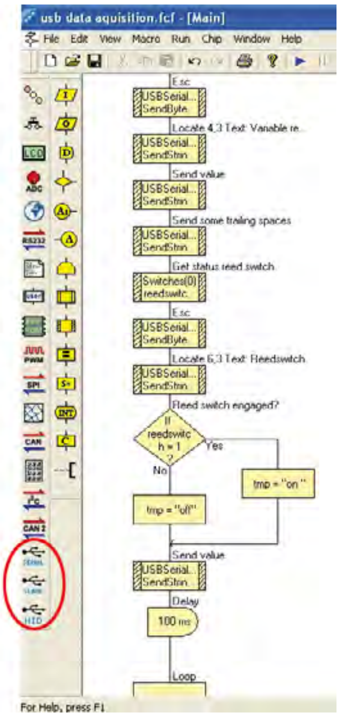


Figure 2. After installation of the USB pack the extra hardware components appear in the component tray.

supply for the circuit comes via the USB port of the PC, so you won't need a separate power supply.

Next you have to install the USB pack in your version of Flowcode. We recommend that you first upgrade to the most recent version of Flowcode V3 if you haven't done so already. After you have unzipped the USB pack you should copy the Flowcode V3 folder to the folder with the same name in the Flowcode installation (this is usually in C:\Program Files\Matrix Multimedia). Next you should run the program 'Install USBPack.bat', which is in the Flowcode V3 components folder, in order to register the new components. On most PCs you should be logged in as Administrator, or have administrative rights, in order to do this. When you next start up Flowcode you'll find that three extra hardware components have appeared in the components tray (**Figure 2**).

In this article we make use of the 'serial' component. This component is specially made to use a USB port with traditional communications packages such as HyperTerm. Click on it in order to add the component to your project, which makes the following USB macros available for use:

- Initialise_serial
- SendByte
- SendString
- ReadByte
- ReadString

We start the program with an Initialise_serial macro. The result of this macro can be stored in a variable (for example, retval). In this demo project we don't make further use of it, but you can use the variable to see if the USB connection was actually made successfully (retval=0), or if it hadn't (retval=255). A loop starts next, where the analog and digital signals are measured and sent to the PC.

Hyperterm

On the PC side something should be done with these measurements. Since the USB component is meant to be used with a traditional communications package, we chose to use the HyperTerm program. This communications program is included for free with Windows XP and earlier versions of Windows. Vista users can use HyperTerminal Private Edition, a free software package from Hillgraeve. Unfortunately, the free version is no longer available from this company, but via Google (search for 'htpe download') you can still find many places where you can legally download a copy of the free version. This version happens to have more functions than the one supplied as standard with Windows XP and earlier, which makes it a worthwhile download for XP users as well. The reason for using this particular package is that it permits the microcontroller to control the cursor on the PC screen. There are three commands that have been used in this project:

1. **Esc H**, put the cursor in the home position, at the top-left of the screen.
2. **Esc Yr,c**, position the cursor in row *r* and column *c*.
3. **Esc J**, clear the screen from the cursor onwards.

With these commands it is easy for the microcontroller to create a screen image and then send it to the PC. A small complication is that the values *r* (the row) and *c* (the column) have to be expressed in ASCII for historical reasons, and should therefore be increased by 32. For example, if you want to select row 11, you have to add 32 to it (resulting in 43) and then send the character that has this ASCII value, in this case a '+'. In the download for this article [2] you'll find a table that you can print out, which makes it a lot easier to choose the right character to position the cursor on the screen. The design for this project, which uses this

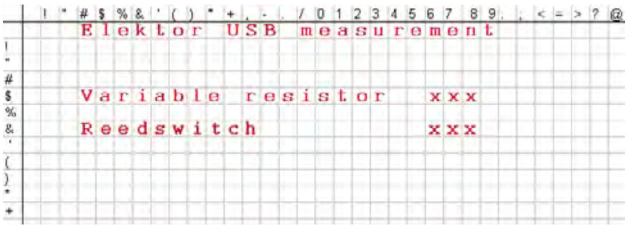
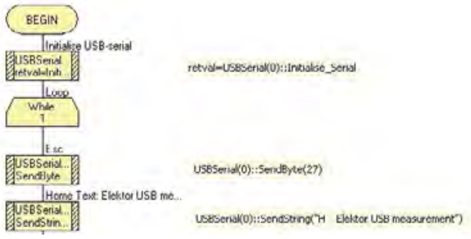
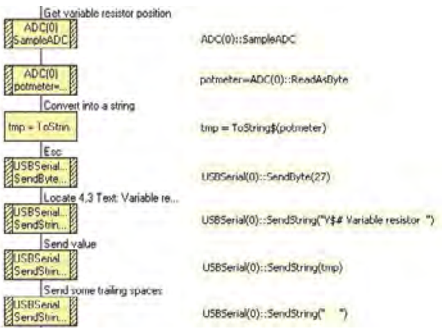

```
 ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
    Elektor USB measurement
!
"
#
$   Variable resistor    xxx
%
&   Reedswitch           xxx
'
(
)
*
+
```

Figure 3. Screen layout for this project.



```
retval=USBSerial(0)::Initialise_Serial
USBSerial(0)::SendByte(27)
USBSerial(0)::SendString("H   Elektor USB measurement")
```

Figure 4. Flowcode for the initialisation of the USB connection and displaying the first line of text.



```
ADC(0)::SampleADC
potmeter=ADC(0)::ReadAsByte
tmp = ToString$(potmeter)
USBSerial(0)::SendByte(27)
USBSerial(0)::SendString("Y$# Variable resistor  ")
USBSerial(0)::SendString(tmp)
USBSerial(0)::SendString("    ")
```

Figure 5. An ADC measurement followed by the transmission of data via USB.

table, is shown in **Figure 3**.

To place the cursor on the top-left of the screen we need to send an Esc H command. The ASCII value for Esc is 27, so we first send '27' via the USB link, followed by an 'H'. The first line of text can be added to the end of that command. We make things a bit easier by adding a few spaces to position the text in the correct column (see **Figure 4**).

Flowcode

Next an analog measurement has to be made via pin A0. First an ADC component is added to the workspace; next it is assigned to the analogue/digital converter on pin A0 (i.e. ADC0) and instructed to carry out an ADC measurement, with a byte as the result. The variable that holds the measurement is called 'potmeter'. You can't send the value of 'potmeter' directly to the PC because HyperTerm then thinks you're sending it a single character. Should the value of potmeter be 65 for example, then HyperTerm will display the letter A (which has an ASCII value of 65) instead of the number 65. This problem is solved by converting 'potmeter' into a string.

The measured value is preceded by the text 'Variable resistor' and from Figure 3 you can see that the text starts at screen position *$#*: row 4, column 3 (counting starts at zero). The command Esc Y is used to select screen positions and this also permits the text to be sent as part of the command. The second program section ends up as shown in **Figure 5**.

After the value has been sent, a few trailing spaces are added. Numbers in Flowcode are left justified. Since the screen isn't cleared, but is simply overwritten, a measurement of '8' that followed a measurement of '100' would be displayed as '800', because the two zeroes would not be overwritten and would therefore still be visible. Adding a few spaces at the end avoids this problem.

The next step is to determine the state of the reed switch. Although it is magnetically activated, technically it is no different from an ordinary switch. You can therefore use a standard Flowcode switch component. After adding this to the workspace the state of the switch can be put into a variable using the macro 'readstate'. We've used 'reedswitch' as the name for this variable. Instead of a 0 or 1 it looks a lot better if the display shows 'on' or 'off'. We use an if/then instruction to implement this. From Figure 3 you can see that the text starts at position &#, which makes the final part of the program look like **Figure 6**. The Flowcode design files can be downloaded free of charge from the Elektor website [2].

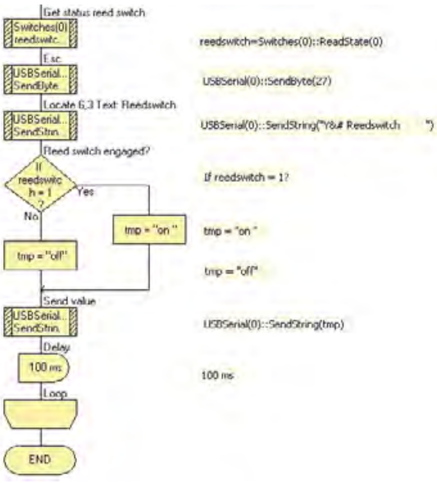
The microcontroller program is now complete and can be programmed into the ECIO40. Once this has happened you'll get a message from your PC stating that new hardware has been found. Ignore this message, but leave the window open! We will need this at a later stage.

USB driver

On the PC side we now need a USB driver that receives the data from the microcontroller via the USB port and then converts it into a virtual COM port, which is a COM port that doesn't exist in hardware, but still appears to exist to a program running on the PC. The program can communicate with this virtual COM port in exactly the same way as if it was a hardware based COM port.

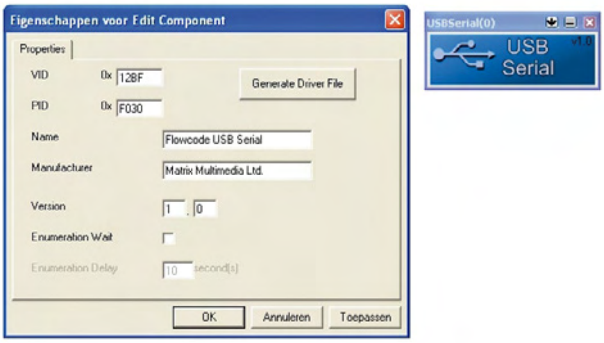
The USB driver is created using Flowcode. On most PCs you should be logged in as Administrator, or have administrative rights, in order to carry out the following steps.

Open the Flowcode USB component. We'll ignore the VID and PID for the time being. Next click on the button 'Generate driver file' (**Figure 7**). You can accept the suggested driver name or think of one yourself. The extension (.INF) is automatically added. Store the driver in the same folder where the rest of your project is stored. This isn't compulsory, but if you don't do this you could end up with a large number of drivers and forget which project they belong to.



