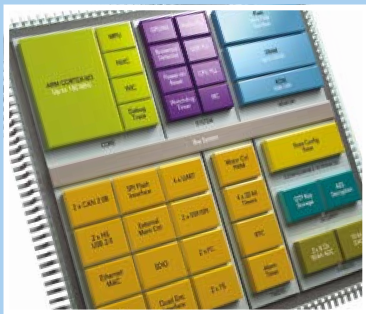
Edition 2/2015 | March & April 2015
www.elektor-magazine.com

FULLY RESTYLED!

elektor

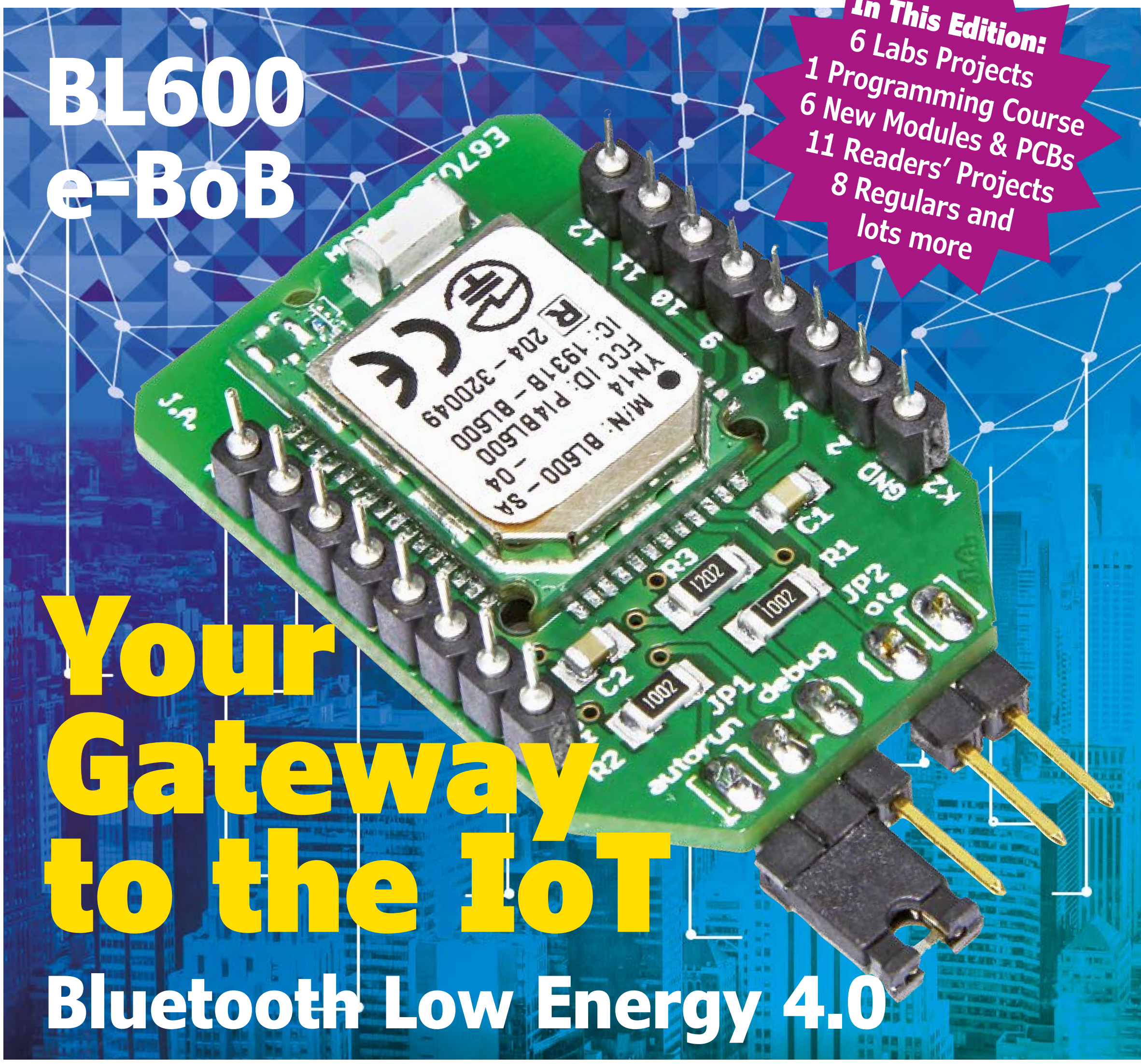
LEARN • DESIGN • SHARE



From 8 to 32 bits:
ARM Microcontrollers
for Beginners (2)

Pump your Guitar
Amp with the
OTA Overdrive

Wireless
Text Messaging
over ZigBee



BL600 e-BoB

In This Edition:
6 Labs Projects
1 Programming Course
6 New Modules & PCBs
11 Readers' Projects
8 Regulars and
lots more

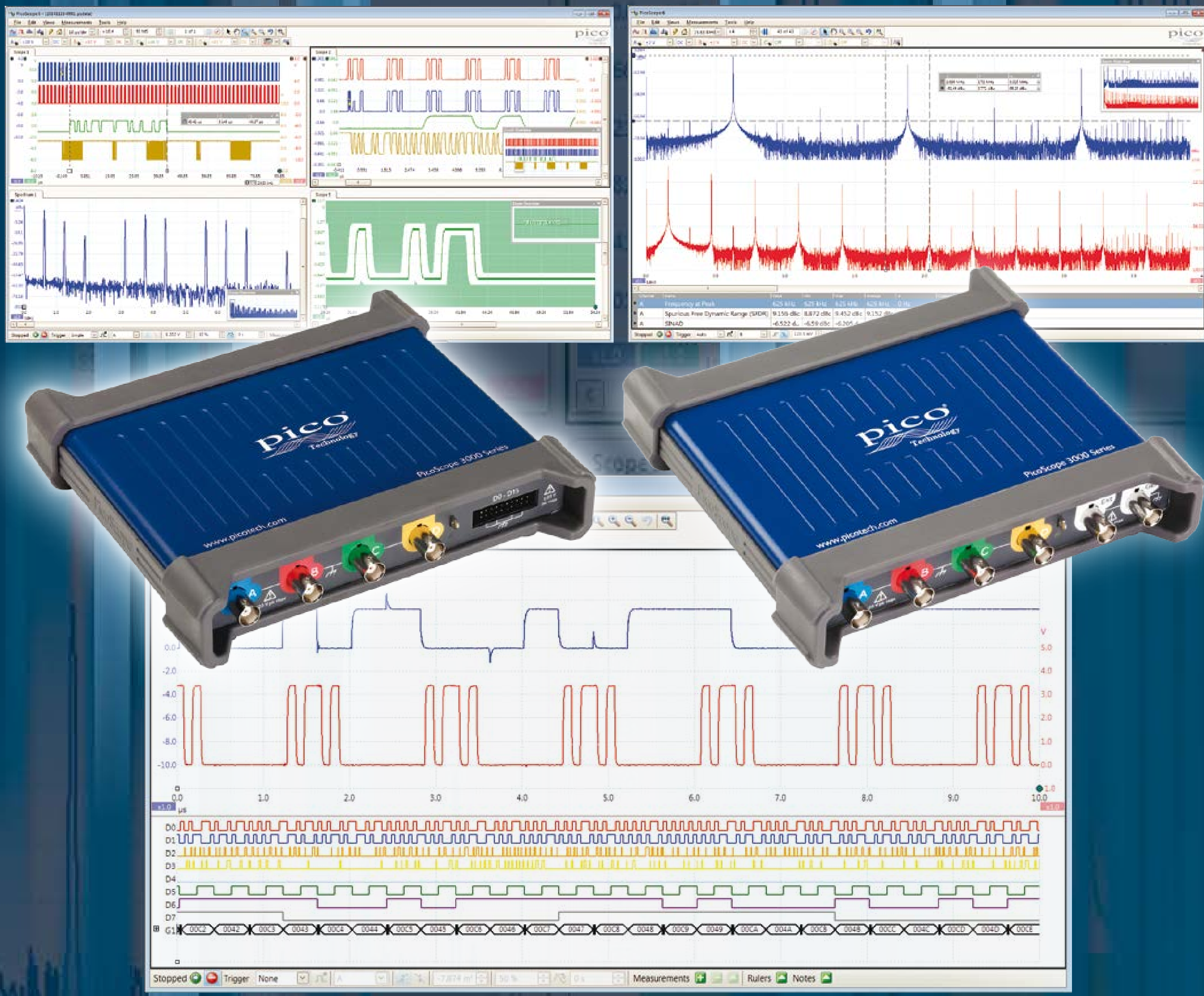
Your Gateway to the IoT

Bluetooth Low Energy 4.0

Welcome to LEARN / DESIGN / SHARE • **ARM Microcontrollers for Beginners** • **Q & A: Level Shifting** • **ARM CSIS Design Contest** • **Copper Pours** • Review: MultiSIM Blue • **BL600-eBoB** • **Guitar Overdrive** • **Radio Controlled Multi-Switch** • **Experimenter's Transistor Tester** • **Electronic Mole Repellent** • ZigTexter Network • **Air-Your-Own DCF77 Time** • **C Projects with one Click** • **Siphonic Rain Gauge** • Security Labels • **Ultrasonic Tape Measure** • **Gyrator-tuned Ferrite Antenna** • **350-V Step-Up Converter** • **HP400H VTVM Restoration** • Animated Electronics pico Technology

THE NEW PICOSCOPE 3000 SERIES

• Up to 200 MHz analog bandwidth • Deep buffer memory up to 512 MS
• MSO models with 16 digital channels • 2 or 4 analog channels
• 1 GS/s real-time sampling



Includes serial bus decoding and analysis (CAN, LIN, RS232, I2C, I2S, SPI, FlexRay), segmented memory, mask testing, spectrum analysis, and software development kit (SDK) all as standard, with free software updates and five years warranty.

www.picotech.com/PS408

Edition 2/2015
Volume 41, No. 459 & 460
March & April 2015

ISSN 1947-3753 (USA / Canada distribution)
ISSN 1757-0875 (UK / ROW distribution)
www.elektor.com,
www.elektor-magazine.com

Elektor Magazine, English edition
is published 6 times a year by

Elektor International Media
78 York Street
London W1H 1DP, UK
Phone: (+44) (0)20 7692 8344

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

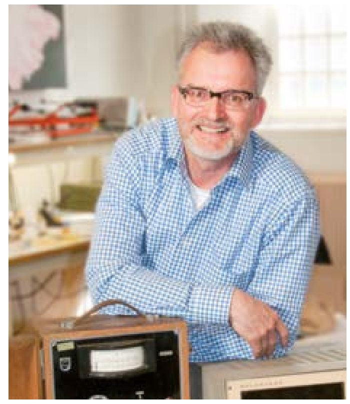
USA / Canada Memberships:
Elektor USA
P.O. Box 462228
Escondido, CA 92046
Phone: 800-269-6301
E-mail: elektor@pcspublink.com
Internet: www.elektor.com/member

UK / ROW Memberships:
Please use London address
E-mail: service@elektor.com
Internet: www.elektor.com/member

Advertising & Sponsoring:
Johan Dijk
Phone: +31 6 15894245
E-mail: johan.dijk@eimworld.com

www.elektor.com/advertising
Advertising rates and terms available on request.

Printed in the USA   Printed in the Netherlands



Get electronic in three easy steps: Learn, Design, Share

As of this edition of Elektor magazine, section coordinators are active on the pages to keep things in check. Where in the past it was difficult to distinguish a course from a hands-on project, or an advertisement from a review, by sub segmenting our publication into three clearly marked sections, Learn, Design, Share, we have given Elektor magazine not only a new appearance but also a better defined mission.
Hopefully you are not in the 'passives' drawer. The process of actively participating in electronics does not stop at sharing with others your top project, repair tip, or theoretical insight: by doing so you and the editorial team jointly help others learn something. Likewise, what seems to be the humble starting point of the process may well be the one with the highest personal impact considering that others care to tell and share things with you. In other words; you never learn from scratch.
In our new formula three Elektor staff members have the ominous task to preside over their section with sway and gusto. We have Jens Nickel on the **Learn** section, which you can expect to be extremely well organized and teaching brightly. **Design** is the domain of Clemens Valens the chief engineer at Elektor Labs and a stickler for reproducibility of our famed projects (note difference between Labs projects and Readers projects). **Share**, then, is turbo'd rather than presided over by Jaime González-Arintero. Jaime in essence wants to share everything until we tell him certain information is so valuable and classy it can pay our salaries. Each section president is fortunate to have a few faithful contributors who continue to fill subsections you have come to appreciate, like tech gossip from the Labs, good old Retronics and Hexadoku. Finally, an editor like myself ensures everything gets presented in a not too riotous way. Reigning supreme but usually off screen is the publisher who has a few words to say in his letter accompanying this 2/2015 edition.
Do not strictly follow the order Learn, Design, Share — just be aware of the virtual doors in between. They're bidirectional and glazed. Enter at any level you want, first do what you are good at, and determine the order you prefer. That's how we worked for you here at Elektor magazine these past 40 years.
Enjoy reading this edition

Jan Buiting, Editor-in-Chief

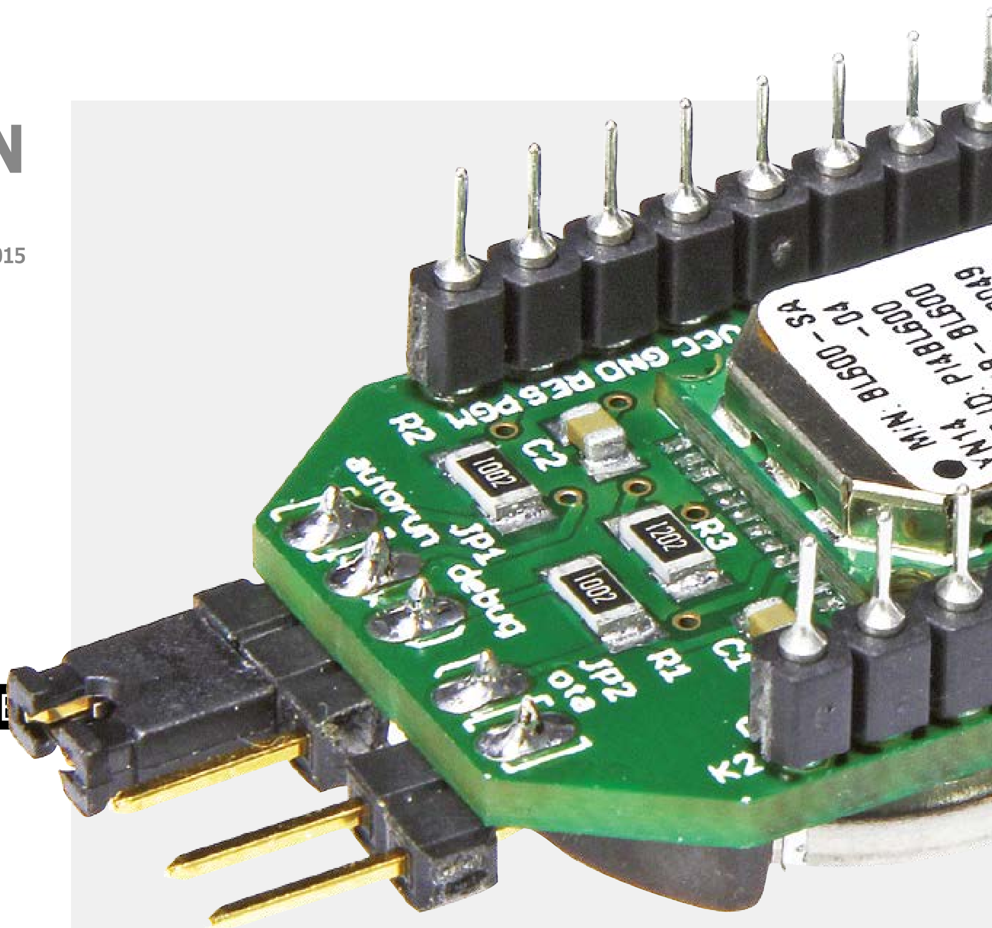
The Circuit

Editor-in-Chief:	Jan Buiting
Publisher:	Don Akkermans
Membership Manager:	Raoul Morreau
Client Executive:	Cindy Tijssen
International Editorial Staff:	Harry Baggen, Jaime González-Arintero, Denis Meyer, Jens Nickel
Laboratory Staff:	Thijs Beckers, Ton Giesberts, Luc Lemmens, Clemens Valens, Jan Visser
Graphic Design & Prepress:	Giel Dols
Online Manager:	Daniëlle Mertens