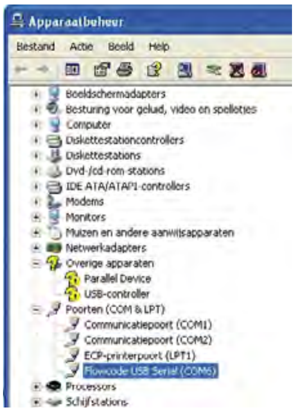
```
reedswitch=Switches(0)::ReadState(0)
USBSerial(0)::SendByte(27)
USBSerial(0)::SendString("Y&# Reedswitch      ")
If reedswitch = 1?
tmp = "on "
tmp = "off"
USBSerial(0)::SendString(tmp)
100 ms
```

Figure 6. Determine the state of the reed switch and send it via the USB link.



VID 0x 128F
PID 0x F030
Name: Flowcode USB Serial
Manufacturer: Matrix Multimedia Ltd.
Version: 1 . 0
Enumeration Wait: □
Enumeration Delay: 10 second(s)

Figure 7. Properties of the USB component in Flowcode.



- Beeldschermadapters
- Besturing voor geluid, video en spelletjes
- Computer
- Diskettestationcontrollers
- Diskettestations
- Dvd-/cd-rom-stations
- IDE ATA/ATAPI-controllers
- Modems
- Monitors
- Muizen en andere aanwijsapparaten
- Netwerkadapters
- Overige apparaten
  - Parallel Device
  - USB-controller
- Poorten (COM & LPT)
  - Communicatiepoort (COM1)
  - Communicatiepoort (COM2)
  - ECP-printerpoort (LPT1)
  - Flowcode USB Serial (COM6)
- Processors
- Schijfstations

Figure 8. The virtual COM port in the Device Manager.

If all is well, the window with 'found new hardware' should still be open. We can now continue with this. Don't give permission to connect to the Internet (your new driver will obviously not be found there) and on the next screen state that you want to install the driver from a specific location. Type in the folder where the Flowcode-generated driver is stored. It should then be installed automatically.

Wait until the driver has been completely installed. The computer should have automatically allocated a number for the virtual COM port. That number can be different for different PCs. Go to the Device Manager (via Control Panel -> System -> Hardware) and look which COM port uses this driver. For our PC it is COM6 (see **Figure 8**).

Start HyperTerm and enter the following communications settings:

● Connect using '6' in our case. (It could be different for you.)	
● Bits per second	38400
● Data bits	8
● Parity	none
● Stop bits	1
● Handshaking	none

You could use other speeds as well, without needing to change the Flowcode program. If you prefer to use 9600 baud, all you need to do is just fill it in. The driver and the USB port on the microcontroller will automatically adapt to this speed.
After HyperTerm is started you'll see the screen shown in **Figure 9**. When you rotate the potentiometer, or hold a magnet near the switch, you'll see the changes on the screen.

Error messages

If the screen isn't static, but starts scrolling, then from HyperTerm you should go to File -> Properties -> Settings and set the emulation to VT52. If you should see the message 'Invalid COM port' or similar appear in HyperTerm, you probably haven't connected the ECIO. The virtual COM port only exists when the ECIO is connected and the bootloader has completed. The correct order of events is therefore: first connect the ECIO, wait until the LED stops flashing and only then start up HyperTerm.

If the PC shows the message 'New hardware or driver not found' it may be caused by the following: when the ECIO is plugged into the USB port of the PC, the PC uses the VID and PID numbers (you will have come across these in the USB serial component) to search for the matching driver. If you have changed either of these numbers then the PC won't recognize the ECIO and you will have to create a new driver (or put the numbers back to their original value).

(080817-I)

Internet links

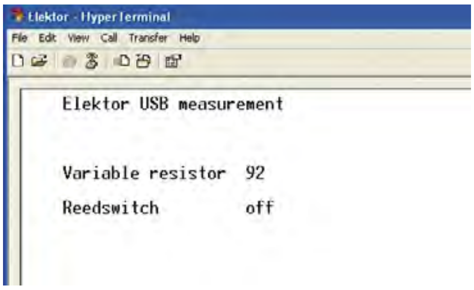
[1] www.matrixmultimedia.com/FlowcodeUSBPack.php

You should have the serial number from your Flowcode CD to hand. This pack contains examples of USB-slave and USB-HID connections including the Visual Basic 6 source code and an interface with LabVIEW.

[2] www.elektor-usa.com/080817

About the author

Bert van Dam writes books for Elektor: PIC Microcontrollers (50 JAL projects for beginners and experts), Artificial Intelligence (20 JAL projects to bring your microcontroller to life), Microcontroller Systems Engineering with Flowcode (45 Flowcode projects for ARM, PIC and AVR microcontrollers).



```
Elektor USB measurement


Variable resistor  92

Reedswitch         off
```

Figure 9. The end result.

16 from 4

Port expansion using 74HC595s

Frank Link (Germany)

Not having enough ports on a microcontroller to drive all the functions of a circuit is a commonly-encountered problem. The clever port expansion boards described here offer a solution, and need just four signals from the microcontroller. The first board provides sixteen digital output bits; the second board, equally simple to drive, can be used to interface to a HD44780-compatible LCD module. A C library is available to facilitate using the boards with AVR microcontrollers.

Features

- Port expansion from four outputs to sixteen outputs
- Additional circuit board to interface to HD44780-compatible LCD using four port pins
- Controlled using software SPI emulation with free choice of microcontroller output pins
- Boards may be cascaded
- Maximum total load on all outputs approximately 70 mA
- C library available for AVR microcontrollers to drive port expansion and LCD interface boards

Some time ago the author developed a board to drive a motorized potentiometer using an ATmega8. Several outputs of the microcontroller were wired in parallel to increase the current available and thereby avoid the need for an extra motor driver IC. Just five port pins were left over on the ATmega8, which made implementing the remaining functionality of the board rather tricky.

Undaunted the author carried out a thorough search of the Internet for simple ways to expand the number of available ports. Finally, however, he decided to design his own hardware- and software-based solution to the problem. The design was based on the well-known 74HC595 shift register, which can be driven using an SPI bus. Unfortunately, however, the SPI pins of the ATmega8 were already committed, and so the SPI protocol had to be implemented in software. The result of the work is described here: a solution which is suitable for all types of microcontroller, with four spare port pins required to drive the port expansion circuit.

We have made a C library available for free download: this supports AVR microcontrollers, but it is relatively straightforward to modify the source code to adapt it to other device families. The library allows the user to control not just one, but any number of connected 74HC595s!

Printed circuit boards

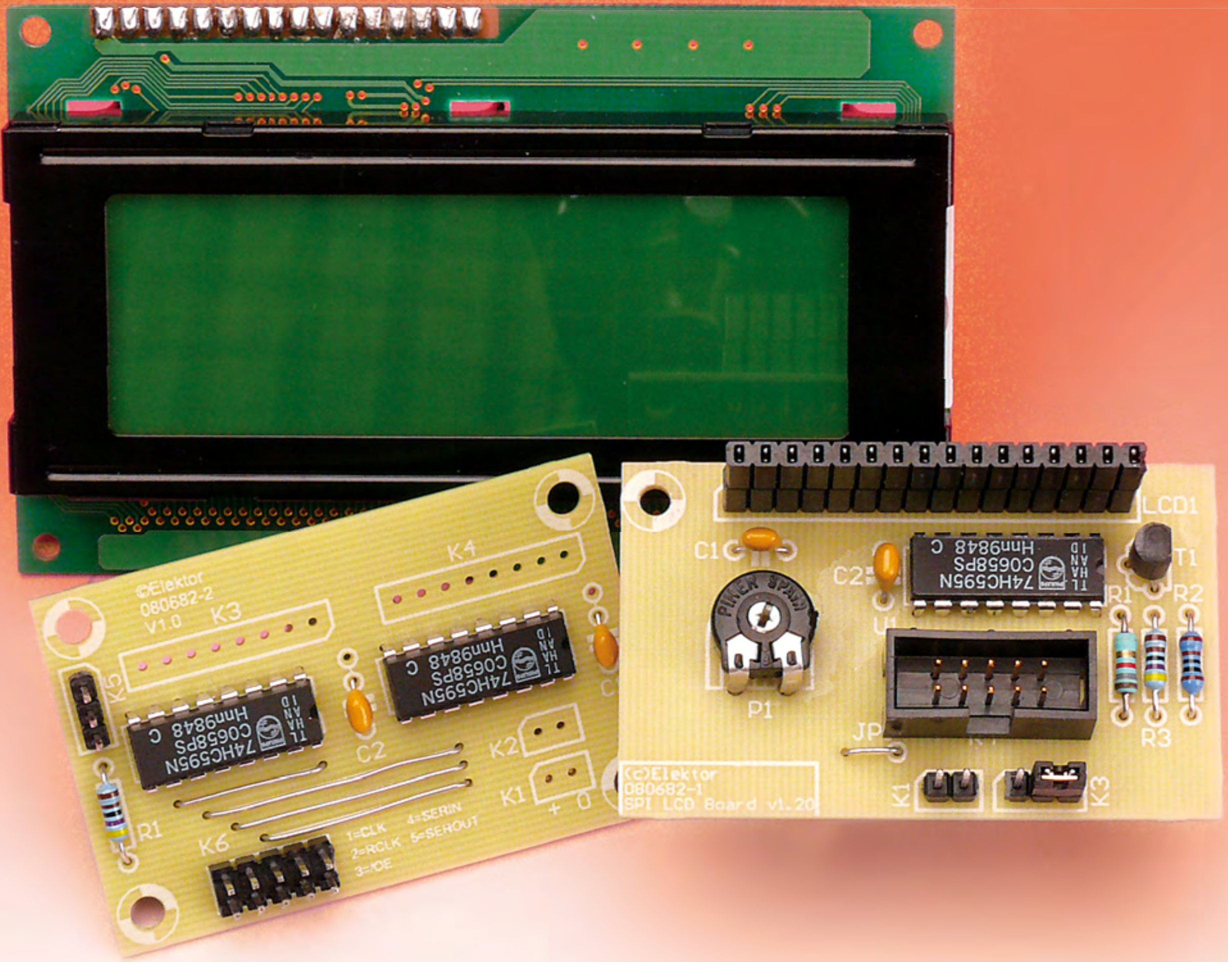
In the interests of ease of use the author produced two printed circuit boards, whose layout was subsequently refined at the Elektor labs. Both boards are available from Elektor [1].