THIS EDITION

Volume 41 – Edition 2/2015

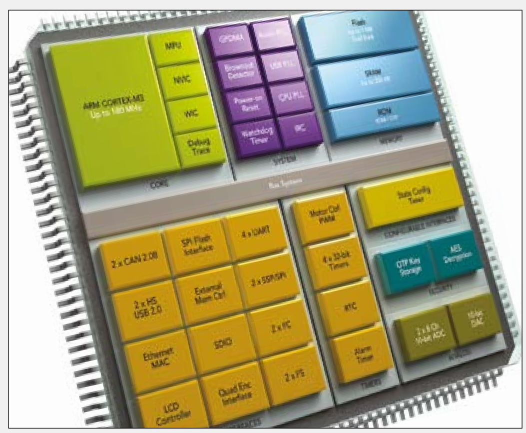
No. 459 & 460 March & April 2015



LEARN DESIGN SHARE

From 8 to 32 bits:
ARM Microcontrollers for Beginners
GPIOs and the U(S)ART

10



LEARN DESIGN SHARE

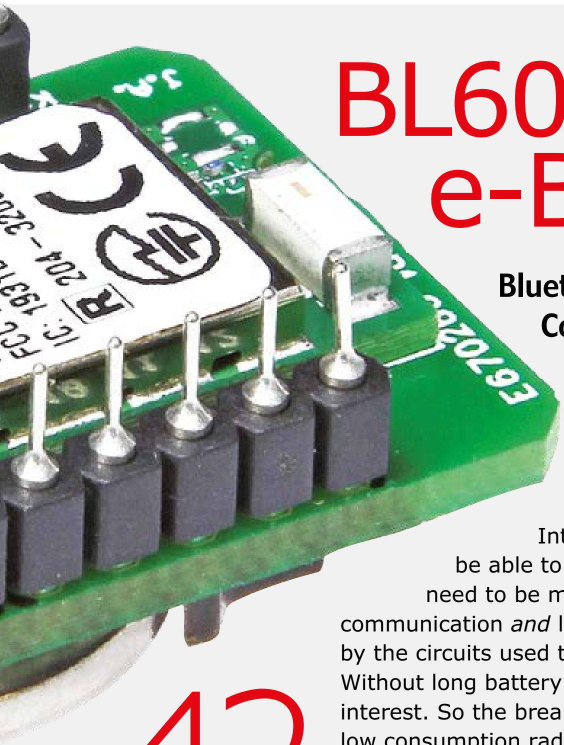
Gitaar Overdrive

With genuine 'germanium-sound'

Unusually in this day and age of DSPs and microcontrollers, the distortion effects generated by this guitar sound effects unit are generated with the aid of paired silicon and germanium diodes.

48

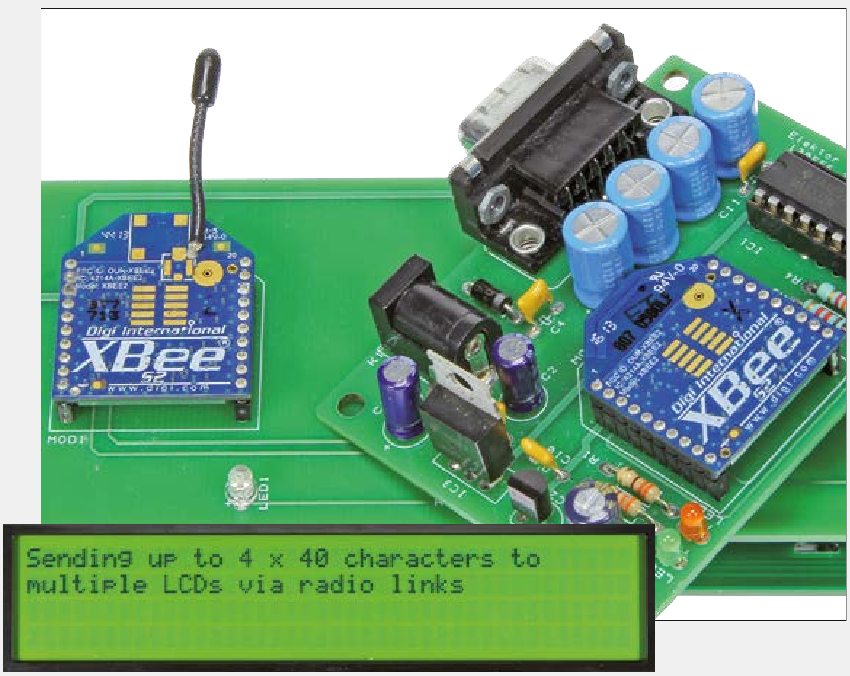
elektor magazine



BL600 - e-BOB

Bluetooth Low Energy Communication Module Part 1

For the network of connected objects (or Internet of things – IoT) to be able to thrive, certain conditions need to be met — in particular, wireless communication *and* low power consumption by the circuits used to connect these objects. Without long battery life, we'll soon lose interest. So the breakout board for the ultra-low consumption radio communication module presented in this article is going to be an ideal accessory for exploring IoT.



Elektor ZigTexter Network

Short-messaging over Zigbee

Here's a flexible system to broadcast text messages wirelessly to family members, neighbors, clients, visitors, colleagues, lazy staff, head cooks and bottle washers — all via Zigbee, with a handy message scheduling function built in.

NEXT EDITION

RS232 Data Logger

This nifty little circuit eavesdrops on an RS-232 link and sends data electrically isolated through a USB port to a terminal emulator running on your PC.

Inductive AM Transmitter with Arduino

Here we slap together an Arduino Uno, an Elektor extension shield and a few components to make a small AM transmitter for local experimenting.

Basement Ventilation System

An intelligent circuit based on a Platino, combating moisture problems in a cellar using two sensors, a ventilator and a basement window control.

Edition 3/2015 (May & June) is published on April 15, 2015.
Delivery of printed copies to Gold members subject to transport.
Content and article titles subject to change.

Elektor Circuits &

Elektor breaks the constraints of a magazine. It's a community of active e-engineers — from novices to professionals — eager to learn, make, design, and share surprising electronics.

56

Countries

233492

Enthusiastic Members

10

Experts &

Elektor Labs

Learn, Design & Share

The techno creative center of Elektor that's steeped in hard core electronics all the way from scribble to PCB, component and kit. Wide open and accessible through its own website, Labs is where projects large, small, analog, digital, new and old skool are sketched, built, discussed, debugged and fine-tuned for replication and use by you.

www.elektor-labs.com

Elektor Community

Become a member, Green or Gold

Membership of the Elektor Community is the surest way to enjoy classic electronics and embedded technologies side by side, ranging from beginner to pro. With direct access to Elektor Labs, forums, discounts, weekly newsletters, biweekly online projects, article archives, search engines, and back articles Green and Gold Members have permanent priority seating. Go GREEN if you want the magazine front to back delivered online only, or GOLD for the sumptuous package including printed copies.

www.elektor.com/memberships

Elektor Web Store

Fill your shopping cart

Elektor has confidence in the products and services generated by Labs, Magazine, and selected business partners. That's why a brightly illuminated online retail store is open 24/7/365, with ordering and payment facilities for clients all over the world. An Aladdin's Cave of electronic parts and gizmos.

www.elektor.com

Elektor Magazine

Close to 1024 pages of surprising electronics a year

If you prefer to absorb electronics over being absorbed, stick to reading Elektor's flagship product DMA'ed to you by their international editorial team. Whether arriving online or on paper every magazine edition is packed with electronics all-sorts for you to enjoy and explore in your own time. Free! Sign up!

www.elektor-magazine.com

Connections Guide

21	463	209651	14:32 FEBRUARY 10 2015
Authors	Publications	Montly Visitors	Print Time

Elektor Post

The e-inspiration weekly

Never monostable and with trigger signals all over the contents, Elektor's dot-Post weekly newsletter has the ability to ring in the weekend with gossip, techtalk, stray bits, previews and news flashes. And a project every other week.

www.elektor.com/newsletter

Elektor Academy

Ride the learning curve

Webinars, seminars, courses, presentations, workshops, lectures, in-company trainings, DVDs, and demos are just a few of the methods Elektor is using to spread the word about electronics both at hobby and professional levels.

www.elektor-academy.com

Elektor TV

We're tubed too

No film set, suits, or Action! but you can rely on a camera rolling whenever things start humming, booting, displaying or smoking at Labs, or indeed any place or event our presenters find video compatible. Check out elektor.tv.

www.youtube.com/user/ElektorIM

Elektor Books & DVDs

E-Information Powerpacks

It's hard to find a field of electronics not covered in depth and with authority by the products in our book and DVD portfolio. From reference work to programming course, 8-bit to ARM, Antenna to Zener diode; it's all there.

www.elektor.com

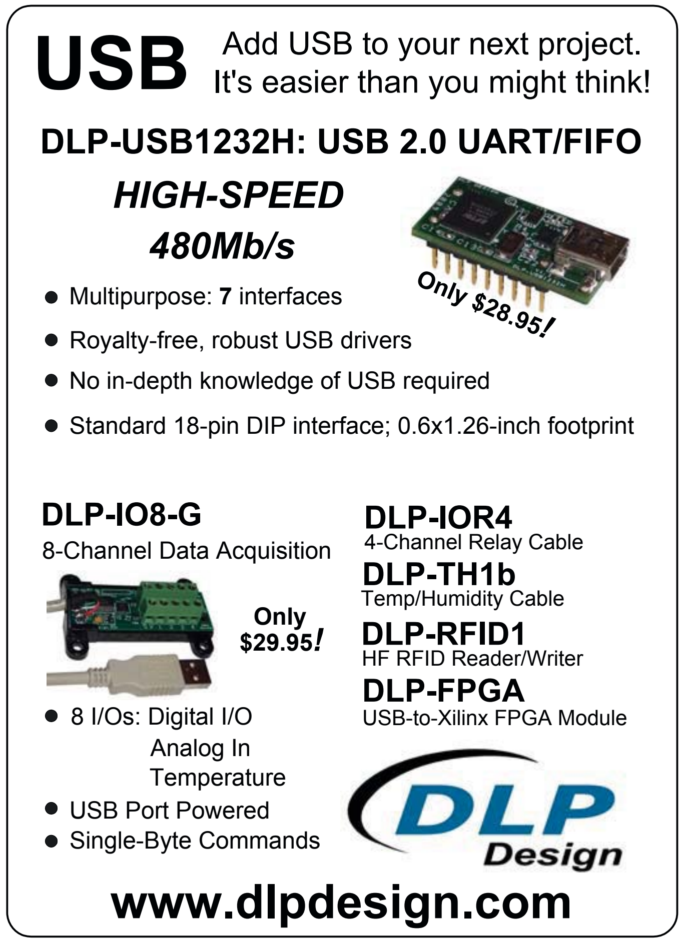
Elektor PCB Service

Boards at your service

Forget the chemicals, get your electronic project to work as expected by ordering a ready-manufactured circuit board. Fast turnaround, pure quality, worldwide shipping.

www.elektorpcbservice.com


The Convenient All-in-One Solution
for Custom-Designed Front Panels & Enclosures
FREE Software
ONLY $90.24 with custom logo engraving
You design it
to your specifications using our FREE CAD software, Front Panel Designer
We machine it
and ship to you a professionally finished product, no minimum quantity required
• Cost effective prototypes and production runs with no setup charges
• Powder-coated and anodized finishes in various colors
• Select from aluminum, acrylic or provide your own material
• Standard lead time in 5 days or express manufacturing in 3 or 1 days
FRONT PANEL EXPRESS
FrontPanelExpress.com
1(800)FPE-9060


USB
Add USB to your next project.
It's easier than you might think!
DLP-USB1232H: USB 2.0 UART/FIFO
HIGH-SPEED
480Mb/s
Only $28.95!
● Multipurpose: 7 interfaces
● Royalty-free, robust USB drivers
● No in-depth knowledge of USB required
● Standard 18-pin DIP interface; 0.6x1.26-inch footprint
DLP-IO8-G
8-Channel Data Acquisition
Only $29.95!
● 8 I/Os: Digital I/O
Analog In
Temperature
● USB Port Powered
● Single-Byte Commands
DLP-IOR4
4-Channel Relay Cable
DLP-TH1b
Temp/Humidity Cable
DLP-RFID1
HF RFID Reader/Writer
DLP-FPGA
USB-to-Xilinx FPGA Module
DLP Design
www.dlpdesign.com


JOIN THE EVOLUTION.
Learn more
1905
1945
2005
Today
Get "mobile smart" in 3 easy steps:
1 Get your AIR for Wiced Smart dev kit at your distributor of choice. (See our website for a current list.)
2 Develop your wireless link and basic app using our exclusive Atmosphere development tool.
3 With our AIR for Wiced Smart module on board, proceed in record time to a prototype and final, mobile-app development!
Anaren ATMOSPHERE
AIR Anaren Integrated Radio
for WICED
Evolve to app-based control with AIR for Wiced Smart!
If you're ready to evolve from fixed control panels populated with dials, buttons, keypads, and LCD displays to mobile-app based control of your embedded product – check out Anaren's AIR for Wiced Smart module, featuring Broadcom's Wiced Smart Bluetooth® chip (BCM20737). Not only does our small-footprint, SMT, and pre-certified all-in-one module save you the time, effort, and trouble of designing your own radio... It's supported by our industry-exclusive Atmosphere development ecosystem that lets you develop your basic embedded code and app code in one, easy-to-use development tool – for a far speedier product development cycle and time-to-market.
Follow the steps at left to join the evolution, right now!
www.anaren.com/AIRforWiced
800-411-6596
In Europe: 44-2392-232392
Anaren®
What'll we think of next?™
Bluetooth

By Jens Nickel

You live and learn …

It's good to be here: As an editor I earned my spurs at a publication in Germany on prefabricated houses, their target demographic was young families. When I started out I must admit I didn't know a lot about building I wouldn't have been able to tell you the difference between a joist and a rafter. The learning curve was steep and fairly soon I was on first name terms with representatives from many of the leading companies. Fate played its part and next I was working for a professional magazine specializing in solar PV installations. Here I was able to delve a bit further into the technical aspects of the technology; most of our readers were professionals. Soon I was au fait with the most recent EU thinking on greenhouse gas emissions, PV wafer fabrication and power inverters.

Now I've been at Elektor for 10 years already — and what can I say? Sometimes at an electronics exhibition I feel like a little boy again, gawping wide-eyed at all the flashy new technology on show. Sometimes I just need to scribble down some key words so that later, back at my hotel I can rely on Google to fill the gaps in my knowledge. Now, whenever I get the time, I enjoy getting to grips with the latest technology.

I'm sure it's the same for you. The term 'learning by doing' is especially true in the field of electronics. Tinkering and experimenting is particularly rewarding, it not only adds to your knowledge but it's also usually fun. That is in essence what Elektor has always been about right from the start, with our articles, books, webinars and seminars, our aim is to share expertise in a positive way.

The new 'Learn' section of our (international) magazine aims to bundle everything that helps you to get a better grasp of electronics as a skill. Occasionally we plan to venture out into sub disciplines hitherto unknown to you. There are two new sub-headings that will be of particular interest, not only from what you will learn but also what you can teach us!

Tips & Tricks

Got a neat solution for a tricky problem? Using components or tools in ways they were never intended to be used? Think your idea to solve a problem is better than a solution you've just read (maybe in Elektor)? Have you discovered a work-around that you want to share with us and other makers? Don't hang about; write to my fellow editor Jan now at editor@elektor.com and he will forward.

Q & A

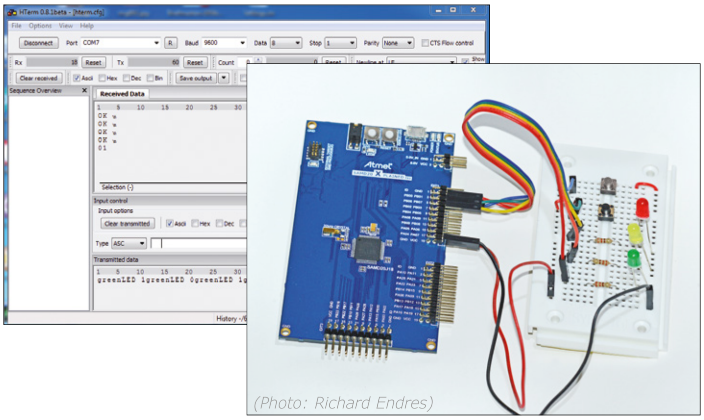
What you've always wanted to know about… Send us your questions and we will get experts to give their advice. We want those tricky problems that confront engineers day to day. In this edition we take a look at level shifting; we welcome your suggestions for the next Q & A topic.

On my desktop…

… For the last couple of weeks I've been looking into the SAM-D20-Board that we are using in our ARM course. After I'd read through the first installment that Viacheslav (the article's author) sent me I was really eager to roll up my sleeves and get down to controlling the World! OK, let's just start with the LED on the board first… It was clear I could use our EFL Lib; it already has some modules which enable remote control via a UART and also provides some other useful functions. I spent some time studying the Atmel Software Frameworks (ASF) and eventually decided, so as not to confuse the calls, that it was easier to set up the functions needed for the EFL Controller file (to address individual pins) from the ASF. Once this was done it all went quite quickly: using the I^2C library routines we covered in the November 2014 edition, a small piece of perfboard and all the necessary connectors I was able to build an EEC/Gnublin port and control it using I^2C commands. The photo shows the familiar Gnublin Relay module hooked up to the SAM D2 board (and also shows that I needed to do a little bit of soldering ;-)). Meanwhile I have integrated the UART functions, which are covered in this installment, into the controller file. Using the existing EFL library modules I can now control the relays directly from the PC J. In the next installment we go a bit further and look into the I^2C hardware.

From 8 to 32 bits: ARM Microcontrollers for Beginners (2)

GPIOs and the U(S)ART



(Photo: Richard Endres)

By Viacheslav Gromov (Germany)

In the first part of this course we went into plenty of detail about how to install the development environment, leaving relatively little time for experimenting. Things are about to change: we will now look at the GPIO pins and then the serial interfaces. We will also control a small circuit built on a breadboard from the PC via the ARM board.

Although we took a quick look at the GPIO pins in the first installment of the series, we did not explore all the available possibilities and commands. Here, however, we will go deeper into the world of digital inputs and outputs.
The second project in this installment builds on the first and makes use of a very important peripheral interface, the USART.

The Atmel Software Framework

The Atmel Software Framework (ASF) is a handy tool that forms part of Atmel Studio, as we saw in the first part. The overall structure of the ASF is shown in **Figure 1**. As you can see, the ASF provides support at all software layers, from the register level to application code. Switching between members of the microcontroller family or between boards is made simple by the fact that only the relevant layers need to be adapted. **Figure 2** shows the structure of the ASF directory that appears in each newly-created ASF-based project. The 'common' directory includes files that are independent of the microcontroller and board, while the files in 'sam0' are specific to the SAM Cortex M0 processor. The directory 'common2' contains files supporting a group (or family) of microcontrollers. Under 'config' are stored important configuration files, for example regarding the source of the processor clock [1][2].
To get things under way, first create a new 'ASF Board Project' (as described in the last installment) called 'The second program'. Then click in the upper toolbar on 'ASF Wizard' and select the newly-created project at the top of the window that opens (**Figure 3**). On the left you can see all the ASF libraries available for the microcontroller. For this project the only

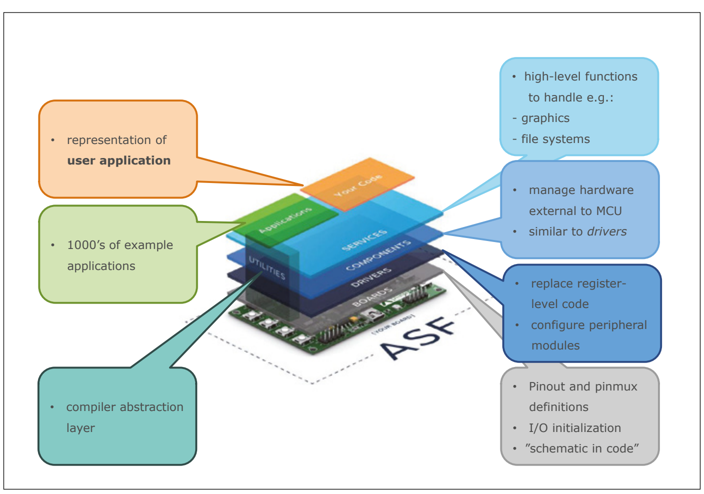


Figure 1. The individual elements of the Atmel Software Framework (all screenshots and block diagrams courtesy Atmel).

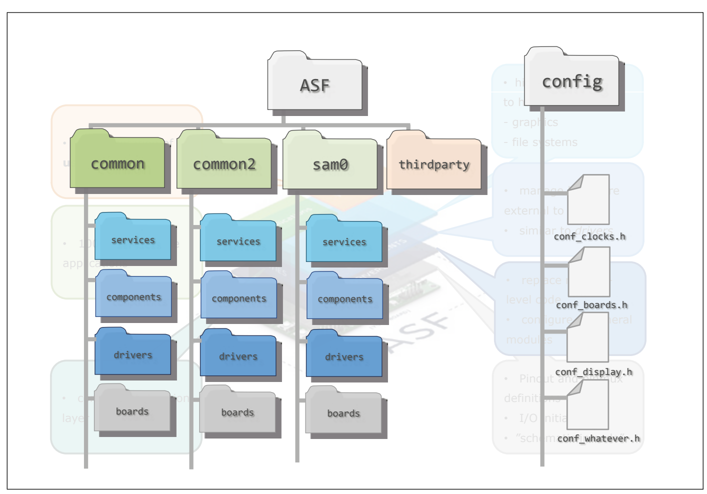


Figure 2. The arrangement of paths under the ASF directory.

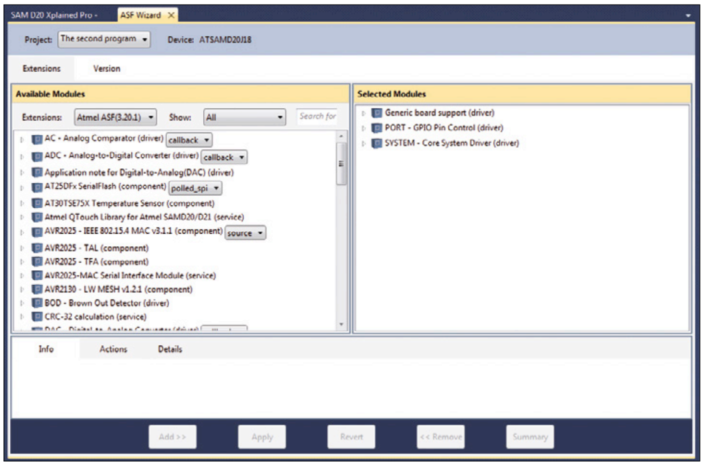


Figure 3. The ASF Wizard includes many well-thought-out libraries, most accompanied by example programs.

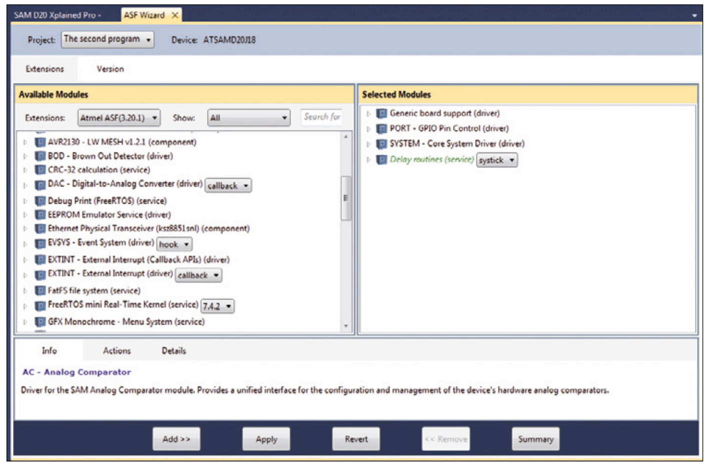


Figure 4. The delay library has been selected, but not yet added to the project directory.

additional library we need is the one that includes the 'delay' routines (select the 'systick' variant). Use 'Add>>' to add this library to the list of modules for the CPU already installed, which cover the clock source, the I/O ports and so on: see **Figure 4**. Clicking on the library brings up a brief description at the bottom of the window: the module provides delay functions using the 24-bit SysTick timer. Click on 'Apply' to cause the library to be included in the main project directory. Finally we arrive at a window like the one shown in **Figure 5**, which summarizes the situation. Click 'OK' and the library is now linked in to the project: the corresponding files will be found under 'src/ASF/common2/services/delay'.

The majority of the ASF libraries are well documented. If, at the top right, you open the ASF Explorer (**Figure 6**) instead of the Solution Explorer, you can click on each of the linked-in ASF libraries to open a description which includes a link to the API documentation and often also a handy 'Quick Start Guide'.

The ports

First a bit of theory. **Figure 7** shows an outline of the structure of the port peripheral module, which is responsible for handling the GPIO pins. Half-way down on the right are the individual port pins themselves. Each port can comprise up to 32 pins, corresponding to the maximum width of a bus access by the ARM Cortex-M0+ core. The pins are connected to a multiplexer, which, depending on the configuration of the control and status registers (above left), connects them in turn to the various peripheral modules. The control and status registers also allow the level on a pin to be set directly, hence using the pin as a digital output. There are five main registers involved in this process, which, as in eight-bit microcontrollers, determine the data direction (input or output), enable pull-up or pull-down resistors, enable the input function of a pin and read its input level or set its output level. A bit in a register corresponding to a given pin can be accessed individually, or entire registers can be accessed at once. In general, detailed knowledge of how the registers work is not needed, but it is useful to understand the basic range of functions available.

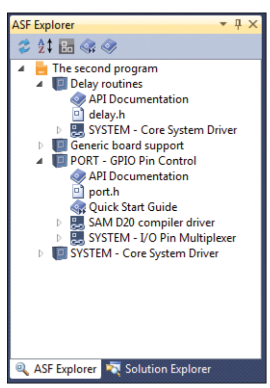
It is possible to enable a pull-up or pull-down resistor on a pin without configuring its input or output function. The result is that the pin is pulled to a fixed level but has a relatively high impedance. An unusual feature is that a pin can be configured as an output and an input simultaneously (called 'output with readback'). All these functions are enabled through particular combinations of values in the five registers.

The CPU can access the ports very quickly using its single-cycle I/O bus. More information can be found on page 287 of the data sheet [3].

Controlling the GPIOs

On the SAM D20 board itself there is only one LED and one button available, but with a small additional circuit (**Figure 8**) we can easily expand the possibilities. An easy way to build the simple circuits in this course is to use an ordinary solderless breadboard, connected to the extension headers on the Atmel board using the plug leads widely available for use with the Raspberry Pi (**Figure 9**).

The code in **Listing 1** can be downloaded from the web page for this project [7]. It begins by including the file 'asf.h', which defines symbolic names for the pins. Before the main routine there is a function which configures the pins (see the Application Note at [4]). It starts by creating a structure (keyword `struct`) of type `port_config` called `config_port_pin`. The structure includes various elements that represent the individual GPIO settings. Each element in a structure can be accessed using the 'dot' ('.') operator; and if we want to pass a complete



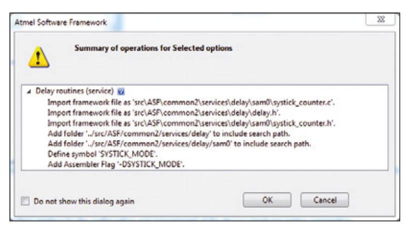


Figure 5. This view shows where the new files are stored within the project directory.

Figure 6. The ASF Explorer also shows where documentation is available for a given library.

Listing 1. Simple control of the GPIOs.

```
#include <asf.h>

#define LEDG  PIN_PB00                //define the pins
#define SW1   PIN_PB06
#define SW2   PIN_PB07

void configure_port_pins(void);

void configure_port_pins(void)        //setup of the configuration of the pins
{
  struct port_config config_port_pin;
  port_get_config_defaults(&config_port_pin);

  config_port_pin.direction = PORT_PIN_DIR_OUTPUT;
  port_pin_set_config(LEDG, &config_port_pin);
//PB00 (green LED) as output
  port_group_set_config(&PORTB, 6, &config_port_pin);
//PB01 (yellow LED) and PB02 (red LED) as output

  config_port_pin.direction = PORT_PIN_DIR_INPUT;
  config_port_pin.input_pull = PORT_PIN_PULL_DOWN;
  port_pin_set_config(SW1, &config_port_pin);
//PB06 (SW1) as input with pull down

  config_port_pin.input_pull = PORT_PIN_PULL_UP;
  port_pin_set_config(SW2, &config_port_pin);
//PB07 (SW2) as input with pull-up
}

int main (void)
{
  system_init();                       //initialization
  delay_init();
  configure_port_pins();               //configuration of the pins

  while (1) {
    if (port_pin_get_input_level(SW1) == 1)
//if PB06 (SW1) is high, green LED is blinking
    {
      port_pin_toggle_output_level(LEDG);
      delay_s(0.5);
      port_pin_toggle_output_level(LEDG);
      delay_s(0.5);
    }
    if (port_pin_get_input_level(SW2) == 0)
//if PB07 (SW2) is low, yellow and red LEDs are blinking
    {
      port_group_set_output_level(&PORTB, 6, 6);
      delay_ms(500);
      port_group_set_output_level(&PORTB, 6, 0);
      delay_ms(500);
    }
  }
}
```

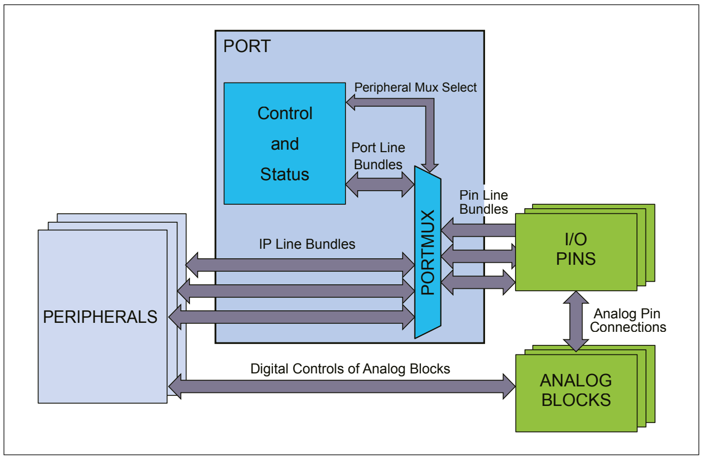


Figure 7. The port peripheral module with its available connections and functions.

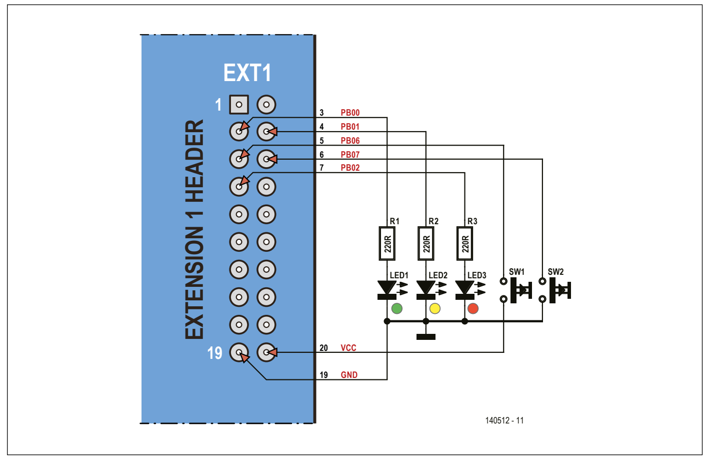


Figure 8. A couple of buttons and LEDs are connected to the first extension connector.

group of settings to a function efficiently, we need only send a pointer to the structure (that is, its start address in memory) as a parameter. It is important to have a good grasp of how this mechanism works, as it is widely used in microcontroller libraries (not just in the ASF). Much more information on the use of structures and pointers in the C programming language can be found on the Internet.

Once the structure has been declared, it can be populated with default values (corresponding to the GPIOs being inputs with pull-ups enabled) using the command `port_get_config_defaults(&config_port_pin)`. This two-step process is typical of the functions in the ASF libraries. The next step is to make the configuration changes required by our particular application. First we set the variable `direction` in the structure to the constant value `PORT_PIN_DIR_OUTPUT`. Next, one of the pins is configured as an output: `port_pin_set_config(LEDG, &config_port_pin)`. As you can see, the function requires only the pin name (here as a symbolic constant) and a pointer to the configuration structure to be passed to it. To apply the configuration that has been set up in the structure to two GPIO pins we use the command `port_group_set_config(&PORTB, 6, &config_port_pin)`. The parameters here are: a pointer to the port; a mask representing the GPIO pins to be affected; and finally a pointer to the configuration structure. Since in our example the yellow and the red LEDs are connected to pins PB01 and PB02, the mask is 6, which when written in binary has one-bits in the second and third positions from the least-significant end: the least-significant bit corresponds to pin PB00.