The first printed circuit board is based around two 74HC595s. This provides a port expansion to sixteen output bits; the boards can be cascaded if even more outputs are required.

Figure 1 shows how the 74HC595 is driven. The SDI (serial data in) signal is used to send data to the device. With each pulse on SFTCLK (shift clock) the data bits are synchronously shifted one place along the registers. Then a pulse on LATCH CLOCK transfers the loaded data from a holding register in the 74HC595 into the output register, and the levels on the output pins change to reflect the transmitted data.

The $\overline{\text{OE}}$ signal is dedicated to a special function of the 74HC595. In order to avoid the situation where the outputs of the device go to an undefined state when power is applied, this signal can be held high using a pull-up resistor (via JP1 in Figure 1). This ensures that all the 74HC595's outputs go instead to a high impedance state when power is applied: a pull-up or a pull-down resistor can be connected to each output to make sure that it carries the desired logic level in this state.

If the jumper is not fitted $\overline{\text{OE}}$ can be driven from the microcontroller: this requires a fifth spare output pin, of



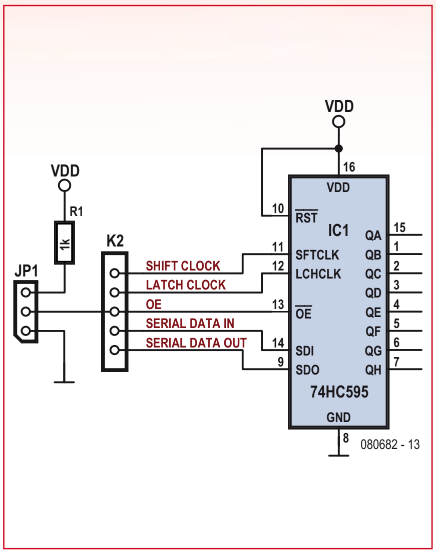


Figure 1. How to drive a 74HC595.

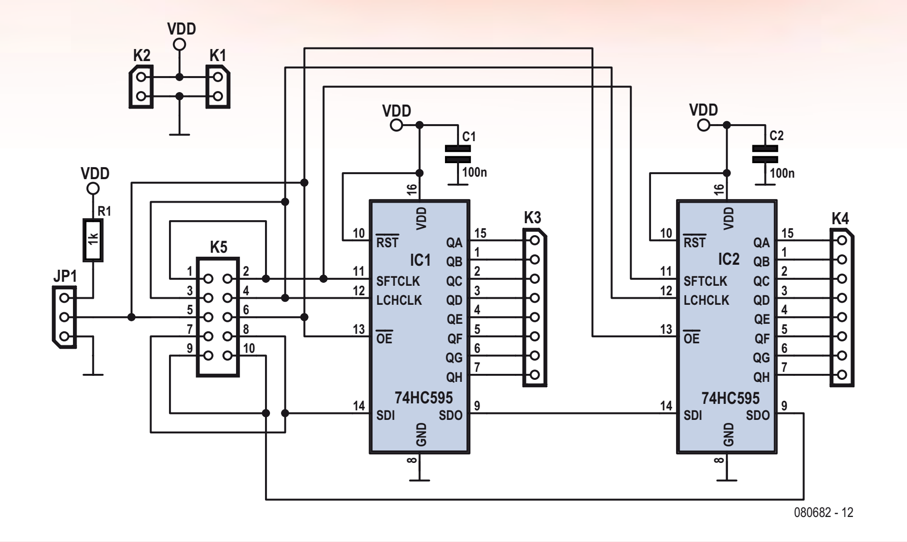


Figure 2. Circuit diagram of the port expansion unit.

course. Code to use OE when initializing the 74HC595 is implemented in the author's software library, but can be excised if it is not required.

Figure 2 and **Figure 3** show the circuit diagram and printed circuit board layout for the port expansion unit. Two 74HC595s are connected in cascade to double the number of available outputs. Cascade connection involves wiring the clock signals in parallel, and the input of the second 74HC595 to the output of the first. When a pulse is applied to SFTCLK, data bits are first shifted through the first device and then through the second.

The microcontroller is connected to K5. As all five pins are connected to both rows of the two-row header it is possible to connect the signals through to a second port expansion board and so use further 74HC595s. Note, however, that the SDI pin of the second board must be connected to the SDO pin of the first.

Power is supplied to the board via K1, and K2 allows this power to be delivered to cascaded boards. K3 and K4 carry the output signals from the 74HC595s, while JP1 determines how the OE signal is driven.

Software

As is conventional, the software library comes in two parts. The first is a header file com74hc595.h which includes function declarations and definitions of the port pins and num-

Listing 1. Function com74hc595_out()

```
void com74hc595_out()
{
unsigned char count = COM74HC595_SIZE;
unsigned char* serp = com74hc595 + COM74HC595_SIZE;
do
{
unsigned char bits;
/* fetch byte from array com74hc595[] */
unsigned char data = *--serp;
/* shift out 8 bits per byte */
for (bits = 8; bits > 0; bits--)
{
PORT_COM74HC595 &= ~(1<<PORT_SER);
if (data & 0x80)
{
PORT_COM74HC595 |= (1<<PORT_SER);
};
data <<= 1;
/* a pulse on the clock signal shifts the data synchronously */
/* by one place through all the shift registers */
PORT_COM74HC595 &= ~(1<<PORT_SCK);
PORT_COM74HC595 |= (1<<PORT_SCK);
}
}
while (--count > 0);
/* a pulse on RCK transfers the data from the shift re-
   gisters to the output latches */
PORT_COM74HC595 &= ~(1<<PORT_RCK);
PORT_COM74HC595 |= (1<<PORT_RCK);
}
```

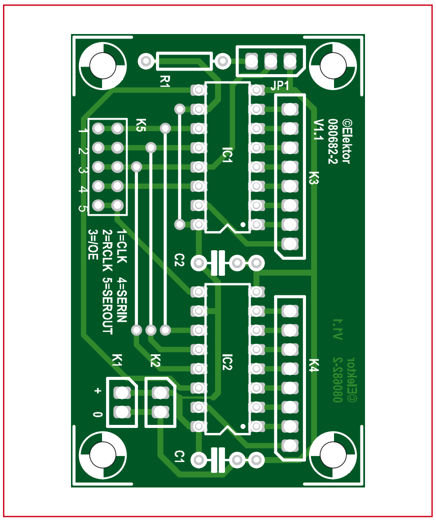


Figure 3. Port expansion printed circuit board.

COMPONENT LIST

LCD board

Resistors
R1,R2 = 1kΩ
R3 = 4kΩ7
P1 = 10kΩ preset

Capacitors
C1,C2 = 100nF

Semiconductors
T1 = BC557
IC1 = 74HC595

Miscellaneous
K1 = 2-way pinheader
K2 = 10-way pinheader (2x5)
JP1 = 3-way pinheader with jumper
PCB no. **080682-1** [1]

ber of 74HC595 devices connected (COM74HC595_SIZE). The second is a .c file containing the C source code that actually drives the ports. The header file will generally need to be modified by the user to specify which port pins on the AVR microcontroller are connected to which signal on the 74HC595 board. There is one restriction: the pins used must all belong to the same port, defined in the header file as PORT_COM74HC595.

The application program must first call the function com74hc595_init(). This initializes all the variables and constants used within the library. When this call completes all the 74HC595 outputs will have been set low.

The functions com74hc595_setBit() and com74hc595_unsetBit() address the individual output pins. Setting bits according to a specified bit pattern can be achieved using the functions com74hc595_setPort() and com74hc595_unsetPort(). Functions com74hc595_setall() and com74hc595_unsetall() can be used to set all port pins high or low respectively with a single call.

The actual work of driving the 74HC595s is carried out in the function com74hc595_out(): see **Listing 1**. It synchronously clocks the individual bits, stored in a character array called com74hc595[], through the chain of 74HC595 devices. The operation proceeds in reverse order, starting with

Listing 2. Simple running light

```
#include <avr/io.h>
#define F_CPU 3686400
#include <util/delay.h>
#include “com74hc595.h”
int main(void)
{
   com74hc595_init();
        while(1)
        {
        for ( int i = 0; i < com74hc595_BYTES; i++ )
        {
               com74hc595_setBit( i );
               com74hc595_out();
               _delay_ms( 2000 );
               _delay_ms( 2000 );
               com74hc595_unsetBit( i );
               com74hc595_out();
               _delay_ms( 2000 );
               _delay_ms( 2000 );
        }
   }
   return 0;
}
```

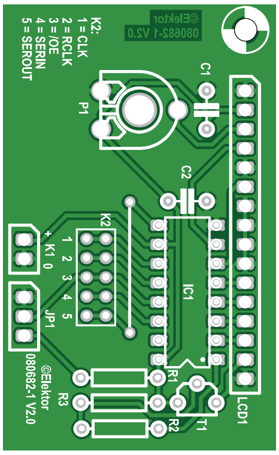


Figure 4. Printed circuit board for the LCD interface circuit.

COMPONENT LIST

Port expansion board

Resistors
R1 = 1kΩ

Capacitors
C1,C2 = 100nF

Semiconductors
IC1,IC2 = 74HC595

Miscellaneous
K1,K2 = 2-way pinheader
K3,K4 = 8-way pinheader
K5 = 10-way pinheader (2x5)
JP1 = 3-way pinheader with jumper
PCB no. **080682-2** [1]

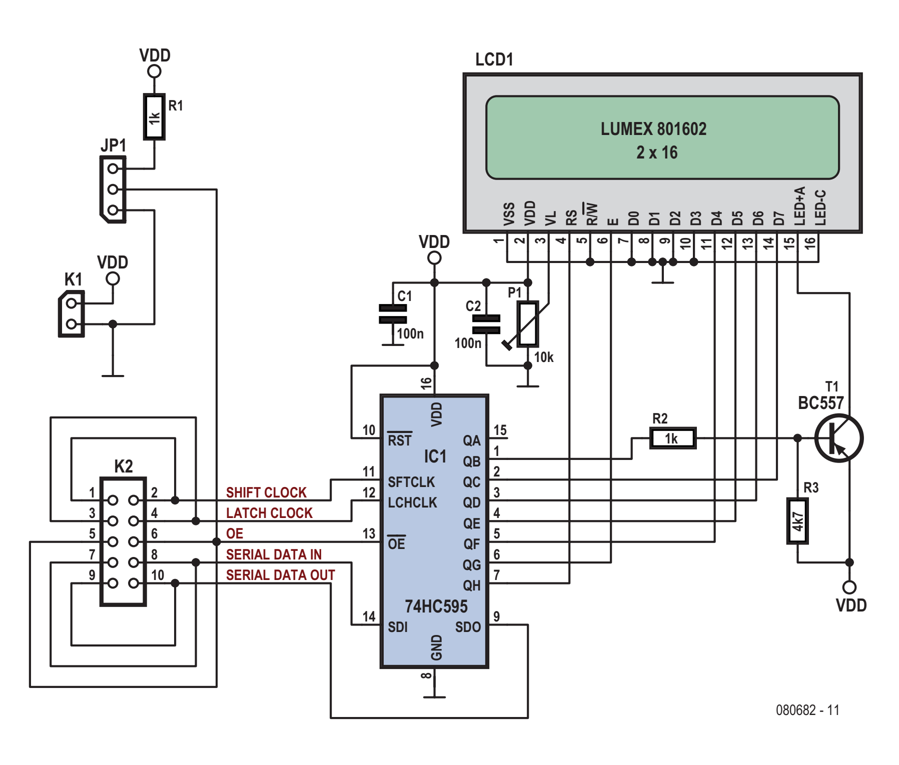


Figure 5. Circuit diagram of the LCD interface circuit.

the last bit. One bit is transferred into the chain by taking the clock signal low and then back high. When all bits have been sent, RCK is taken high and then back low.

A simple running light application example is shown in **Listing 2**.

LCD interface

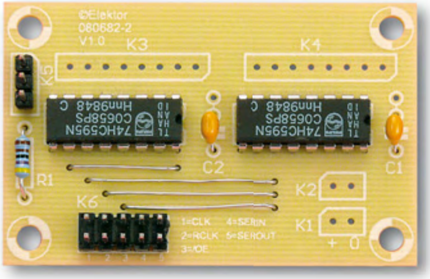
It is very common in microcontroller-based applications to want to display the values of internal variables. This gives rise to the idea of controlling a HD44780-compatible LCD module over just four wires with the help of a 74HC595. In addition, we can also allow the microcontroller to switch a backlight on and off.

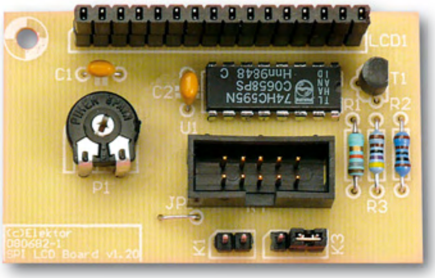
The resulting circuit is shown in **Figure 4** and **Figure 5**. The LCD is connected to the outputs of the 74HC595: as the module is to be used in four-bit mode, pins D0 to D3 of the LCD are connected to ground. The functions of K2, K1 and JP1 are as described above for the port expansion board.

The 10 kΩ trimmer should be soldered to the track side of the board to simplify adjusting the LCD contrast after assembly. All the other components are hidden underneath the LCD module.

The author has also written functions to control the LCD and made them into a library. Again this comprises a header file (com74hc595_LCD.h) and the main C code file. The central functions are com74hc595_lcd_nibble() and com74hc595_lcd_enable(); all the other functions are built upon these routines.

In com74hc595_lcd_nibble() (**Listing 3**) the four data bits are first set to zero. Each bit is then checked to see if it needs to be set: the backlight flag must be included in the same bit array, and so its state must also be tested and the appropriate bit set if necessary. Finally the information is sent to the 74HC595 using com74hc595_out().





The function com74hc595_lcd_enable() (**Listing 4**) tells the HD44780-compatible LCD controller that new data bits are available. This is done by setting the LCD's enable signal (EN) high for 20 μs and then low again.

Before the library can be used a couple of settings need to be made in the header file [1]. The first is the clock frequency at which the microcontroller is operating: this is needed to ensure that the timing for the LCD can be calculated exactly. If the LCD board is used as one member of a chain on 74HC595s, the macro LCD_PORT needs to be adjusted to reflect its position within the chain. It is also of course necessary to specify which port pins are to be used for the SPI port.

Listing 5 shows our version of the familiar 'Hello World' example program, demonstrating correct initialization of the two libraries and subsequent use of the LCD driver functions. The full range of commands available in the library is described in the supplementary documentation available for download (along with source code) from the project pages accompanying this article [1]. The author is continuing to work on further expansion plans, and interested readers are invited to contact him by e-mail at FrankLink61@aol.com.

(080682-I)

Internet Link

[1] www.elektor-usa.com/080682

Listing 3. Function ..lcd_nibble()

```
void com74hc595_lcd_nibble ( unsigned char d )
{
   com74hc595_unsetBit( LCD_B4 );
   com74hc595_unsetBit( LCD_B5 );
   com74hc595_unsetBit( LCD_B6 );
   com74hc595_unsetBit( LCD_B7 );

   if ( d & 1<<4 ) com74hc595_setBit( LCD_B4 );
   if ( d & 1<<5 ) com74hc595_setBit( LCD_B5 );
   if ( d & 1<<6 ) com74hc595_setBit( LCD_B6 );
   if ( d & 1<<7 ) com74hc595_setBit( LCD_B7 );

   if ( BackLightState == ON ) com74hc595_unsetBit( LCD_LIGHT );
   if ( BackLightState == OFF ) com74hc595_setBit( LCD_LIGHT );

   com74hc595_out();

        com74hc595_lcd_enable();
}
```

Listing 4. Function ..lcd_enable

```
void com74hc595_lcd_enable(void)
{
   com74hc595_setBit( LCD_EN );
   com74hc595_out();
   _delay_us(20);
   com74hc595_unsetBit( LCD_EN );
   com74hc595_out();
}
```

Listing 5. 'Hello World!'

```
#include <avr/io.h>
#include "com74hc595.h"
#include "com74hc595_LCD.h"

int main(void)
{
com74hc595_init();
        com74hc595_lcd_init();
        com74hc595_lcd_light_on();
        com74hc595_lcd_data('T');
        com74hc595_lcd_data('e');
        com74hc595_lcd_data('s');
        com74hc595_lcd_data('t');
        com74hc595_set_cursor(0,2);
        com74hc595_lcd_string("Hello World!");
        while(1)
        {
        }
     return 0;
}
```

i-TRIXX

Electronics inside out!

The free e-magazine about internet, computers, hacking, tweaking, modding, gadgets, geekstuff, gaming and DIY electronics. The e-zine you have to check out now!

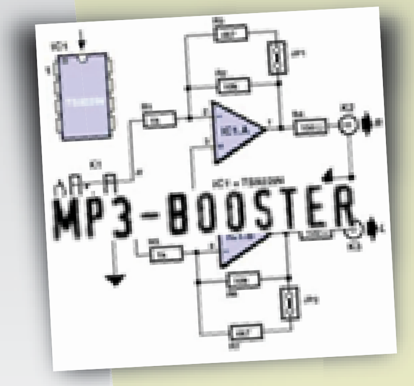
Receive i-TRIXX in your mailbox

FOR FREE !

each Wednesday

Play the i-TRIXX Quiz and win!

Test your knowledge about internet, computers and electronics. Take your chance now and play the i-TRIXX Quiz. There are great prices to win!



From the Elektor labs:

Simple, useful and fun electronic circuits!

Check **i-TRIXX.com** and subscribe now!

Powered by elektor electronics worldwide

Hyper Flea Mart!

Seemed like it was raining com

Antoine Authier (Elektor Labs)

深圳电子市场

I had to go twice to China, to launch and then monitor volume production of the Elektor Reflow Oven. When I visited the city of Shenzhen, a real world center for the production of electronic appliances, one place I visited is going to haunt my memory as an electronics technician for a long time: an electronic component market.

How can I describe such a place? Market, big store, hypermarket… At first sight, the way the merchandise is displayed in bulk brings to mind the profusion of a fruit and vegetable market, with its varying degrees of order — or disorder. In the center of the building, you find all grouped together the passive components — resistors, inductors, and capacitors — together with crystal oscillators and certain cheap discrete semiconductors like triacs, transistor, LEDs, and so on, along with mechanical components like potentiometers, push-buttons, switches, etc., and spacers, washers and screws. Here and there near the entrances, there are a few tools on offer, like for example multimeters and soldering irons.

… I get lost. It's a jungle!

Higher added-value semiconductors, hard to find singly but readily available on reels or in trays, are mainly sold from small glazed stands round the edge of the building.
The layout, abundance, and diversity of the components brings to mind a Wallmart-like hypermart. The prices are incredibly low. Rumor has it that certain of these market traders are selling empty cases! I don't know if that's true, but in any event, the oscillators I found there for one of our forthcoming projects work perfectly.
Once over this initial wonderment, a little pause is called for: I spot the escalators, allowing me to gain a little height and discover the immensity of the place I'd just been touring. The visit goes on and on, floor after floor, and in the absence of comprehensible directions for finding this or that product, I get lost. It's a jungle!

Even though it's all a bit like a department store, it seems to me jolly badly organized. I pop out for a moment for a breath of air — no less polluted outside than it is stale inside. Once outside, you only have to cross the road to enter another building housing another component market just like the first. The center of Shenzhen comprises a whole block of immense skyscrapers, totally dedicated to the sale of electronic equipment.
Everything imaginable is sold there, from ancient transistors in TO3 packages to the latest ARM core, via drums of cable, OEM modules, measuring instruments… For Western visitors (if they don't speak Chinese), the difficulty is finding things…
The market sellers are usually crammed into a tiny square shop, behind a transparent counter displaying their main lines and leading products, surrounded on three sides by overloaded shelves. Thankfully, sales don't take place at the top of people's voices — the atmosphere is rather calm. Despite an impressive number of air-conditioning units, the air is hot and heavy.