Next we set up two GPIO pins as inputs to read the two buttons. We will configure one input to have its internal pull-up resistor enabled, the other its pull-down resistor. We write the value `PORT_PIN_DIR_INPUT` into the structure element `direction`, and the value `PORT_PIN_PULL_DOWN` into the element `input_pull`. We then apply this configuration to GPIO pin PB06 (to which SW1 is connected), which activates the pull-down resistor. We finally modify the contents of the structure for a 'pull-up' configuration and apply it to pin PB07, to which SW2 is connected.

The main routine begins with the familiar `system_init()` command and then the command `delay_init()`, which initializes the SysTick timer for use by the delay function. The command `configure_port_pins()` calls the configuration function described above. The code then drops into a simple infinite loop, consisting of two if-blocks in which the `port_pin_get_input_level()` command is used to check the level on the two GPIO pins that are connected to the pushbuttons. In the first block an output GPIO pin connected to an LED is toggled using the command `port_pin_set_output_level();` after a 0.5 s delay it is toggled again using the same command, followed by a further 0.5 s delay. The second block is similar, but uses the command `port_group_set_output_level()` to set the levels on two output pins simultaneously. This function requires not only a mask representing the set of port pins to be affected, but also a mask representing the set of desired values.

The code can be debugged using the 'Start Debugging and Break' icon in the toolbar (or ALT-F5). Once the code has been compiled without errors the debug window will open (**Figure 10**). The program can now be executed and tested one command at a time using 'Step Over' (or F10), which is a good way to help track down possible bugs. On the right is a display of the contents of the most important registers, and below a dump of the contents of various memory areas. To the left of the dump there would appear a list of local variables, but there are none in our example code. The icon 'Start without Debugging' can be used to run the program as normal:

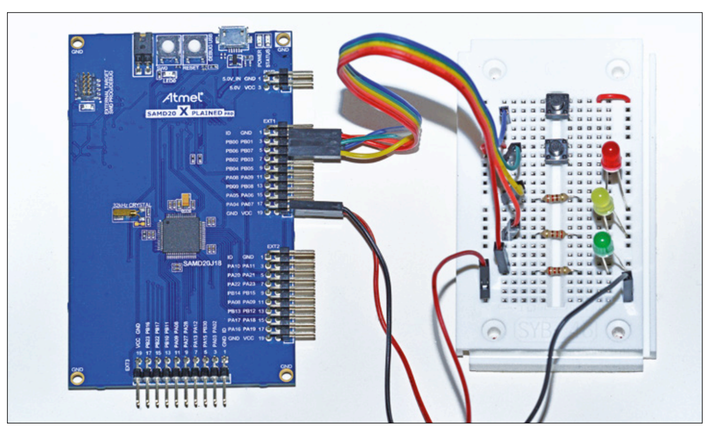


Figure 9. This method of prototyping makes modifications and extensions straightforward (photo: Richard Endres).

try holding down both buttons, and you should see the LEDs flash alternately.

The SERCOM module

The SERCOM modules in the SAM D20 are versatile, and can be configured to implement a wide range of serial bus protocols. The 'pads' (connections for clock and data signals) can be routed to any desired GPIO pins. The module also offers a sleep mode. The block diagram (see **Figure 11**) shows at the top the main register groups for controlling the configuration

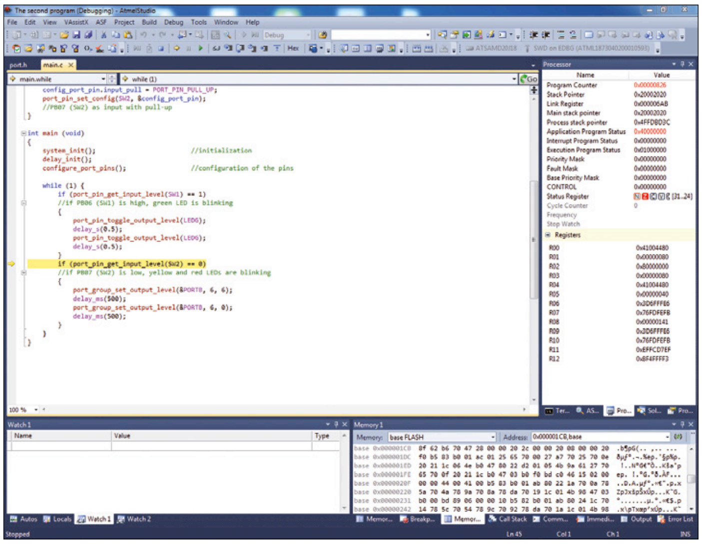


Figure 10. The debug window.

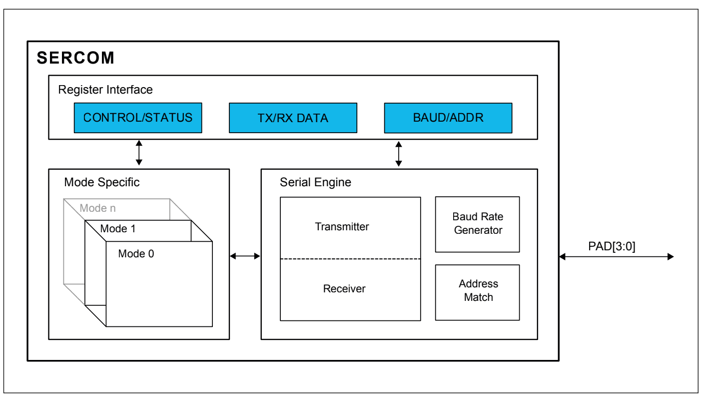


Figure 11. The SERCOM peripheral is very versatile.

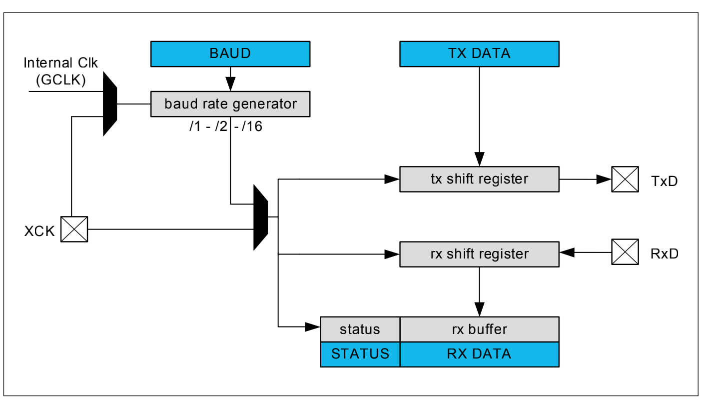


Figure 12. The SERCOM peripheral configured as a USART.

of the SERCOM module, for setting baud rates and addresses, and of course for sending and receiving data. The various serial modes (I²C, SPI and UART) are indicated below the registers as well as the block that actually implements the reception, transmission and comparison of addresses and data.

A SERCOM module configured as a USART is very flexible: for example it can handle from 5 to 9 data bits per frame along with one or two stop bits, and data can be sent and received LSB first or MSB first. In synchronous mode a third pin is used for the clock signal XCK. A range of internal and external clock sources is available, and they can be divided by ratios down to 1:16. The USART can trigger an interrupt under various conditions, such as when a data byte is received. There is a digital filter on the input signal. The simplified diagram in **Figure 12** shows the clock source and baud rate divider on the left and the transmit, receive and status registers on the right. A more complete description can be found in [3], from page 336 onwards.

Testing the USART

For the second example program of this article we will demonstrate the USART in asynchronous (UART) mode. In this experiment we will only need the existing USB cable that is already connected to the board and a terminal emulator program on the PC, as the EDBG device on the board is connected to one of the SERCOM modules on the microcontroller and does the conversion between serial and USB. We have also already installed the necessary Windows driver on the PC in the previous article in this series.

Create, as before, a new project called 'The first project with UART' and add the SERCOM UART library to the project using the ASF Wizard as described above (see **Figure 13**). Make sure you choose the 'callback' version (rather than the 'polled' version) as this avoids the need to get involved with UART interrupts. Instead, we simply tell the ASF library which of our functions are to be called when a certain number of characters has been received by the UART, or when a given string has been sent.

The code for this first UART project can be found on the web pages accompanying this article [7].

At the beginning of the program we declare a structure of type `usart_module` called `usart_instance`, which we will use later when calling the UART functions.

Next we declare our callback function, called `usart_read_callback()`, which is called whenever ten characters are received in `string1`. Within this function the `strcmp()` function is now used to compare this string against two possible ASCII commands. If a match is found, the green LED is turned on or off as appropriate. Note that the `string.h` header file should be included before the `strcmp()` function is used.

The program sends the string 'OK' followed by a space and a newline character as acknowledgement, using the command `usart_write_buffer_job(&usart_instance, (uint8_t *) string5, 4)`. The three parameters here are a pointer to the USART instance structure, the string to be sent and the total character count.

If you enter the string 'SW1 status' in the terminal emulator, the program will poll the state of button SW1 and then output the result using the function `usart_write_job(&usart_instance, XX)`, where XX is 48 or 49. In addition to the pointer to the USART instance structure this function also needs a single character, given as an ASCII value, which it will output. Here 49 is the ASCII code for the digit '1', 48 the code for the digit '0'.

The next function is our callback routine `usart_write_callback()`, which is called whenever a transmission has been completed. We have arranged for it to toggle the states of the yellow and red LEDs each time it is called.

In the configuration function `configure_usart()` we first declare a structure of type `usart_config` called `config_usart` and populate it with its default settings. We then set the baud rate and the pins that are to be used by the USART: in our case these are the pins that are connected to the EDBG device. Finally we initialize the SERCOM in question as a USART using a pointer to the structure. From this point on we can access the USART via the instance structure `usart_instance`. The final function call registers and enables the callback functions.

In the main routine the configuration and initialization functions described above are called and then global interrupts are enabled. Within the while-loop is the single command `usart_read_buffer_job(&usart_instance, (uint8_t *)string1, sizeof(string1))`, which in the background reads exactly ten characters (the size of `string1`) and then calls the callback function.

As can be seen, with the help of the ASF libraries it is possible to send or receive individual characters or even complete strings in the background without blocking the execution of the main program. This is the advantage of using the 'callback' version of the library over the 'polled' variant: the latter must always wait in a while-loop until each command is completed. If you wish to use the callback-based library without having to specify in advance the exact number of characters that are expected (which is not always known), a slightly more complex approach is needed. Arrange for the callback function to be called on receipt of each character. In that routine store the character in a buffer and initiate another receive task to wait for the next character. The buffer can be checked on each occasion to see whether a complete string has been received.

Note that it is important to configure the terminal emulator program with the correct baud rate (9600 baud in this case) and to use the correct COM port. It is also necessary to ensure that when a string is sent no spurious CR or LF character is added. The EDBG device requires that the DTR signal of the (virtual) serial port is activated, and this must also be configured in the terminal. The screenshot in **Figure 14** shows the required settings in HTerm.

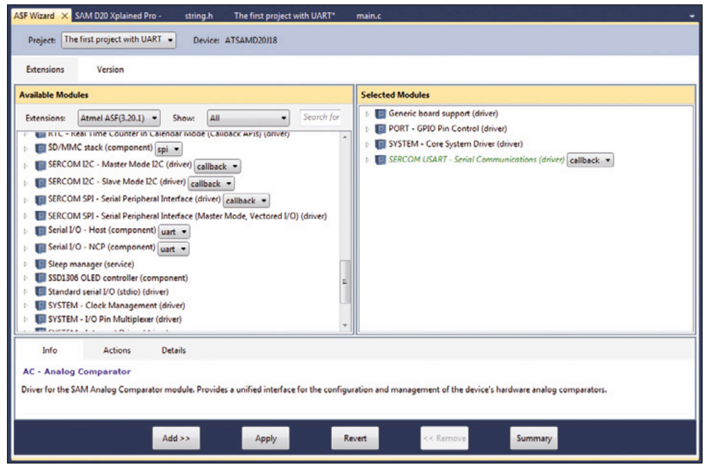


Figure 13. The USART library is selected in the ASF Wizard and can now be added to the project directory by clicking on 'Apply'.

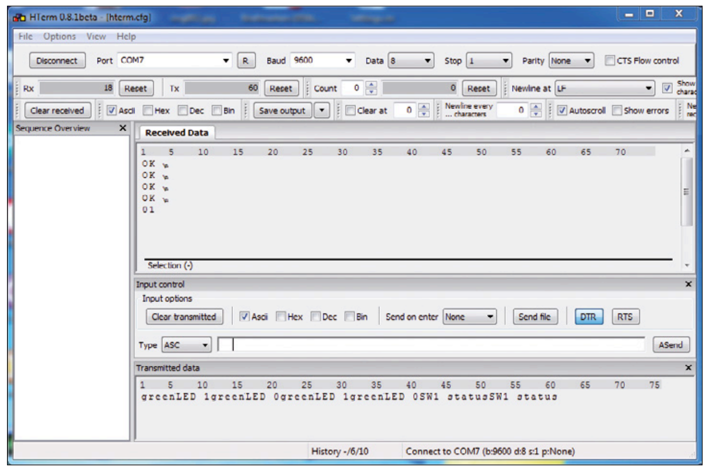


Figure 14. The program can be tested with the help of a terminal emulator. The required settings for HTerm are shown here.

Congratulations! You should now be in a position to look at building your own projects for the Cortex-M0+ microcontroller. Browse at your leisure through the various ASF libraries and the very helpful manuals. Some 400 pages of documentation on the libraries can be found at [6].

(140512)

Web Links

[1] http://asf.atmel.com

[2] www.atmel.com/images/doc8432.pdf

[3] www.atmel.com/Images/atmel-42129-sam-d20_datasheet.pdf

[4] www.atmel.com/Images/Atmel-42113-SAM-D20-D21-Port-Driver-PORT_Application-Note_AT03248.pdf

[5] www.atmel.com/Images/Atmel-42118-SAM-D20-Serial-USART-Driver-SERCOM-USART_Application-Note_AT03256.pdf

[6] www.atmel.com/Images/Atmel-42139-ASF-Manual-SAM-D20_Application-Note_AT03665.pdf

[7] www.elektor-magazine.com/140512

Q&A (Almost) Everything You Always Wanted to Know About… Level Shifting

Contributors: **Burkhard Kainka, Malte Fischer, Luc Lemmens, Clemens Valens**
Compiled by **Jaime González-Arintero**

5, 3.3, 2.5, 1.8, 1.5, 1.2 volts… as technology evolves, manufacturers move to lower voltage levels to reduce power requirements to the minimum. As with humans who do not speak the same language, not all components speak at the same 'loudness'. Although we can find several voltage-level translators in the market to do nearly all possible conversions, it's hard to not get *lost in translation*.
With such a broad subject, we know this short compilation may be just the tip of the iceberg, so if you have more Q's, we'll do our best to supply the A's… Just let us know!

Q ***Which voltages are considered as a logic 1 (normally: High), which as a logic 0 (normally: Low)?***

A Every logic family has threshold voltage levels accurately defining logic High" and logic Low, and it also changes for input and output signals. For 5 V TTL devices any input level exceeding 2 V will be read as a High (V_{IH}), and everything below 0.8 V, as Low (V_{IL}).
The output values are a bit more restrictive so to say, meaning that 2.7 V would be the minimum voltage provided by a device as a High signal (V_{OH}), and 0.4 V would be the maximum output voltage for a Low signal (V_{OL}). **Figure 1** illustrates these thresholds more clearly. With 3.3-V CMOS, all values are almost the same, except for the maximum voltage being lower (since V_{CC} is obviously 3.3 V). In some microcontrollers, these voltage levels could be slightly different. One of the most famous cases is the popular ATmega328, which provides greater noise margins.

But beware, when a pin "receives" an input signal with a level between these thresholds (meaning it's neither High nor Low), the response is "not defined". The situation is called floating state, and it could go High, Low or even flick from one to another. The main problem here is that 'limbo' input levels may cause the internal input circuits to draw excessive current. A typical mistake is having floating inputs in MCUs in Sleep mode, and measuring power consumption way above the specified values in the datasheet. It may drive us nuts until finding out why!

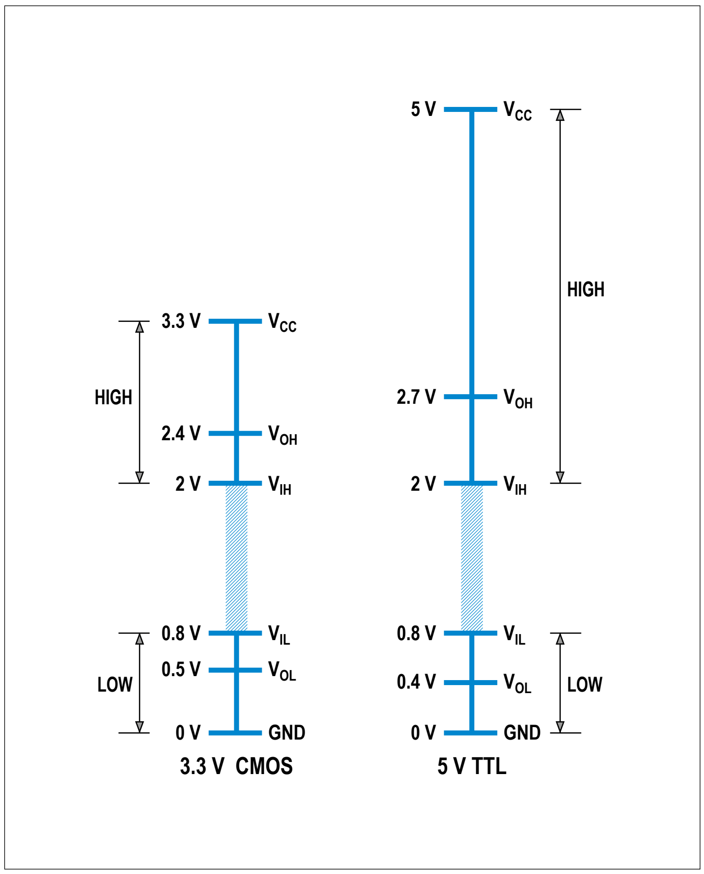


Figure 1. Logic High, Low or floating?

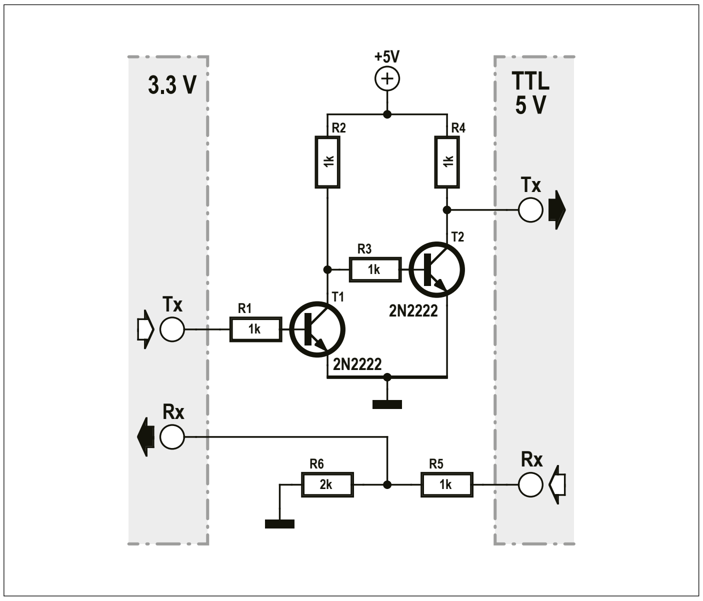


Figure 2. Logic-level *jump* from 3.3 V to 5 V. Simple yet effective.

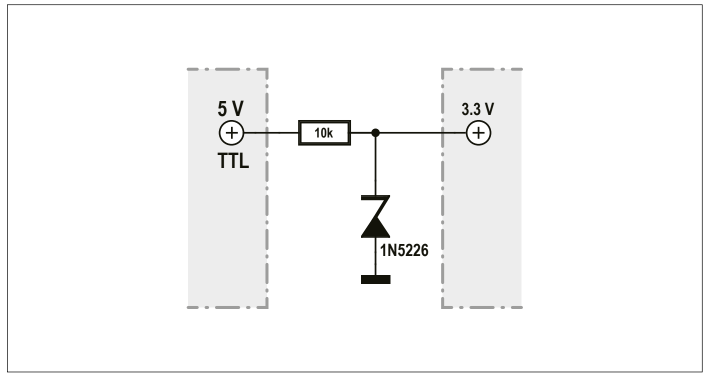


Figure 3. Unidirectional shifting from 5 V (TTL) to 3.3 V.

Q ***Are microcontroller inputs tolerant of too high levels (for instance, 5 V on a 3.3 V microcontroller)?***

A That strongly depends on the manufacturer/family/type of MCU, and thus it's always recommended to check the datasheet. If so, the number of 5-V tolerant I/Os will be clearly stated, as it happens with some low power microcontrollers.

Q ***What about level shifting from a 3.3-V device to 5 V (TTL)? And the other way around? Just a couple of resistors and we're ready to go?***

A Sometimes you want to interface a bunch of 'classic' parts running at 5 V TTL levels to a 3.3-V MCU. The shifting at the RX line is easy-peasy using a simple voltage divider, since we just want to lower the voltage. However, for TX, in order to be on the safe side, we need to raise the voltage swing. The task may be handled by a very simple circuit consisting of two NPN transistors (using only one would invert the signal) and a few resistors. **Figure 2** shows the suggested circuit. Although it's not the most elegant solution, it's really simple, cost-effective, and it gets the job done.

For unidirectional shifting from 5 V to 3.3V only (and not the other way around), a straightforward solution is to use a 3.3-V zener diode like the 1N5226, and a resistor. This will easily drop the voltage to a value slightly below the zener voltage. In the circuit depicted in **Figure 3** a 5-V level at the input will result in a voltage over 3 V, but certainly below 3.3 V.

Note that some solutions may not work with high-speed signals — it always depends on the case.

Q ***Do we need level shifting working with I²C?***

A In many cases, the open-collector configuration of I²C eliminates the need of level shifting by simply connecting the pull-ups on SDA and SCL to 3.3 V. However you should always check the datasheets of all components up for connection to the bus. For example, some 5 V I²C devices have a V_{IH} (minimum input voltage level interpreted as a logic High) of $0.7 \times V_{DD}$. In this case, it would be 3.5 V, so 3.3 V might be too low to ensure a safe margin for reliable I²C communication.

A very simple design shown in this application note from NXP Semiconductors [1] uses just two FETs and four resistors (two for each voltage level) to split the bus into a 3.3 V and a 5 V section.

Q ***When implementing level shifting with translator ICs like the TXB0108, sometimes the chip doesn't behave as expected, or problems are experienced with other surrounding components. Why?***

A Some level shifters have pull-up resistors on their I/Os that may interfere with the rest of the system, like I²C devices. There are many different types of integrated voltage-level translators, with different number of shifters, (mix of) mono- and bi-directional, automatic, or switched (meaning that the direction is determined by selection pin(s)), etc. and there's no general rule that applies to all of them regarding pull-up resistors. You should always refer to the datasheet.

(140560)

Web Link

[1] http://j.mp/AN10441

Stay tuned!

Q & A's next edition will cover...

Overvoltage and Overcurrent Protection

DesignSpark Tips & Tricks

Day #18: Copper Pours

By **Neil Gruending** (Canada)

Copper pours are a useful PCB layout tool — let's take a look at how to use them in DesignSpark.

I was working on the circuit boards for the Hactor project [1] recently when I needed to fill in some areas on the boards with copper in order to make some simple ground planes. This process is called pouring copper and the filled in areas are called copper pours. Today we will look at how use pours in DesignSpark.

Making a copper pour

The first step for making a copper pour is to draw an outline for where you would like the pour to go using the Add ➞ Copper Pour Area menu. I normally use the polygon tool but you can use the rectangle or the circle tools too. When you're ready to fill in the outline, right click on the outline and choose Pour Copper and you will see the Pour Copper Windows shown in **Figure 1**.

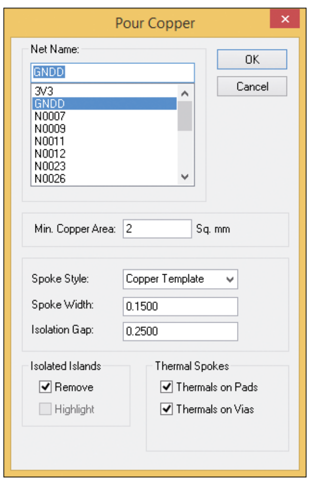


Figure 1. Copper pour window.

Here is where you can choose the net to use for the pour and all of the other basic pour parameters. Most of the options control thermal spoke connections that you can use within a pour to make soldering easier like **Figure 2**. The thermal Spoke Width and Isolation Gap defines the length and width of the small copper connections between the pad and the pour. Normally you would enable thermal connections on through-hole pads but I like to let pours directly connect to vias without thermals.

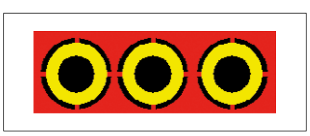


Figure 2. Thermal connections.

The minimum area parameter tells DesignSpark to ignore this pour if the calculated pour area is smaller than the specified minimum value. DesignSpark will also automatically delete isolated islands or any copper that isn't electrically connected to anything if the remove isolated islands parameter is checked.

Adjusting copper pours

Let's take a look at a simple example where I wanted to pour a ground plane around the 0.1 inch (2.54 mm) pitch header shown in **Figure 3**. The pour has connected to all of the ground pins on the connector but why didn't the pour go between the pins? It's pretty clear that there was enough room for the copper. The reason is that DesignSpark actually draws a copper pour as a sequence of lines so that it can be exported in a Gerber format later. It's like drawing the pour using a fixed width pen, and if the pen can't fit between the pads in our header then that's where the pour will stop. Fortunately DesignSpark uses the copper pour outline width as the pen width so making is smaller will allow DesignSpark to get the ground plane between the pads.

By default DesignSpark used a width of 0.3048 mm when I drew my outline, which is pretty big. You change the outline width by clicking on one of the segments and by changing its style. For this example I had to go down to 0.10 mm to get the pour to look like **Figure 4**. I figured this out by trial and error, but don't forget that you have to repour the copper every time you change the width. In general you want to make the outline only small enough to make the pour fit where you want it to go because going too small can cause problems later in manufacturing.

You can also control the spacing between pours and other objects by using the Spacings tab in the Design Technology window. For our header example the spacing rules would affect pin 3, which isn't connected to the ground pour. It would also affect how close the pour would get to other objects along its boundaries like the circuit board edge.

Keepouts

DesignSpark also allows you to exclude portions of a circuit board from all pours using pour keepouts. You define a keepout area using the copper pour drawing tools, except that you don't associate it with a net or pour it. Instead you right click on the outline and open the properties window like in where you check the Pour Keepout option. Now DesignSpark will treat the area as an object to be avoided and will follow all of the spacing rules to go around it.

Ordered pours

Pouring copper pours manually is fine when you only have one per layer or when pour areas don't intersect. That's because when pours intersect, the order that they are poured matters. For example, let's look at a modified version of our header example that contains another pour like in **Figure 5**. If you pour the small pour first and then the larger ground pour will

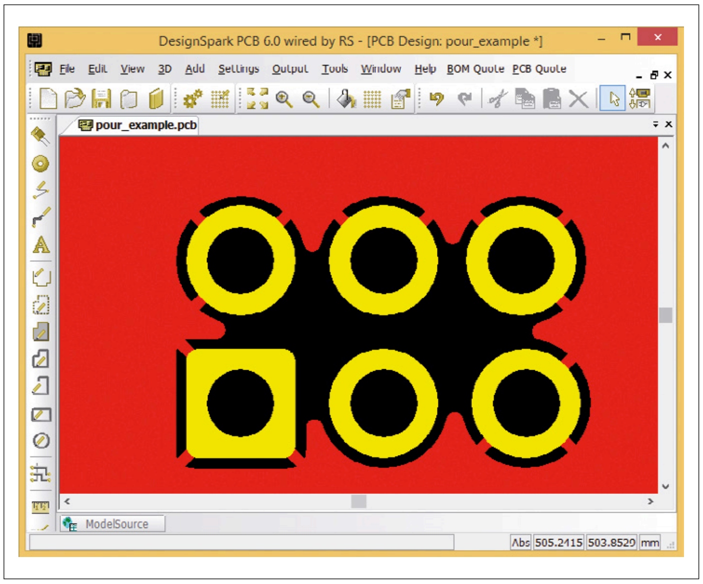


Figure 3. Initial header pour.

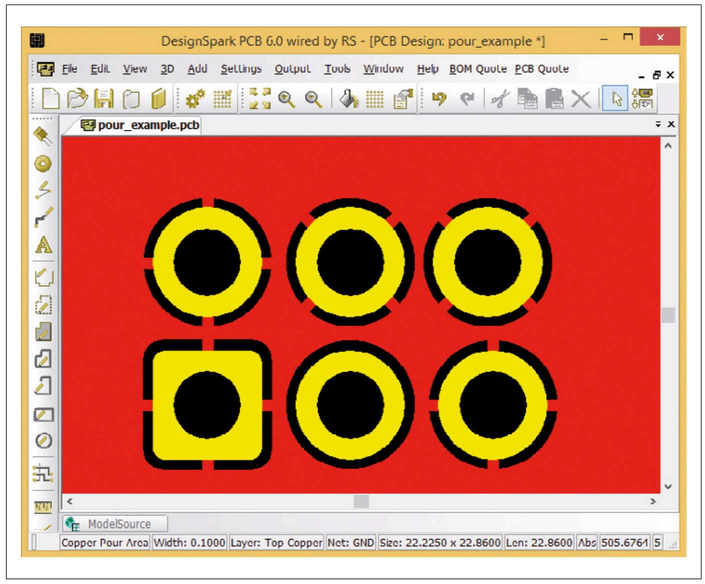


Figure 4. Final header pour.

go around it as intended. However if you pour the large ground pour first it will pour into the smaller pour's area and it wouldn't be possible to pour the smaller one later.

Fortunately DesignSpark allows you to specify the pour order for each pour so you don't have to. When DesignSpark is pouring all of the copper pours, it will first pour all of the pours with order number 0 (the default), then 1 and so on until all of the pours are finished. DesignSpark also allows you to have multiple pours set to each order number which lets you organize pours into groups. You set the order number for a pour using the Order field in the copper pour properties window.

Note that DesignSpark will only follow the pour order when repouring the entire board using the Pour Copper command when no pours are selected. If you run the Pour Copper command on a pour when it's selected than only that pour will be poured.

Clearing Pours

Normally you would add all of the pours to a board after you've finished routing it but what if you want to make changes after all of the pours are finished? If you try to route a track through a copper pour after it has been poured DesignSpark will raise a bunch of DRC errors and then you would still have to repour all of the affected areas anyways. A better solution is to use the Clear Copper command which will remove a poured copper area. It works just like the Pour Copper command in that it will remove a pour if one is selected, otherwise it will remove them all if none are selected. All of the copper pour area outlines will still remain but now you can make any changes to the board without raising DRC errors and then repour the copper areas after you're done.

Conclusion

'Today' we focused on how to use copper pours in DesignSpark. Next time we will look at some more of DesignSpark's PCB layout tools.

(140526)

Web Link

[1] www.elektor-labs.com/project/hacktor-the-elektor-dog-a-pcb-platform.13453.html

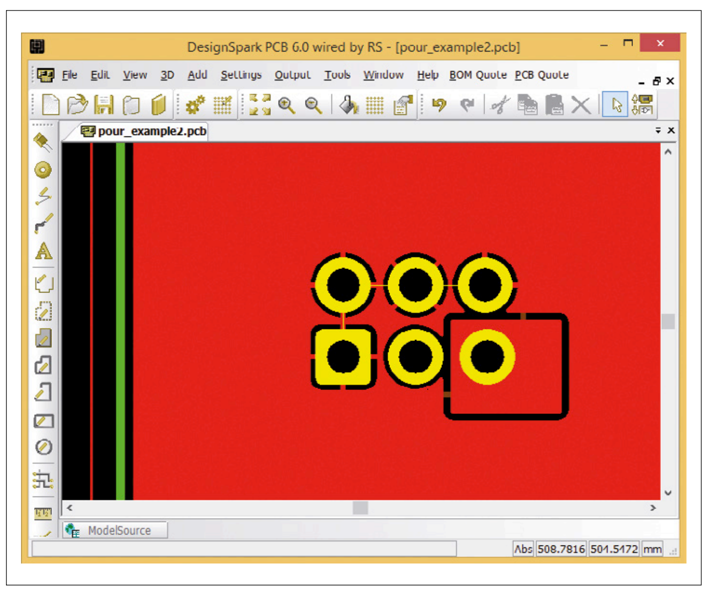


Figure 5. Intersecting copper pours.