Business is brisk, but in the endless series of shops, it's not uncommon to see stallholders who've fallen asleep, or young mothers occupied with their babies. When they're not patiently waiting for customers, the traders can often be seen playing on their portable computers, which take the place of cash registers.
It all makes a fascinating sight. I saw, for example, one young woman who was sorting oscillators in tiny SMD packages at an improbable speed, without ever making a mistake.

So just who are the customers that come here? Hard to tell. There are lots of small electronics production companies in this area; I imagine that their proprietors come to buy components for their daily production. You also find a few tourists, who are easy to spot!
In these markets, just like everywhere in Asia, you can eat and drink at any time of the day. In Summer, the heat is intense, so it's a good idea to come armed with a small fan and some cold drinks. A notepad and pen will prove very useful — along with a good sense of direction!
Communicating in English with the Chinese in China, especially in a place like this, seemed to me very hard if not impossible. An inter-

ponents…

preter is really a must for anyone who doesn't (yet) have a command of Mandarin.

In Hong Kong, there aren't any of these component markets. Not being terribly profitable any more, this business has moved from the much more highly developed megapolis to nearby mainland China. On the other hand, there are whole streets, in particular Apliu Street in the Sham Shui Po district of Kowloon, of shops for electronics tools, test instruments, and finished products: valve amplifiers, watches, mobile phones, computers, and spare parts for them. Prices here are comparable with those in the US or Europe — but of course, less tax.

Western visitors steeped in electronics should certainly not miss out on the profound impressions left by a few hours of total immersion in one of these Chinese electronic components markets like the ones in Shenzhen. It's an unforgettable cultural and technical experience.

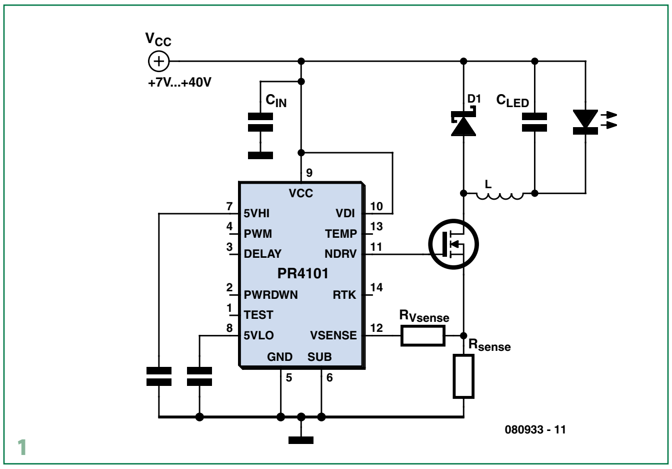
(081192-I)

PR4101 Dimmable Power LED Driver

Burkhard Kainka (Germany)

Semiconductor firm PREMA [1], based in Mainz in Germany, will be well known to many *Elektor* readers for their PR4401 device [2] which formed the basis of the LED driver board that we gave away with the September 2007 issue of the magazine. That device is capable of powering a white LED from a single cell. More recently the company has produced the PR4101, a step-down driver designed for power LEDs. It requires an input voltage of between 7 V and 40 V and at its output can drive one or more LEDs at constant current, irrespective of fluctuations in this voltage. The drive current is relatively smooth, which is beneficial for the life of the LED. The external components required are a power MOSFET, a coil, a Schottky diode and a couple of resistors and capacitors. Output currents in excess of 1 A are possible.

The PR4101 is suitable for applications running from 12 V or 24 V automotive batteries, as well as for situations where it is desired to replace a 12 V halogen bulb powered from a mains transformer. In this case, as with halogen bulbs, the IC allows dimming using phase angle control. Further special features of the device include an overtemperature detection circuit that is also capable of protecting the LEDs using an external temperature sensor, reducing the drive current at higher temperatures. An external control input allows the device to be powered down, for example by a remote control, with a standby current of less than 35 µA.



1

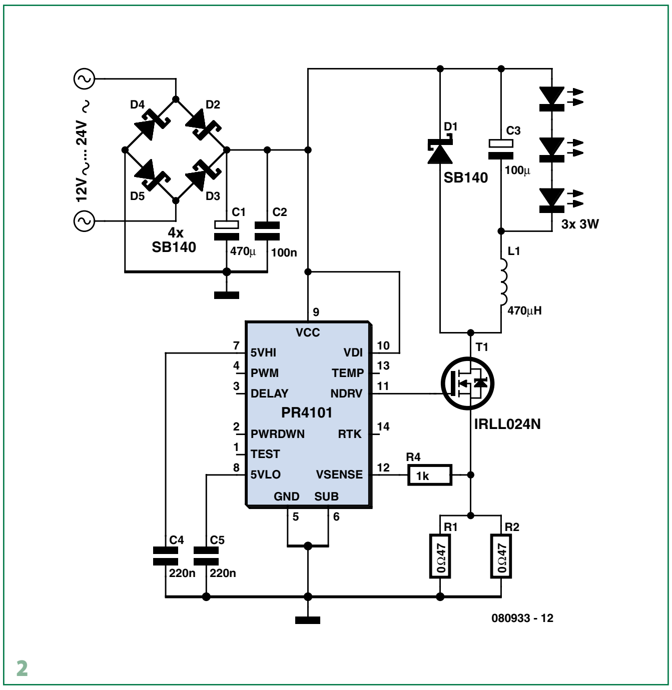
The LED current can be set using either an external sense resistor or a pulse-width modulated signal at the device's PWM input. Conversion efficiency in a typical 12 V application with three 3 W LEDs wired in series is over 90 %; in the least favorable case (an input voltage of 40 V, driving a single 3 W LED) efficiency is still greater than 60 %.

Figure 1 shows the basic PR4101 circuit. The LED current is determined by the value of the current sense resistor R_{sense}, while we have fixed the value of resistor R_{Vsense} at 1 kΩ. R_{sense} is selected using the equation $R_{sense} = 0.2\ V / I_{LED}$. Typical example values are 0.27 Ω for 3 W LEDs (I_{LED} = 750 mA) and 0.56 Ω for 1 W LEDs (I_{LED} approximately 350 mA).

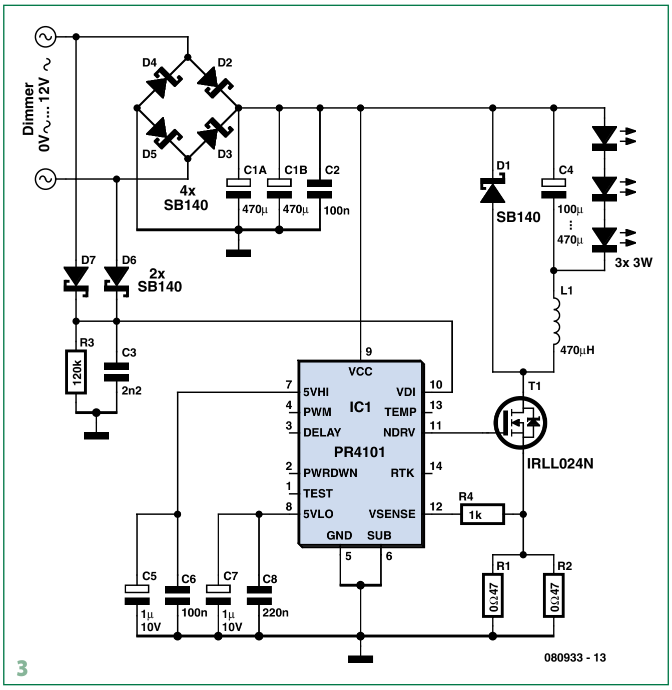
Figure 2 shows a circuit designed for direct connection to a halogen bulb transformer, which can either be an ordinary iron-core transformer or an 'electronic transformer' (see the article 'Electronic Transformers Revealed and Explained' in the December 2008 issue). Normally these deliver an AC voltage at 12 V, although the circuit shown will operate with input voltages of up to 24 V. Thanks to the use of Schottky diodes in the bridge rectifier and a 470 µF electrolytic for smoothing, a 12 VAC input is enough to drive up to three 3 W LEDs connected in series. The parallel combination of R1 and R2 forms an effective current sense resistor of 0.47 Ω / 2, or approximately 0.24 Ω. This sets the LED current at around 850 mA.

The circuit in **Figure 3** sports an additional rectifier in the form of D6 and D7. These allow the brightness of the connected LEDs to be controlled by a dimmer switch employing either a conventional thyristor or TRIAC arrangement or an electronic dimmer or electronic transformer. The PR4101 can be used with dimmers that use either leading-edge or trailing-edge phase control. The undervoltage detection input VDI is fed from the AC input, rectified by D6 and D7 but not smoothed. The device can thus observe the mark-space ratio of the dimmer control waveform and switch the converter on and off in synchrony. The result is that the LED's brightness can be controlled just as if it were an incandescent lamp.

In the PR4101 datasheet [3] PREMA describe a demonstration board for the device that allows



2



3

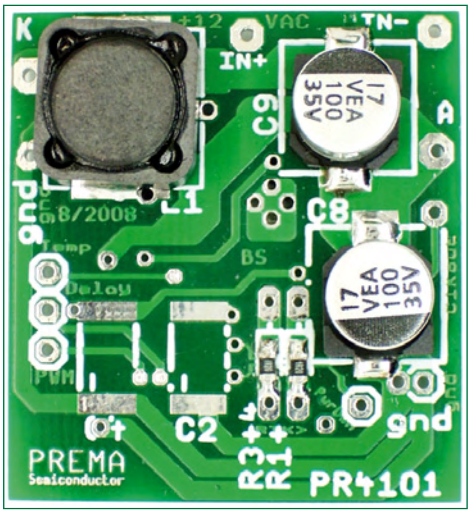
all its features to be exercized. These include an integrated temperature sensor, temperature compensation of the LED current, and a soft-start function. The circuit of the demonstration board is broadly the same as Figure 3, and is suitable for demonstrating brightness control using a halogen bulb transformer with phase control dimming on the mains side.