MultiSIM Blue

A free CAD tool for schematic drawing, simulation and PCB design

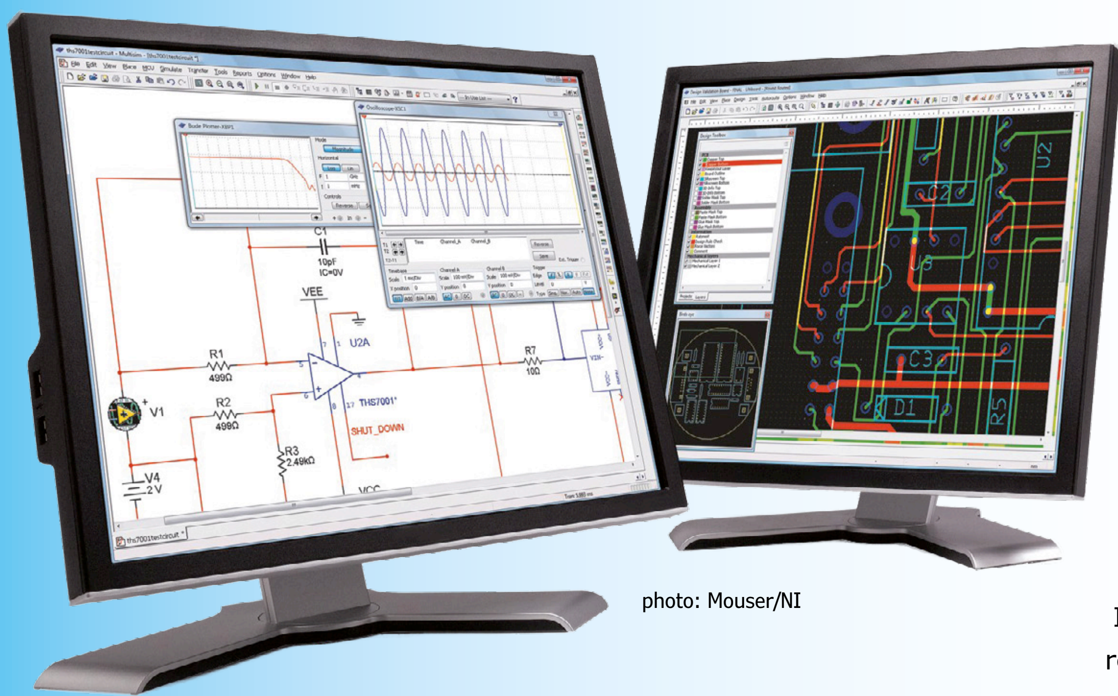


photo: Mouser/NI

By Harry Baggen (Elektor Netherlands Editorial)

A short while ago, parts vendor Mouser followed the lead of Farnell and RS with a free CAD package for drawing schematic diagrams and designing PCB layouts – a modified version of the well-known MultiSIM package from National Instruments. Here we review the package.

Figure 1. The start window offers several options.

In recent years, major parts vendors have been offering free software in an effort to attract customers and promote customer loyalty. Farnell (element14) was the first with a free version of Cadsoft's Eagle CAD package. Competitor RS Components, not wanting to be left behind, responded with the DesignSpark PCB package, followed later by DesignSpark Mechanical for 3D design. Mouser, another major player in the parts business, came out with its own free design software a few months ago, consisting of a modified version of the MultiSIM package from National Instruments. MultiSIM is an extensive software suite with a large range of features, and it costs a pretty penny if you buy the full version. I was therefore very interested to see how much the Mouser version is restricted in terms of features and functions — after all, it's a free version.

When you compare the various CAD packages, you see that the main limitations of the Eagle package are the maximum PCB size (half Eurocard) and just two copper layers. With DesignSpark there are not any real limitations, but some of its capabilities are not easy to use unless you are an expert (see our series of *DesignSpark Tips & Tricks* articles). The most significant limitation with MultiSIM Blue turns out to be the maximum number of components, which is 65. A major advantage of the MultiSIM Blue package is the ability to generate a bill of materials (BOM), which can then be used to order the

I was very interested to see how much the Mouser version is restricted in terms of features and functions -- after all, it's a free version!

parts directly from Mouser. However, what I like most about this software is the circuit simulation capability.

Installation

When you click the link to download the software, you are redirected to the National Instruments site. There you have to log in or create a new NI user account. After you do that, the download link appears. It downloads a small loader program, and when you run the loader it starts downloading the actual package, which tips the scales at 775 MB.

After the download is complete, you can start the installation. The overall package consists of a large number of modules, and it takes quite a while before they are all installed – more than enough for a coffee break. Remarkably, the license you have to accept for the installation states that this version is not allowed to be used for educational purposes. That's probably because NI doesn't want students and teachers to use this package, since they offer special educational versions with their own licenses. Another remarkable aspect is the stated duration of the software licenses. The license for MultiSIM runs until mid-2017, but the license for Ultiboard (the PCB layout program) is only good for one year. This is apparently related to the agreement between Mouser and NI, but we certainly hope this restriction will be relaxed. Otherwise you could be forced to switch to a different package just when you've gotten comfortable with this one. However, for the time being that isn't a real problem.

After completing the installation and restarting the computer, you can launch the program. The first thing to appear is a small selection window (**Figure 1**), which allows you to view a project BOM, download new components for the Mouser database, jump to some sample circuits, or run MultiSIM Component Evaluator (version 13.0).

I started by updating the database. That took a long time — more than 30 minutes — with the progress bar moving in a strange, jittery manner during the process. However, everything went okay and the result was a database containing nearly 100,000 components. That's a lot less than Mouser's full catalog product line, but I assume the database will be expanded at regular intervals.

MultiSIM displays the schematic drawing window when it starts up. If you have used this sort of software already, most things will look reasonably familiar. However, it still takes a while to get used to the details because (as usual) there are a whole lot of functions and buttons that work just a bit differently than what you are used to. The schematic drawing program is largely similar to the corresponding programs of the other packages. While playing with it I looked up several things with the Help function, but that turned out to be the general Help for the regular MultiSIM package. Sometimes the differences between the various versions are mentioned, but overall it's fairly confusing. It would be better to have a Help file specifically adapted to this version, with the irrelevant things removed.

MultiSIM has two different component databases, namely a general component library and a database of Mouser components. An advantage of the Mouser database is that it includes a lot of (physical) information and an order number for each component, but there's also a downside. If you want to place a resistor in your schematic from the database, you first have to select the type and brand of resistor you want to use. At that point you suddenly discover that there's very long list of resistors to choose from. Of course, after a while you know what type you normally use, but it still takes a bit of searching. If you want to simulate your circuit, you are well advised to use the general components library.

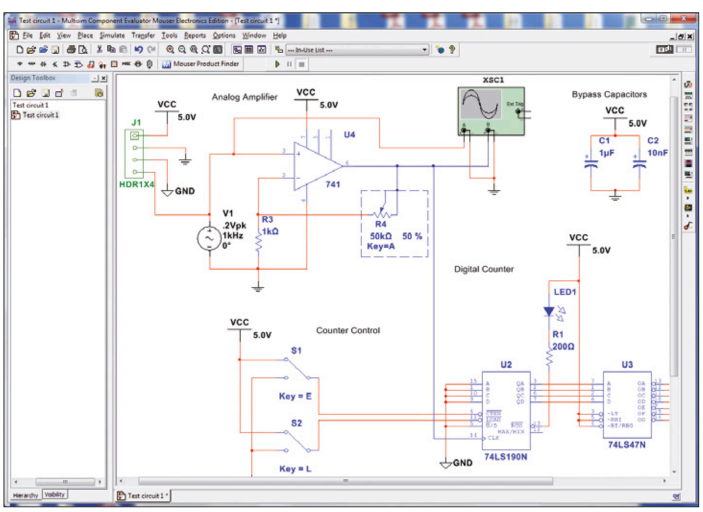


Figure 2. The schematic drawing window in MultiSIM. The switch at the top right starts the simulation.

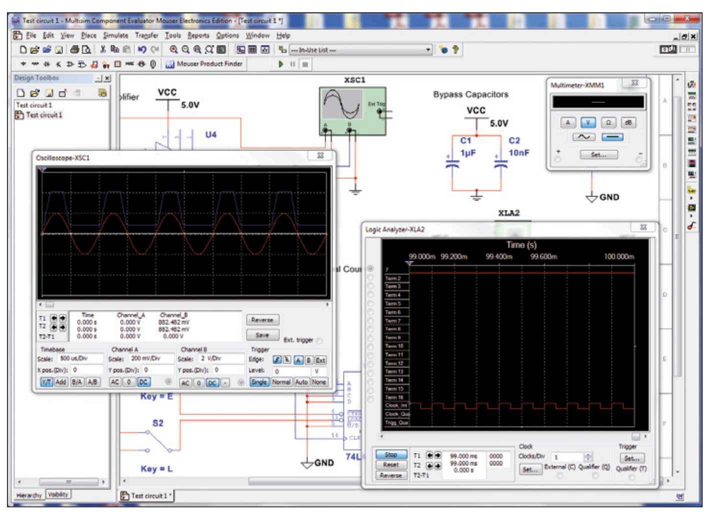


Figure 3. All sorts of instruments can be connected for simulation, and they all look and work a lot like real instruments.

MultiSIM is a polished CAD package with lots of features, including simulation capability -- something that no other free CAD package offers up to now.

Another disadvantage of the Mouser database is the limited choice of components. Although 100,000 components may sound like a large number, when you look closely you see (for example) that your choice of microcontrollers is limited to Microchip — with 4,287 different types and variants. A broader selection is urgently needed.

As previously mentioned, a special feature of this package is the simulation capability. MultiSIM, which is based on Electronics Workbench, started off as a pure simulation program and has been enhanced with a PCB layout program in the form of Ultiboard. The simulation part is very straightforward. You can easily add specific instruments to the schematic, and when you double-click an instrument you see an enlarged view. There you always have a control panel that looks a lot like that of a real instrument, which makes it fairly easy to use the instrument and adjust the settings. There are also interactive functions available, such as switches that you can actuate, LEDs that light up, or illuminated character displays. The simulator is mixed-mode, which means you can combine and simulate analog and digital circuits.

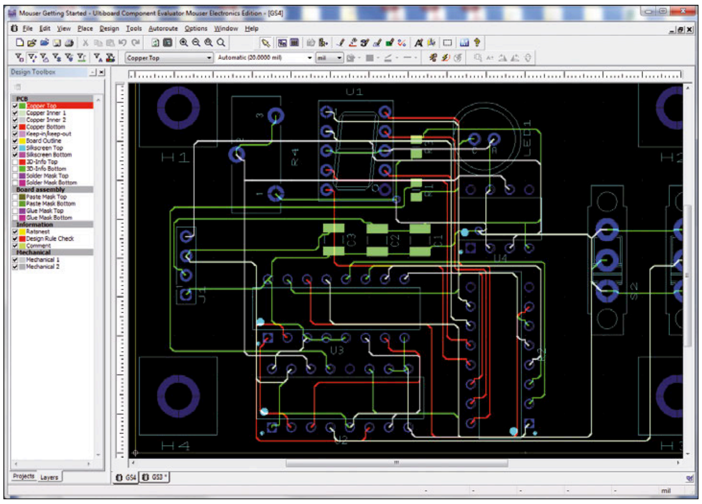


Figure 4. PCB layout design in the separate Ultiboard window. The autoplace and autoroute functions are handy for creating a rough layout.

After you have drawn the schematic (and run a simulation if you want to), you can press the Transfer button to send the netlist to the PCB layout program. That opens Ultiboard Component Evaluator in a separate window, where all the components in the schematic and their interconnections are shown along with a PCB outline. Then you can manually place all the components and see which arrangement gives the best results, or you can use the automatic placement function and allow the program to try to find a good layout. After that you can use the autorouter to sort out the track routing.
Ultiboard has lots of features and settings, but here again you notice that the way everything works is just a bit different from other PCB layout programs, so it takes some getting used to. The autoplacer and autorouter functions turn out to work very nicely, at least with the relatively simple examples I tried. There is also a 3D view function that lets you view the PCB with the components from all angles if the physical dimensions of the components on the board have been entered in the database. An important function that is missing in this “light” version of MultiSIM and Ultiboard is forward and backward annotation, which updates the PCB layout when changes are made in the schematic or updates the schematic when changes are made in the PCB layout. That means you have to handle updates manually. For example, if you change a resistor from 0.25 W to 1 W in the schematic, you have to adjust the PCB layout yourself. That’s a time-consuming and roundabout process, so hopefully things will be improved in future. It also means that this version is only suitable for small projects.

Once the PCB layout is finished, the Gerber files for the PCB fabricator can be generated. With the BOM generated by the program, you can recover some of the extra time you spent on the schematic drawing if you limited yourself to Mouser parts. After you save the BOM and click the “View Project BOM” button in the start window, you can see the prices of the selected components directly and put them in the shopping cart on your browser.

Conclusion

MultiSIM is a polished CAD package with lots of features, including simulation capability – something that no other free CAD package offers up to now. The idea of coupling it with the Mouser database so you can generate a BOM and an order list is clever, but the database needs to be enlarged drastically because a lot of current parts are missing. You can live with most of the limitations of this free version, such as the maximum number of components, as long as you keep your circuit designs fairly small. The lack of forward and backward annotation is a major weakness — the PCB layout has to be manually edited if any changes are made to the schematic, or vice versa, which takes a lot of patience and considerably increases the risk of mistakes. If this function were added, the package would be quite acceptable and a strong rival for other packages such as Eagle and DesignSpark.

(140314-I)

Web Link

www.mouser.com/multisimblue

Klystrons

Weird Component # 12

By Neil Gruending (Canada)

A few months ago we took a look at magnetrons and how they generate microwaves. Magnetrons work well when you just need a high power microwave energy source but they can't normally amplify a microwave signal like a klystron can. Klystrons aren't very common but they fill a very important niche in high power microwave applications like radar, communications and even atomic physics. Let's take a closer look at how they work.

A klystron is a vacuum tube that uses a microwave RF input to vary the velocity of electrons flowing through the tube.

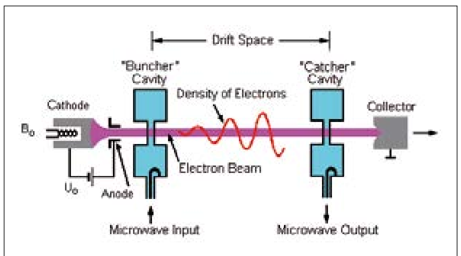


Figure 1. Klystron operation.
Source: Charly Whisky at English Wikipedia.

Figure 1 shows a simplified klystron that uses two cavities, one for the input and one for the output. The cathode is used to generate an electron beam by applying a very high voltage between the cathode and anode. Those electrons then travel to the first cavity which is called the buncher cavity which either speeds up or slows down electrons based on the microwave input. The electrons then go through the drift space which is long enough to let the electrons gather into groups by letting faster electrons catch up to slower moving ones. The catcher cavity then absorbs the electron groups and transmits them out of the microwave output. Any electrons that aren't absorbed by the catcher cavity are absorbed by the collector and are dissipated as heat. Modern klystrons usually use electromagnets to help focus the beam and utilize more than two cavities to increase the tube gain.

Another type of klystron is called a reflex klystron. It also modulates electron velocity but is constructed differently using a resonant cavity. The cavity allows the reflex klystron to oscillate and it's those oscillations that modulate the electron velocity. With some exceptions modern semiconductors have replaced reflex klystrons.

Large radar systems and high power microwave transmitters use large klystrons like in **Figure 2**, some of which are capable of up to tens of megawatts of output power. But sometimes even that isn't enough power. For example, Stanford University is working on a linear particle accelerator that needs thousands of 75-MW klystrons to power it. Those types of systems are out of reach for hobbyists, but fortunately for us there's the 2K25 klystron.

Figure 2. Klystron.
Source: Enoch Lau at English Wikipedia.

2K25 klystrons are readily available and their low power output makes them much more practical to experiment with. They are a reflex klystron design and were typically used as the local oscillator in 3-cm (9.6-GHz) radar receivers. They came in two different versions, one where you tune the oscillation frequency using a screw (**Figure 3a**) and the other which is tuned electronically using a tuning diode (**Figure 3b**). The best part about them though is that there's a lot of information online available about how to use them, so why not give it a try?