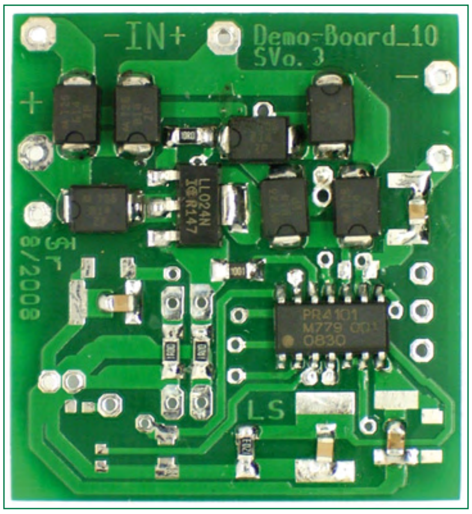
(080933-I)

[1] www.prema.com

[2] www.elektor.com/070100

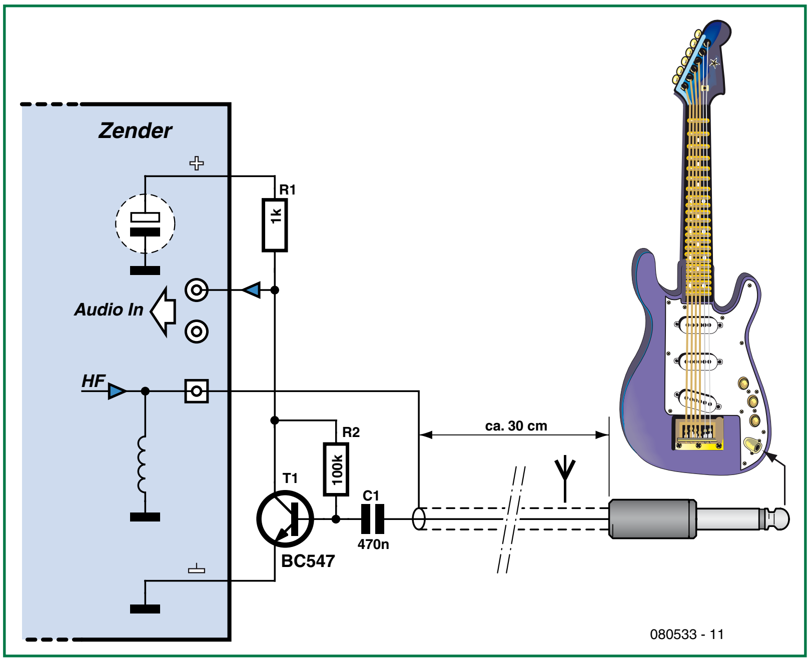
[3] www.prema.com/pdf/pr4101.pdf





Simple guitar transmitter

Gert Baars (The Netherlands)



To be able to play an electric guitar you need, in addition to the guitar itself, a guitar amplifier. The guitar then needs to be connected with a cable to the amplifier, which could be considered an inconvenience. Most guitar amplifiers operate off the AC power line.

An electric guitar fitted with a small transmitter offers several advantages. The guitar can be made audible via an FM tuner / amplifier, for example. Both the connecting cable and amplifier are then unnecessary. With a portable FM broadcast radio or, if desired, a Gettoblaster you can play in the street or in subway stations (like Billy Bragg) and everything is then battery powered and independent of a fixed power point. You may need a permit though.

Designing a transmitter to do this is not necessary at all. There are currently various so-called Walkman/MP3 player transmitters available at very low cost. The range of these devices is often not more than 10 meters, but that's plenty for our application.

For this purpose we bought such a transmitter, the brand name of which is König and the model type is FMtrans20. After fitting the batteries and turning the transmitter on, a carrier signal can be detected on the radio. Four channels are available, so it should always be possible to find an unused part of the FM band.

A short cable with a 3.5 mm stereo audio jack protrudes from the enclosure. This is the audio input. Having taken some measurements, it appeared that the required signal level for sufficient modulation is about 500 mV$_{pp}$. If a guitar is connected directly, the volume level of the radio will have to be turned up very high to get sufficient sound, so high that the noise from the modulator becomes quite annoying. A preamplifier for the guitar signal is therefore essential.

To build this preamplifier into the transmitter we will first have to open the enclosure. In our unit this is done with four screws in the battery compartment. After opening the enclosure and a little investigation it appears that the two audio channels are combined. This is therefore a single channel (mono) transmitter. Because the audio preamplifier can be turned on and off at the same time as the transmitter, we can use the 'on-board' power supply of the transmitter for our power supply as well. In our case that was about 2.2 V. This voltage is available at the positive terminal of an electrolytic capacitor. 2.2 Volts is not enough to power an opamp. But with a single transistor the gain is already big enough and the guitar signal is sufficiently modulated.

The final implementation of the modification involves soldering the preamplifier circuit along an edge of the PCB so that everything still fits inside the enclosure. The stereo cable is replaced with a 30 cm long microphone cable, fitted with a guitar plug (mono jack). The screen braid of the cable acts as an antenna as well as a ground connection for the guitar signal. The coil couples the low-frequency signal to ground, while it isolates the high-frequency antenna signal.

While playing the cable with the transmitter just dangles below the guitar, without being a nuisance. If you prefer, you could also secure the transmitter to the guitar with a bit of double-sided tape.

(080533-I)

Hexadoku

Puzzle with an electronics touch

Hopefully you're ready to get cracking again with a Hexadoku puzzle freshly brewed for you. Get out your pencil, scrapbook or note paper (not a notebook), take a deep breath and participate! All correct solutions we receive enter a prize draw for an E-blocks Starter Kit Professional and three Elektor Shop vouchers. Have fun!

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 **$425**

and three **Elektor SHOP Vouchers** worth $55.00 each.

We believe these prizes should encourage

all our readers to participate!

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

PARTICIPATE!

Please send your solution (the numbers in the grey boxes) by email to:
hexadoku@elektor-usa.com - Subject: hexadoku 03-2009 (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 April 2009.**

PRIZE WINNERS

The solution of the January 2009 Hexadoku is: **4395C**.

The **E-blocks Starter Kit Professional** goes to: George Hardill (UK).

An **Elektor SHOP voucher worth $55.00** goes to: Jean-Louis Vidaud (France); Bernt Hörmann (Germany); Carsten Bohemann (Germany).

Congratulations everybody!

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

(c) PZZL.com

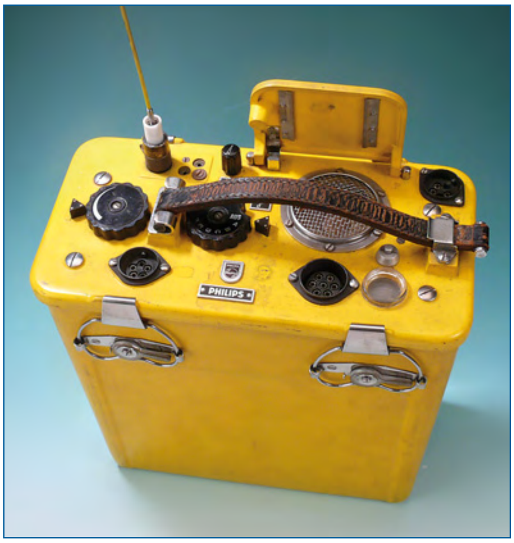
A	5	C	8	B	E	2	9	F	6	0	3	1	7	4	D
2	0	D	3	7	4	F	8	C	B	1	E	A	5	6	9
B	E	9	7	1	3	D	6	5	4	A	8	2	F	0	C
4	1	6	F	0	5	A	C	9	2	7	D	8	3	E	B
9	B	4	5	C	8	E	A	6	7	F	1	D	0	2	3
C	7	2	E	6	D	1	4	A	5	3	0	B	9	8	F
8	A	F	1	3	0	B	2	D	E	4	9	5	C	7	6
3	D	0	6	9	F	7	5	2	C	8	B	E	4	1	A
1	9	5	B	F	2	4	7	0	8	6	C	3	D	A	E
6	4	A	0	5	C	3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

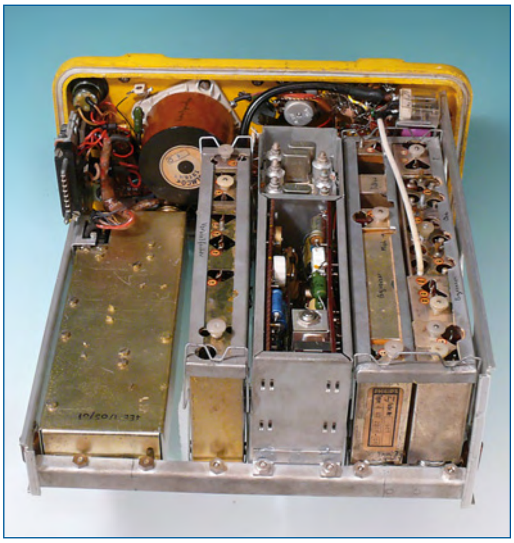
Elbe-Weser marine pilot transceiver (ca. 1960)

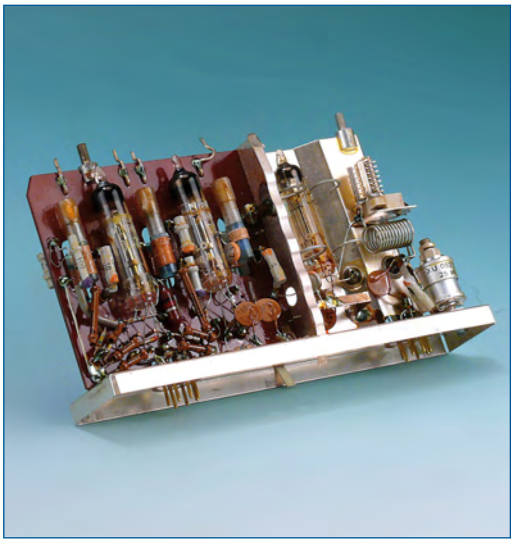
Jan Buiting (Elektor UK/US Editorial)

Nothing, zilch, nada from Google on this bright yellow, portable, NBFM 70 MHz receiver/transmitter from Philips duly labelled “Fabr. Nr. LO 601” and “Type 4RR105-01”. It goes by two other names as well — having peeled off a sticker applied over another one on the side of the chassis, I was able to read ‘LO 777’ and ‘4RR110-01’ in the respective boxes. Fortunately, the radio is not a total mystery. My archives indicate that it was used by pilots assisting with ship navigation around the light vessel ‘Elbe 1’ in the (huge) Elbe-Weser area in northwest Germany. In the early 1960s the busy ship traffic in this area was controlled by a marine operational centre in Cuxhaven using a 300-feet high mast and an advanced shore-base radar system supplied by Philips.