(140535)

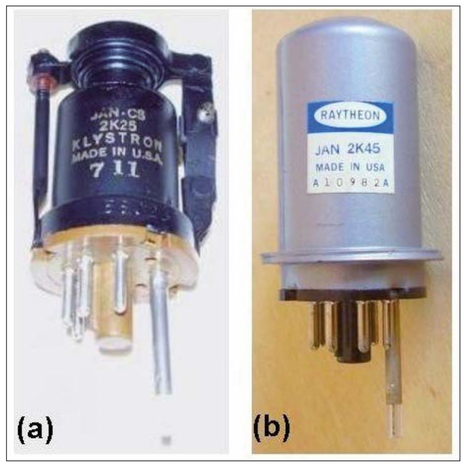


Figure 3. 2K25 Klystrons.
Source: www.slac.stanford.edu/cgi-wrap/getdoc/slac-pub-7731.pdf.

Tips and Tricks

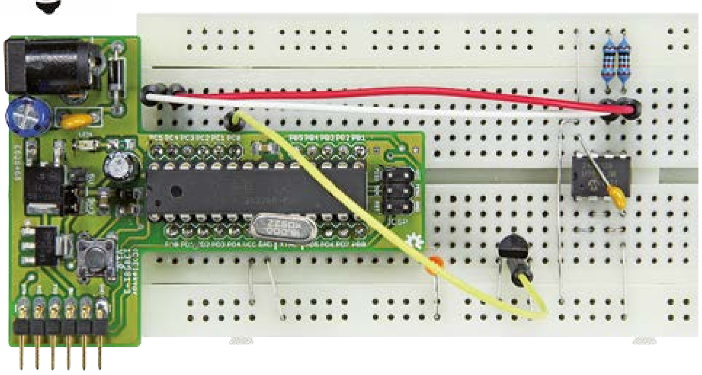
From readers for readers

Compiled by **Ralf Schmiedel**

We are always pleased to read the clever solutions our readers have devised to solve some tricky problems. Also of interest are your suggestions for new projects. We will be publishing a selection of your contributions here.

Capacitor current measurement

Trigger: T-Board 28: Lowest-Power Exercising, Elektor December 2014, p. 20 (140413)



Dear Elektor Editor,

An elegant and low cost way to measure current consumption of low-energy microprocessors is to measure the discharge time of a capacitor, used as the power source.

Use a high quality electrolytic capacitor in the range of 100 µF to 2000 µF and if possible, measure the value accurately. Charge the capacitor to 3.3 V, connect the microprocessor and watch the voltage level drop when the microprocessor is active. The sink rate indicates the current drawn.

The method can be used to a minimum operating voltage of approximately 2.7 V.

The current is given by: $I = C \times \Delta V / \Delta t$

A multimeter and a stop watch will allow you to get values for ΔV and Δt (you should take into account capacitor self-discharge). This works really well, shows when current flows and costs almost nothing.

Sven Guttke

Hi Sven,
Neat idea, the instantaneous current drawn by a low-power microcontroller is likely to be fluctuating depending on work load. This method shows the charge removed from the capacitor and is therefore effectively integrating the current. Incidentally you will find that many of the newer DVMs now have a capacitance measurement range.
Ralf Schmiedel

Be kind to your Probes

Oscilloscope probes should be treated with respect, the cable has a very thin inner conductor that is quite fragile. Make sure that the scope probe body isn't allowed to fall from the bench while the other end is plugged into the scope and don't leave a probe on the floor where it can be stood on. Keep them away from corrosive fluids.

While most scope probes appear to be quite robust they do not respond well to misuse and can suddenly fail. Worse still they can develop an intermittent fault and give unreliable measurements.

A probe should only be used with the instrument for which it has been calibrated; indiscriminate swapping of probes can give rise to errors.

Hanan Boasson

Lumen instead of Lux

Reference: Lux Meter with 1 lx to 100 klx measuring ranges, Elektor October 2014, p. 42 (130109)

Dear Elektor team,

I found your article on the Lux meter very interesting. It rightly points out that the light output claimed by Far East suppliers of LEDs should be subject to scrutiny to say the least.

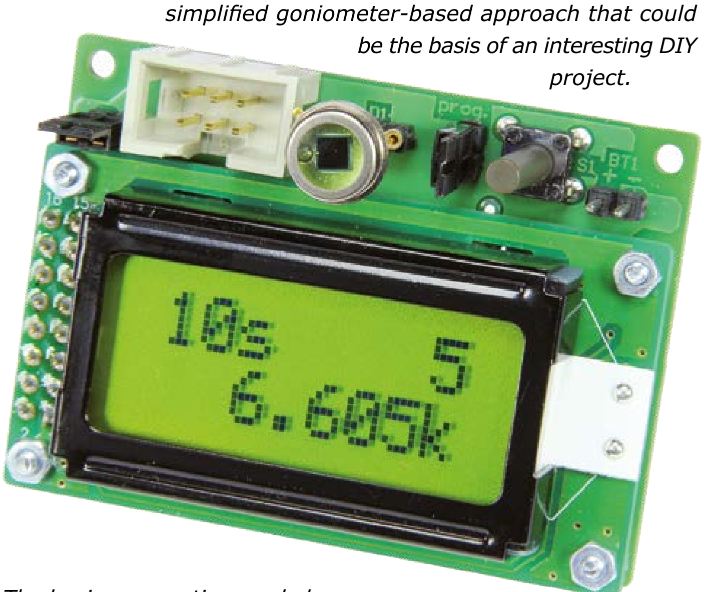
I was a bit disappointed to find that the project describes a Lux meter which can't make the measurements referred to.

A low-cost Lumen meter to measure the characteristics of a light source would be interesting. As far as I know there isn't one available.

The two methods used to measure Lumens use:

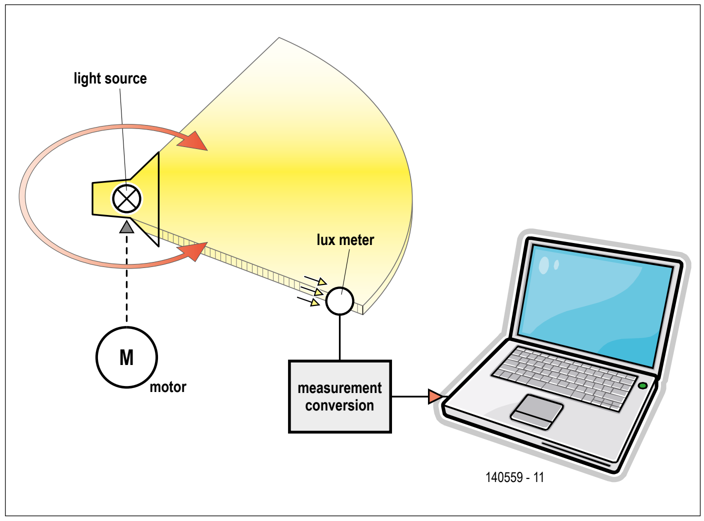
1. *An integrating sphere*
2. *A goniometer*

Both methods are used in conjunction with a spectrometer to give accurate results. A measuring setup with the integrating sphere (Ulbricht sphere) works out at about $30,000 minimum. A goniometer-based system is not much cheaper. There is however a simplified goniometer-based approach that could be the basis of an interesting DIY project.



The basic assumption made here is that the light pattern from most sources is approximately symmetrical. To make an estimate of the output over the full sphere we can get away with measuring a linear profile at right angles to the main axis. Mechanically this could be achieved by attaching the light source to a rotatable frame. A spectrometer can be used to measure the intensity from a number of points over a 180° half circle to build up the profile. At each step the value of illuminance (E) in Lux is measured and stored together with the area of the 'strip' of the sphere (A) in m² (The formula can be found in Wikipedia). The luminous flux Q = E A can then be calculated for the step angle. Summing all the measurements made in the half-circle gives a figure for the total luminous flux in Lumens.

The light source could be mounted on a rotatable frame driven by a stepper motor working in voltage-mode so the circuit would not need to be too complex, and wouldn't require microstepping. A bridge driver circuit could do the job and it would only need a couple of output signals from a microcontroller.



A note on the spectrometer: The 'Lumen' is a physiologically weighted value of the total radiant flux (in W). The weighting takes into account the sensitivity characteristics of the eye – the Lumen can only be spoken of in terms of perceived brightness by a human observer.

To make sure the Lumen or Lux measurement is accurate it's necessary to take into account the (V-Lambda-curve, CIE 1931 Definition, see Wikipedia) weighting curve. For this reason most measurement systems use a spectrometer, which allows the spectral content of the light source to be evaluated to calculate the Lux or Lumen value.

Without taking this into consideration the measurement will be badly distorted. Take a look at 'Fig. 5 – Relative Spectral Sensitivity vs. Wavelength' in the data sheet for the BPW21 photodiode: www.vishay.com/docs/81519/bpw21r.pdf. You can see that the sensitivity of the sensor at 450 nm is approximately ten times higher compared to the eye's sensitivity. Daylight-type LEDs pump out a good deal of blue light around this wavelength which would give a false reading when measuring LED light sources.

Read more about this in this Osram application note:

http://bit.ly/1z0gTOT

The SFH5711 is designed for precision measuring and has the correct spectral sensitivity:

http://bit.ly/14LJGtK

It also has a logarithmic output characteristic which makes it easier to use for light sources with a wide dynamic range (from 3 to 80 klx).

Martin Melzer

Editor's note: In his original email Martin refers to the LightSpion Lumen measuring device which uses the method he describes above. The company Viso Systems:

www.visosystems.com/products/lightspion

market this equipment and their web site indicates that a patent is pending on the design of their system.

Say Hello to: Rpi Compute Module

CM + CM IO = Super RPi

By **Tony Dixon** (UK)

The Raspberry Pi has given students and enthusiasts access to a cheap and powerful embedded computing platform that just didn't exist beforehand. Less well known is the number of commercial projects the Pi has found its way into. With this in mind the Raspberry Pi Foundation extended the Pi platform into the commercial sector. The result is the Raspberry Pi Compute Module and its optional host board to harness all that I/O.

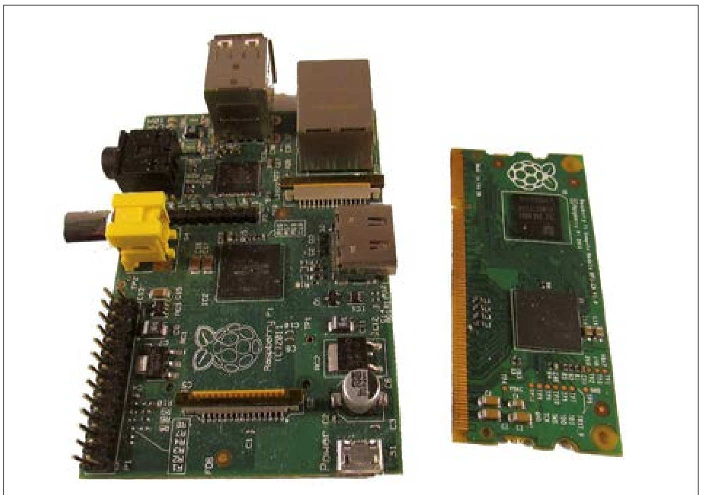


Figure 1. Raspberry Pi computer (left) and the new Raspberry Pi Compute Module (right).

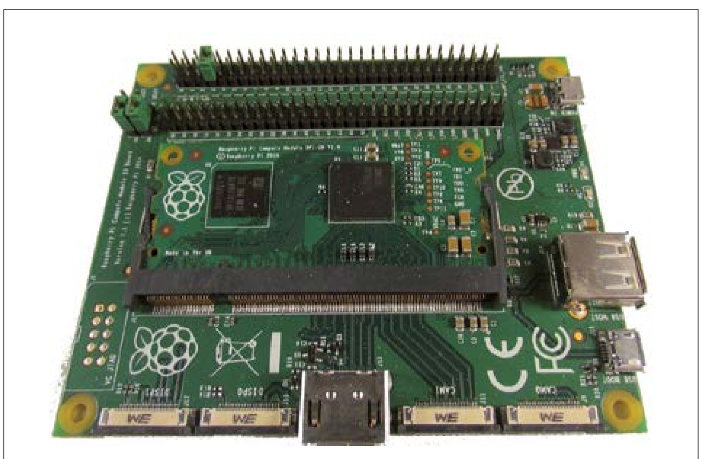


Figure 2. Such amounts of I/O, supply, and control lines packed on 200 microscopic SO-DIMM PCB edge contacts calls for a dedicated motherboard to be workable. Here the CM is seen plugged onto the CM IO Board, the assembly creating a kind of Super RPi.

The CM is a vision of simplicity. It has just two chips: the first the same Broadcom 2835 SoC with 512 MB RAM found on the Raspberry Pi Model B, and the second a 4-GByte eMMC Flash memory device. These are mounted onto a 200-contact SO-DIMM shaped board similar to a laptop DDR2 SO-DIMM. Highlights of the RPi Compute Module's specification are shown in **Table 1**.

Having done away with the connectors and adopting the SO-DIMM footprint has allowed the CM to be not only a much smaller unit than the RPi but also a much more versatile platform, with access to almost all of the Broadcom 2835 signals, including the majority of GPIO signals (46 GPIO's in fact). For these 46 GPIO's we have up to six alternative functions, so we have access to the interfaces listed in **Table 2**. You can compare the sizes of the RPi and the RPi CM in **Figure 1**.

I/O, I/O, more I/O

CM GPIO details are listed in **Table 3**. The standard Raspberry

Table 1. RPi Compute Module (CM) design highlights.

System on Chip (SoC):	Broadcom BCM2835
Architecture:	ARM1176JZF-S
Clock:	700 MHz
GPU:	VideoCore IV GPU
RAM:	512 MBytes
Interfaces:	HDMI, TV Out, USB, 2x DSI LCD, 2x CSI Camera
Size:	67.6mm x 30mm (SODIMM)

Table 2. RPi-CM GPIO alternative functions.

Number	Alternative Function
1	SD-Card Interface
2	UART (one with CTS & RTS signals)
2	I²C
2	SPI (one with 2 chip selects and the other with 3 chips selects)
2	PWM outputs
1	I²S/PCM
1	Secondary Address/Data Bus
3	General Purpose Clock Outputs
46	GPIO

Pi Model A and B signals are there so existing Pi add-on boards can be used. As well as these, the additional GPIO signals and additional interfaces allow for much larger control scenarios.

The CM's SO-DIMM pinouts are shown in **Table 4**. Besides the GPIO signals the SO-DIMM pinout brings out the standard HDMI, TV Out, USB, DSI LCD and CSI Camera ports from the original Pi. In addition it also includes access to a second DSI LCD and a second CSI camera port. The CM can be plugged onto another new product called the Raspberry Pi Compute Module I/O Board. The assembly of CM plugged onto the CM I/O board is shown in **Figure 2**.

Getting the CM started requires three separate power supplies, 3.3 V, 1.8 V, and 2.5 V. Ideally these should be SMPS circuits. If your application doesn't use the TV-out function then the 2.5-V supply is not required, simplifying things slightly.
More design details can be found on the Raspberry Pi Github site [1].
The CM is designed to boot from built-in eMMC, making it a self-contained unit.

Conclusion

The Raspberry Pi Compute Module and the associated I/O host board are a welcome addition to the Raspberry Pi family of

Table 3. RPi-CM GPIO functionality.