The transceiver measures 300×270×130 mm and should weigh about 7 kg with its NiCd battery pack installed. The case is made from polyester resin reinforced with glass fabric. It is claimed to be resistant to moisture, even sea water. The case is opened by turning four levers inside chrome-plated clamp rings on the sides.







Greeted by the smell of pertinax and solder rosin so typical of 1950s/60s radio gear, my immediate impression is that although the equipment construction is tidy to paranoid levels, this is not a product from the Philips Telecommunications Industry (PTI) factories in Holland. The clues are in the mixed brands of components used: an Ohmite power resistor, Eroid and Erofoil capacitors, a Preh pot, Valvo semiconductors and a Holmco loudspeaker. None of these would have been used by PTI Holland in the 1960s — **all** components would be Philips’ own make. My best guess is that the LO 601 (or 777) was commissioned for production by Lorenz and then rebadged for Philips Germany.

The next surprise is that the ceramic 7-pin tube sockets fitted on the chassis do not hold tubes but serve as through connects to circuitry neatly packaged in modules! Inside the modules I found ‘pencil’ tubes of the Philips DF/DL series. These are also known as ‘battery tubes’ because they were designed for use in battery-operated MW/LW/FM radios. The letter ‘D’ indicates a directly heated cathode for 1.2-VDC operation (yes, that’s five in series at 6 volts).

The receiver is a crystal controlled double superheterodyne with unknown first and second IFs, employing a 7.5435 MHz xtal oscillator for the second mixer. Only the AF section is transistorised with a few OC (Germanium) transistors.

The transmitter is a conventional design too with crystals, frequency multiplier stages and a simple RF power amplifier — all with just four DF73 miniature pencil tubes.

The +70 V and +150 V HT supply is a conventional power multivibrator using two OC16s, two transformers and a selenium bridge rectifier — electronics’ foulest smelling part when it burns out!

The controls on the radio have a ‘feel’ no longer encountered on modern equipment. Tactile and audible feedback (‘clicks and springs’) reign supreme here and the general impression is one of extreme sturdiness and attention to ergonomic detail. This is helped by precision engineered constructions like the channel selector knob that also acts as the on/off switch. The 10×75 mm hinged lid on the loudspeaker grille has a stud on it that drops into a matching opening in the channel knob when this is turned to the Off position.

Remarkably the radio has a 240 Ω balanced cable connection right beside to the expected (50 Ω unbalanced) SO239 socket, which takes a ¼ wave whip antenna.

Unfortunately the radio is missing its original PTT microphone and the battery charger cable. Still, the yellow box came alive with a characteristic inverter whine and a loud FM hiss when connected to a 6 volts, 2 amps power supply. On channel C (71.7500 MHz simplex) I was able to measure 750 mW of transmit power and about 1.5 μV receiver sensitivity (for 12 dB S/N). Enough to cover up to six to seven miles out at sea. If you can help me find the service documentation or the microphone set for this early ‘handheld’, please let me know.

(080962-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**

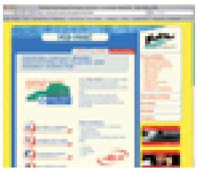
AVIT RESEARCH

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.

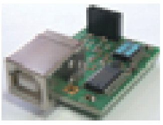
BETA LAYOUT

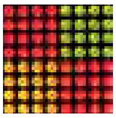


www.pcb-pool.com

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

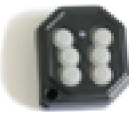
- USB to I2C
- Microcontrollers
- Forth
- Serial Devices

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-usa.
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

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

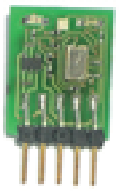
FIRST TECHNOLOGY TRANSFER LTD.

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:

- 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

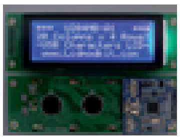
FTDI designs and sells USB-UART and USB-FIFO interface i.c.'s.
Complete with PC drivers, these devices simplify the task of designing or upgrading peripherals to USB

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

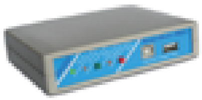


http://www.lcdmodkit.com

Worldwide On-line retailer

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

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

products and services directory

www.elektor-usa.com

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

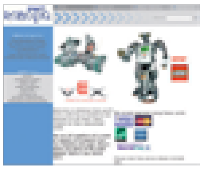
http://www.robot-electronics.co.uk

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

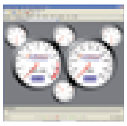
Internet Technical Bookshop,
1-3 Fairlands House, North Street, Carshalton,
Surrey SM5 2HW
email: sales@robotiq.co.uk Tel: 020 8669 0769

SCANTOOL.NET

http://www.scantool.net

ScanTool.net offers a complete line of PC-based scan tools.

- 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-usa.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/W
- 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 $8.00 DISCOUNT

www.elektor-usa.com/offer

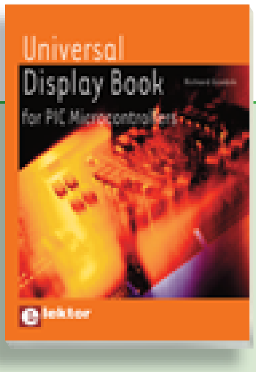
Books

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. For example a silent alarm, a people sensor, a radar, a night buzzer, a VU meter, a RGB fader, a serial network, a poetry box and a sound super-compression. You can build 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. Even after you've built all the projects it will still be a valuable reference guide to keep next to your PC. You can also use this book 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. This way you'll learn a lot about the project and the microcontroller being used, and you can expand the project to suit your own needs…making it ideal for use in schools and colleges.

446 pages • ISBN 978-0-905705-70-5 • US $52.00



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 $39.90

Visual Basic
5.0, 6.0, VBA, .NET, 2005
Vincent Himpe
for Electronics Engineering Applications
elektor

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 $56.00

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

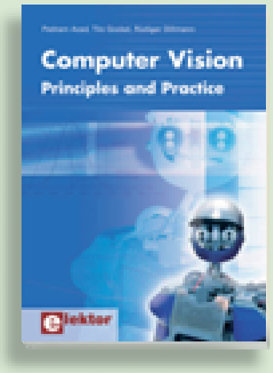
Microcontroller Systems Engineering
45 projects for PIC, AVR and ARM
elektor
Bestseller!

45 projects for PIC, AVR and ARM

Microcontroller Systems Engineering

This book covers 45 exciting and fun projects for PIC, AVR and ARM microcontrollers. Each project has a clear description of both hardware and software with pictures and diagrams, which explain not just how things are done but also why. As you go along the projects increase in difficulty and the new concepts are explained. You can use it as a projects book, and build the projects for your own use. Or you can use it as a study guide.

336 pages • ISBN 978-0-905705-75-0 • US $52.00

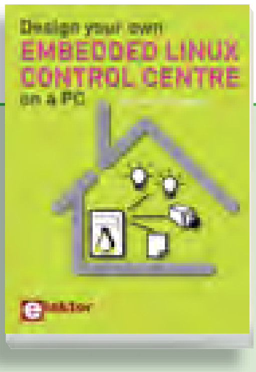


Principles and Practice

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 $58.00



A DIY system made from recycled components

Design your own Embedded Linux control centre on a PC

This book covers a do-it-yourself 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 $42.00

More information on the Elektor Website:

www.elektor-usa.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

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 $29.00

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 $39.00

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 $38.00

All articles published in 2008

DVD Elektor 2008

This DVD-ROM contains all editorial articles published in Volume 2008 of the English, Spanish, Dutch, French and German editions of Elektor magazine. Using Adobe Reader, articles are presented in the same layout as originally found in the magazine. The DVD is packed with features including a powerful search engine and the possibility to edit PCB layouts with a graphics program, or printing hard copy at printer resolution. Thanks to the use of an Internet browser program, the requirements for the PC are minimal.

ISBN 978-90-5381-235-8 • US $35.00

Bestseller!

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 $145.00

DCC Command Station

Electronics is making more and more inroads into the domain of model trains. Trains are now controlled with digital codes, and in many cases the entire system can be operated from a computer. Elektor presents a design for the device that forms the heart of a digitally controlled model railway: the DCC Command Station. The computing power in this design is provided by a highperformance ARM7 processor.

Kit of parts incl. programmed ARM module

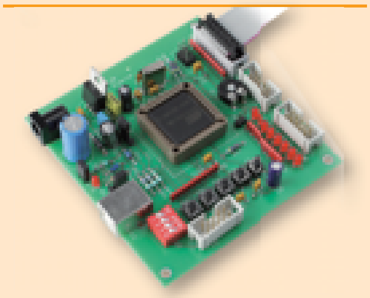
Art. # 070989-71 • US $162.00

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 $119.95



USB Flash Board

Flash microcontrollers are easy to program. In the past, program code was usually downloaded via a serial interface, but nowadays many PCs (especially laptops) only have USB ports. This versatile Flash Board provides a solution to this problem. It is built around an AT89C5131A, which is an extended 8051-family microcontroller with an 80C52 core and a Full Speed USB port. As a sort of bonus, the IC has a complete update interface for downloading new firmware. Atmel also provides suitable software in the form of its FLIP program, which is available free of charge.

Kit of parts incl. PCB and all components

Art. # 070125-71 • US $72.50

Kits & Modules

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

Product Shortlist

March 2009 (No. 387)

		US $
M16C TinyBrick		
080719-91	Kit of parts: TinyBrick-PCB with SMD parts and microntroller premounted; plus all other parts	87.50

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	82.50

January 2009 (No. 385)

		US $
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)

		US $
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)

		US $
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)

		US $
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

Bestsellers

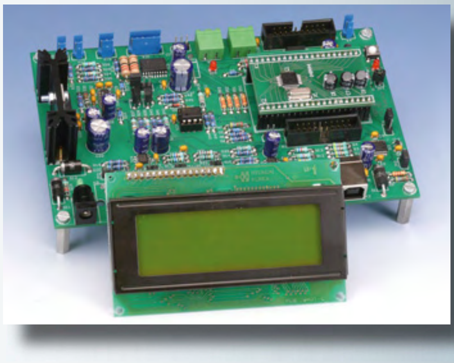
Books	1	→	**Embedded Linux Control Centre** ISBN 978-0-905705-72-9	US $42.00
	2	↗	**PIC Microcontrollers** ISBN 978-0-905705-70-5	US $52.00
	3	↑	**Microcontroller Systems Engineering** ISBN 978-0-905705-75-0	US $52.00
	4	↘	**Universal Display Book** ISBN 978-0-905705-73-6	US $39.90
	5	→	**Computer Vision** ISBN 978-0-905705-71-2	US $58.00
CD-ROMs	1	→	**FPGA Course** ISBN 978-90-5381-225-9	US $29.00
	2	→	**ECD 4** ISBN 978-90-5381-159-7	US $38.00
	3	↗	**Ethernet Toolbox** ISBN 978-90-5381-214-3	US $39.00
	4	↗	**Home Automation** ISBN 978-90-5381-195-5	US $28.00
	5	↑	**USB Toolbox** ISBN 978-90-5381-212-9	US $39.00
Kits & Modules	1	↗	**Software Defined Radio** Art. # 070039-91	US $145.00
	2	↗	**Stand Alone OBD2 Analyser** Art. # 070038-93	US $119.95
	3	↘	**Remote control by Mobile Phone** Art. # 080324-71	US $94.50
	4	↑	**DCC Command Station** Art. # 070989-71	US $162.00
	5	↑	**Formula Flowcode Buggy** Art. # HP794	US $169.00

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

elektor SHOP

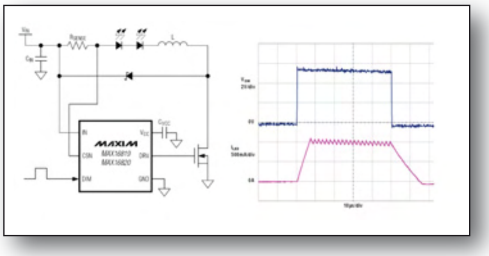
Elektor US
PO Box 876
Peterborough NH 03458-0876
USA
Toll-free: 888-924-9465
Phone: 603-924-9464
Fax: 603-924-9467
E-mail: custservus@elektor.com

NEXT MONTH IN ELEKTOR



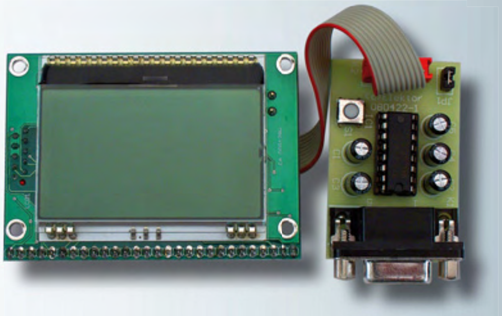
Test bench for model engines

High speed combustion engines for model airplanes, helicopters and so on require running in before they can be built into the model with confidence. Running in is a bothersome process if carried our manually. So, a test bench was designed that handles the process automatically. A servo is used to operate the throttle valve and the engine is automatically cycled through short periods of high and low speed, with constant rev counting and temperature monitoring. All measurement data can be sent to a PC for displaying and analysis.



SPICE simulation for LEDs

Hysteretic high-brightness LED drivers offer simple and low-cost implementations that require a minimum of external components. The complete circuit forms a self-oscillating system that is often not clearly understood by circuit designers. Consequently, they have trouble defining the right component values in the circuit. In this article we explain methods of LED driver design using simulation models in SPICE.



C Display

Our May 2008 issue featured a very straightforward graphical control and measurement device using the compact M16C Display Board and its built-in BASIC interpreter. Readers who prefer programming in C can also use the design if they build this mini circuit and download some free software from the web.

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

Elektor USA's April 2009 issue comes on sale on Thursday, March 16, 2009.
US mainland subsribers will receive their copy between March 11 - 14, 2009.

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

Elektor on the web

All Elektor UK/Europe 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.

elektor electronics worldwide

AT A SPECIAL INTRODUCTION RATE

Home | News | Magazine | Shop | Subscribe now | Forum UK site | Service

Search

Subject Area
-- Choose an option

Articles
- Informative articles
- Projects

Products
- Books
- CD-/DVD-Roms
- Kits & Modules
- E-blocks
- PCBs
- Controllers
- Magazines
- Elektor Credits
- Subscriptions

My Elektor
Log in
Email:
Email
Password:
Remain logged in

News
Annual Jahrgang Année Jaargang Anual
All articles from Volume 2008 on DVD-ROM
More

Revolutionary all-electric scalable powertrain
New low-cost 2.4 GHz transceiver
Miniature embedded device server runs Linux
DVB-TH terrestrial broadcast tuner boasts low power consumption
ORDERS TAKEN! Elektor volume 2008 on DVD
» News Overview

Subscribe now
Browse online
More

Formula Flowcode Buggy
USB-programmable robot vehicle (incl. CD-ROM)
More

Subscribers only
Order this bestseller now!
- $10.50 Discount
- Free P&P
Design your own EMBEDDED LINUX CONTROL CENTRE on a PC
More

i-TRIXX
Receive i-TRIXX in your mailbox for free each Wednesday!
i-TRIXX Electronics inside out! Free e-zine about hacking, tweaking and DIY electronics
More

Subscribe to our free E-weekly Newsletter!
Please enter your E-mail address:

Brand new
- DVD Elektor 2008
- Microcontroller Systems Engineering
- Transistor Curve Tracer (080068-91)

Bestsellers
- Buy 10 Elektor Credits (or multiples of 10)
- Microcontroller Systems Engineering
- LED Top with Special Effects

Top downloads LIVE DATA
- C Sharp (1)
- Programming FPGAs in C
- Elektor February 2009

PCB-PROTOTYPES
Calculate here...
PCB-POOL®

Also on the Elektor USA website:

- Electronics news and Elektor USA announcements
- Readers Forum *(linked to UK website)*
- PCB, software and e-magazine downloads
- Surveys and polls
- FAQ, Author Guidelines and Contact

Order Form 03-2009

Description		Price each	Qty.	Total	Order Code
DVD Elektor 2008	NEW	US $35.00			
Microcontroller Systems Engineering	NEW	US $52.00			
CapSense Buttons Kit (080875-91)	NEW	US $39.95			
CapSense Sliders Kit (080875-92)	NEW	US $39.95			
PIC Microcontrollers		US $52.00			
Design your own Embedded Linux Control Centre on a PC		US $42.00			
		Sub-total			
		Shipping & Handling US$ *(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*

EL03 *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-usa.com

elektor
electronics worldwide

Subscription Order Form 03-2009

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.

Name/Company

Street & No.

City/State/Zip *Country*

Tel *E-mail*

EL03 *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-usa.com

ORDERING INFORMATION

To order contact customer service:

Toll-free:	1-888-924-9465
On-line at	www.elektor-usa.com
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-usa.com

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.

NEW!
The International electronics magazine
elektor
Annual Jahrgang Année Jaargang Anual
All articles from Volume 2008 on DVD-ROM

Index of Advertisers

Avit Research, Showcase	www.avitresearch.co.uk	78
Beta Layout, Showcase	www.pcb-pool.com	17, 78
Bitscope	www.bitscope.com	2
ByVac, Showcase	www.byvac.com	78
C S Technology Ltd, Showcase	www.cstech.co.uk	78
Decibit Co. Ltd, Showcase	www.decibit.com	78
Designer Systems, Showcase	www.designersystems.co.uk	78
DLP Design	www.dlpdesign.com	17
EasyDAQ, Showcase	www.easydaq.biz	78
Easysync, Showcase	www.easysync.co.uk	78
Elnec, Showcase	www.elnec.com	78
Eurocircuits	www.eurocircuits.com	69
First Technology Transfer Ltd, Showcase	www.ftt.co.uk	78
FlexiPanel Ltd, Showcase	www.flexipanel.com	78
Front Panel Express	www.frontpanelexpress.com	37
Future Technology Devices, Showcase	www.ftdichip.com	78
General Circuits	www.pcbcart.com	37
Hameg	www.hameg.com	15
Labcenter	www.labcenter.com	88
Lcdmod Kit, Showcase	www.lcdmodkit.com	78
Linx Technologies	www.linxtechnologies.com	43
London Electronics College, Showcase	www.lec.org.uk	78
MikroElektronika	www.mikroe.com	10, 11
MQP Electronics, Showcase	www.mqp.com	78
Netronics, Showcase	www.cananalyser.co.uk	79
Parallax	www.parallax.com	43
Robot Electronics, Showcase	www.robot-electronics.co.uk	79
Robotiq, Showcase	www.robotiq.co.uk	79
RS Components	www.rswww.com/electronics	43
Saleae LLC	www.saleae.com	3
Saelig	www.saelig.com	21
ScanTool, Showcase	www.obd2cables.com, www.scantool.net	79
Showcase		78, 79
USB Instruments, Showcase	www.usb-instruments.com	79
Vesta Technology	www.vestatech.com	37
Virtins Technology, Showcase	www.virtins.com	79

Advertising space for the issue of 20 April 2009
may be reserved not later than 24 March 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.

LEAP INTO THE FUTURE...