GPIO ID	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
GPIO0	SDA0	SA5	<reserved>			
GPIO1	SCL0	SA4	<reserved>			
GPIO2	SDA1	SA3	<reserved>			
GPIO3	SCL1	SA2	<reserved>			
GPIO4	GPCLK0	SA1	<reserved>			ARM_TDI
GPIO5	GPCLK1	SA0	<reserved>			ARM_TDO
GPIO6	GPCLK2	SOE_N/SE	<reserved>			ARM_RTCK
GPIO7	SPI0_CE1_N	SWE_N/SRW_N	<reserved>			
GPIO8	SPI0_CE0_N	SD0	<reserved>			
GPIO9	SPI0_MISO	SD1	<reserved>			
GPIO10	SPI0_MOSI	SD2	<reserved>			
GPIO11	SPI0_SCLK	SD3	<reserved>			
GPIO12	PWM0	SD4	<reserved>			ARM_TMS
GPIO13	PWM1	SD5	<reserved>			ARM_TCK
GPIO14	TXD0	SD6	<reserved>			TXD1
GPIO15	RXD0	SD7	<reserved>			RXD1
GPIO16	<reserved>	SD8	<reserved>	CTS0	SPI1_CE2_N	CTS1
GPIO17	<reserved>	SD9	<reserved>	RTS0	SPI1_CE1_N	RTS1
GPIO18	PCM_CLK	SD10	<reserved>	BSCSL_SDA/MOSI	SPI1_CE0_N	PWM0
GPIO19	PCM_FS	SD11	<reserved>	BSCSL_SCL/SCLK	SPI1_MISO	PWM1
GPIO20	PCM_DIN	SD12	<reserved>	BSCSL/MISO	SPI1_MOSI	GPCLK0
GPIO21	PCM_DOUT	SD13	<reserved>	BSCSL/CE_N	SPI1_SCLK	GPCLK1
GPIO22	<reserved>	SD14	<reserved>	SD1_CLK	ARM_TRST	
GPIO23	<reserved>	SD15	<reserved>	SD1_CMD	ARM_RTCK	
GPIO24	<reserved>	SD16	<reserved>	SD1_DAT0	ARM_TDO	
GPIO25	<reserved>	SD17	<reserved>	SD1_DAT1	ARM_TCK	
GPIO26	<reserved>	<reserved>	<reserved>	SD1_DAT2	ARM_TDI	
GPIO27	<reserved>	<reserved>	<reserved>	SD1_DAT3	ARM_TMS	
GPIO28	SDA0	SA5	PCM_CLK	<reserved>		
GPIO29	SCL0	SA4	PCM_FS	<reserved>		
GPIO30	<reserved>	SA3	PCM_DIN	CTS0		CTS1
GPIO31	<reserved>	SA2	PCM_DOUT	RTS0		RTS1
GPIO32	GPCLK0	SA1	<reserved>	TXD0		TXD1
GPIO33	<reserved>	SA0	<reserved>	RXD0		RXD1
GPIO34	GPCLK0	SOE_N/SE	<reserved>	<reserved>		
GPIO35	SPI0_CE1_N	SWE_N/SRW_N	<reserved>			
GPIO36	SPI0_CE0_N	SD0	TXD0	<reserved>		
GPIO37	SPI0_MISO	SD1	RXD0	<reserved>		
GPIO38	SPI0_MOSI	SD2	RTS0	<reserved>		
GPIO39	SPI0_SCLK	SD3	CTS0	<reserved>		
GPIO40	PWM0	SD4		<reserved>	SPI2_MISO	TXD1
GPIO41	PWM1	SD5	<reserved>	<reserved>	SPI2_MOSI	RXD1
GPIO42	GPCLK1	SD6	<reserved>	<reserved>	SPI2_SCLK	RTS1
GPIO43	GPCLK2	SD7	<reserved>	<reserved>	SPI2_CE0_N	CTS1
GPIO44	GPCLK1	SDA0	SDA1	<reserved>	SPI2_CE1_N	
GPIO45	PWM1	SCL0	SCL1	<reserved>	SPI2_CE2_N	

products. The CM is bristling with GPIO and interfaces. If you have a project and the existing Pi Model A/B footprint doesn't meet your space then the CM is ideal for you, allowing you to build the perfect board and enclosure for your needs. Likewise, if you have an advanced hardware project in mind then the additional I/O will be a great benefit if not a must have.

(140320)

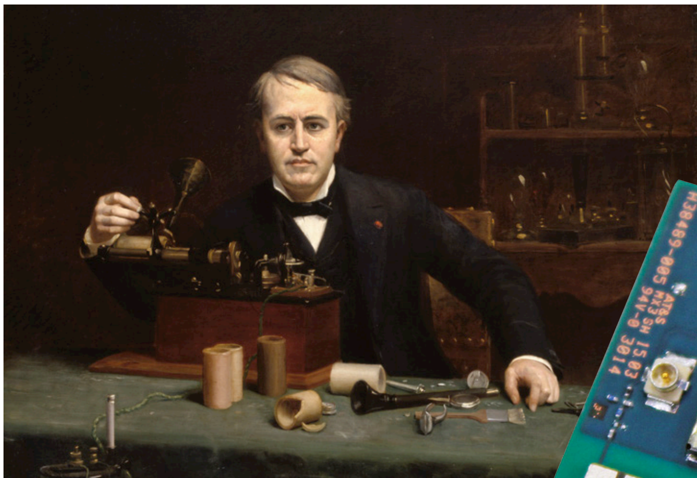
Reference

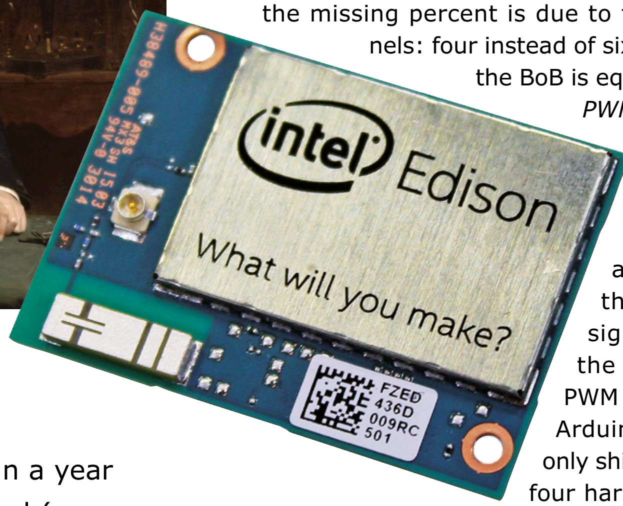
[1] RPi Compute Module Hardware Design Guide: https://github.com/raspberrypi/documentation/tree/master/hardware/computemodule

Table 4. RPi CM 120-pin SO-DIMM pin functions.

GND	1	2	EMMC_DISABLE	DSI0_DP0	101	102	DSI1_CN
GPIO0	3	4	Not Connected	GND	103	104	GND
GPIO1	5	6	Not Connected	DSI0_CN	105	106	DSI1_DP3
GND	7	8	Not Connected	DSI0_CP	107	108	DSI1_DN3
GPIO2	9	10	Not Connected	GND	109	110	GND
GPIO3	11	12	Not Connected	HDMI_CK_N	111	112	DSI1_DP2
GND	13	14	Not Connected	HDMI_CK_P	113	114	DSI1_DN2
GPIO4	15	16	Not Connected	GND	115	116	GND
GPIO5	17	18	Not Connected	HDMI_D0_N	117	118	DSI1_DP1
GND	19	20	Not Connected	HDMI_D0_P	119	120	DSI1_DN1
GPIO6	21	22	Not Connected	GND	121	122	GND
GPIO7	23	24	Not Connected	HDMI_D1_N	123	124	Not Connected
GND	25	26	GND	HDMI_D1_P	125	126	Not Connected
GPIO8	27	28	GPIO28	GND	127	128	Not Connected
GPIO9	29	30	GPIO29	HDMI_D2_N	129	130	Not Connected
GND	31	32	GND	HDMI_D2_P	131	132	Not Connected
GPIO10	33	34	GPIO30	GND	133	134	GND
GPIO11	35	36	GPIO31	CAM1_DP3	135	136	CAM0_DP0
GND	37	38	GND	CAM1_DN3	137	138	CAM0_DN0
GPIO0-27_VREF	39	40	GPIO0-27_VREF	GND	139	140	GND
GPIO28-45_VREF	41	42	GPIO28-45_VREF	CAM1_DP2	141	142	CAM0_CP
GND	43	44	GND	CAM1_DN2	143	144	CAM0_CN
GPIO12	45	46	GPIO32	GND	145	146	GND
GPIO13	47	48	GPIO33	CAM1_CP	147	148	CAM0_DP1
GND	49	50	GND	CAM1_CN	149	150	CAM0_DN1
GPIO14	51	52	GPIO34	GND	151	152	GND
GPIO15	53	54	GPIO35	CAM1_DP1	153	154	Not Connected
GND	55	56	GND	CAM1_DN1	155	156	Not Connected
GPIO16	57	58	GPIO36	GND	157	158	Not Connected
GPIO17	59	60	GPIO37	CAM1_DP0	159	160	Not Connected
GND	61	62	GND	CAM1_DN0	161	162	Not Connected
GPIO18	63	64	GPIO38	GND	163	164	GND
GPIO19	65	66	GPIO39	USB_DP	165	166	TVDAC
GND	67	68	GND	USB_DM	167	168	USB_OTGID
GPIO20	69	70	GPIO40	GND	169	170	GND
GPIO21	71	72	GPIO41	HDMI_CEC	171	172	VC_TRST
GND	73	74	GND	HDMI_SDA	173	174	VC_TDI
GPIO22	75	76	GPIO42	HDMI_SCL	175	176	VC_TMS
GPIO23	77	78	GPIO43	RUN	177	178	VC_TDO
GND	79	80	GND	VDD_CORE	179	180	VC_TCK
GPIO24	81	82	GPIO44	GND	181	182	GND
GPIO25	83	84	GPIO45	1V8	183	184	1V8
GND	85	86	GND	1V8	185	186	1V8
GPIO26	87	88	GPIO46_1V8	GND	187	188	GND
GPIO27	89	90	GPIO47_1V8	VDAC	189	190	VDAC
GND	91	92	GND	3V3	191	192	3V3
DSI0_DN1	93	94	DSI1_DP0	3V3	193	194	3V3
DSI0_DP1	95	96	DSI1_DN0	GND	195	196	GND
GND	97	98	GND	VBAT	197	198	VBAT
DSI0_DN0	99	100	DSI1_CP	VBAT	199	200	VBAT

Intel Edison: What Will You Make?





By **Clemens Valens** (Elektor.Labs)

At the end of September 2014, less than a year after the introduction of the Galileo board (now at revision 2) Intel launched its tiny Edison module. Marketed as a low-cost, product-ready, general purpose compute platform it is targeted at entrepreneurs of all sizes, from pro makers (serious amateurs working out of their garage like Steve & Steve did in the late seventies) to consumer electronics and companies working in the Internet of Things range (IoT) and wearable computing products.

It only takes a glance at the specifications (see inset) to discover the impressive feature set of the Edison module. It has the looks of the innards of a smartphone (the SoC is said to be a smartphone SoC) and it is probably not too difficult to make one based on an Edison module.

Yocto Linux

Edison ships with Yocto Linux installed (even though you are supposed to update it immediately). The Yocto Project aims at making the creation of custom Linux distributions for embedded products easy. For the typical Edison user this is of no importance, although it does offer a lot of flexibility for the advanced user who wants to customize the OS.

Even Edison uses Arduino

Without a breakout board (BoB) the Edison has little use, hence Intel offers two hardware kits to get started with the module: the Edison kit for Arduino and the Edison breakout board. Their **specifications** are printed separately. In this article I will use the kit for Arduino.

The Arduino BoB is 99% compatible with the Arduino 1.0 pinout, the missing percent is due to two missing PWM channels: four instead of six. To compensate for this the BoB is equipped with the so-called *PWM swizzler* (a swizzler is a hotdog with a spiral cut that more than triples the marinade absorbing surface of the sausage), a clever jumper block that lets you move PWM signals around. Thanks to the swizzler you can route PWM signals to each of the six Arduino PWM pins, penalizing only shields that need more than four hardware PWM signals (you can of course do PWM in software). The swizzler seems a good solution, but unfortunately there is a snag. The Edison PWM outputs have a default Arduino pin. If you move such an output to another Arduino pin then the default pin is effectively unconnected.

> Without a breakout board (BoB) the Edison has little use, hence Intel offers two hardware kits to get started with the module

So only use the swizzler if you really have to.

There are a few other subtleties that may get you into trouble, so if you want to get serious with this BoB I suggest you read the “Intel Edison Kit for Arduino Hardware Guide” very carefully from beginning to end.

Getting started

Initially you download large quantities of bytes from the Intel website. You will need the Arduino IDE for Edison (and Galileo), USB serial port drivers from FTDI and some more drivers for the Edison board from Intel and you should download the latest Yocto image. Except for the FTDI drivers all these files can be found on the Edison download page. While you wait for the downloads to complete, go find either two micro USB cables or, if you don’t have two of these cables like me, one micro USB cable and a 12-V power adapter (I did this, but it

Edison Module Specifications

- 22 nm Intel SoC that includes a dual core, dual threaded Intel Atom CPU at 500 MHz and a 32-bit Intel Quark microcontroller at 100 MHz
- 40 general purpose I/O (GPIO) ports, configurable e.g. as SD card interface, 2x UART, 2x I²C, SPI, I²S, 4x PWM, USB 2.0 OTG
- 1 GB RAM (LPDDR3)
- 4 GB flash memory (eMMC)
- Dual-band Wi-Fi
- Bluetooth 4.0
- Power supply: 3.3 to 4.5 V
- Power consumption in standby mode: no radios: 13 mW; Bluetooth 4.0: 21.5 mW; Wi-Fi: 35 mW
- 35.5 × 25.0 × 3.9 mm (1.4 × 1.0 × 0.15 inches)
- 0 to 40°C (32 to 104°F)
- Yocto Linux v1.6

Edison Board for Arduino Specifications

- Supports Arduino Sketch, Linux, Wi-Fi, and Bluetooth
- Board I/O compatible with Arduino Uno (except 4 PWM instead of 6 PWM)
- 20 digital I/O pins, including 4 (not 6) pins as PWM outputs
- 6 analog inputs
- UART (Rx/Tx)
- I²C
- ICSP 6-pin header (SPI)
- Micro USB device connector OR (via mechanical switch) dedicated standard size USB host Type-A connector
- Micro USB device (connected to UART)
- SD card connector
- DC power supply jack (7 to 15 VDC input)

Edison Breakout Board Specifications

- Slightly larger than the Edison module
- Exposes native 1.8 V I/O of the Edison module
- 0.1 inch grid I/O array of through-hole solder points
- USB OTG with USB Micro Type-AB connector
- USB OTG power switch
- Battery charger
- USB to device UART bridge with USB micro Type-B connector
- DC power supply jack (7 to 15 VDC input)

Web Link

[1] Intel Edison: www.intel.com/content/www/us/en/do-it-yourself/edison.html

will result in a lot of cable swapping). You should also assemble your kit. Contrary to Intel I think you are clever enough to figure this part out all by yourself. In case you run into problems, check the Intel website [1].

Then:

1. Install all the drivers on your computer; be sure to do this with administrator privileges.
2. Copy the latest Yocto image to the Edison module—for this you need a micro USB cable connected to J16, the micro USB connector closest to the switch. After connecting the cable to the PC the module will show up as a thumb drive. Make sure that it is empty (if necessary delete all files present), then unpack the Yocto image to the Edison thumb drive.
3. Flash the Edison module with the Yocto image—connect the micro USB cable to the other micro USB connector (J3), the one in the corner of the board. A serial port will appear. Get its number from the Device Manager (Windows, detailed procedures for Linux and Mac users can be found on the Intel website [1]) and open this port with a terminal program like Tera Term or PuTTY and with port settings 115200n81. Press Enter twice to get the login prompt (“edison login:”). Type “root” followed by enter, then “reboot ota” (followed by enter). Wait until the login prompt reappears (after about two minutes) and you’re done.
4. Unpack the Arduino IDE and launch arduino.exe.
5. Reconnect the micro USB cable to J16, the micro USB connector closest to the switch. Besides the thumb drive this will also create another serial port—Intel Edison Virtual Com Port (COM33 in my case)—that you will use for Edison Arduino programming.
6. In the Arduino IDE click Tools followed by Board and select the Edison module.
7. In the Arduino IDE click Tools followed by Serial Port and select the port that corresponds to the Intel Edison Virtual Com Port (COM33 in my case).
8. In the Arduino IDE click Files→Examples→01.Basics→Blink to load the Hello World test program.
9. In the Arduino IDE click the Upload button, sit back and wait until LED DS2 (close to J4 in the middle of the board) starts blinking.

Adding Wi-Fi

If you got this far you are ready to start Arduino programming as if you had a very complex almost compatible Arduino board. Let’s go one step further and connect the Edison to your Wi-Fi network. Here is how:

1. Connect the micro USB cable to micro USB connector (J3), the one in the corner of the board and open the serial port from the previous Step 3 with a terminal program like Tera Term or PuTTY and with port settings 115200n81. Press Enter twice to get the login prompt (“edison login:”). Type “root” followed by enter.
2. Type in “configure_edison --setup” and press Enter (note the double dash in front of “setup”). This will start a wizard that will take you through the motions. If you

don't want to do a certain step, simply skip it by pressing Enter.

3. If you go through all the steps you should end up with a message saying that you can now connect to the module with a browser at IP address such and so (192.168.2.46 in my case). Do so and if you see a welcome page you are lucky. I wasn't, even though my router listed it, but without a name ☹.

What will you make?

According to my router my Edison module has no host name. Using the command "hostname –fqd" in the terminal returned the error "Host name lookup failure". Pinging Google resulted in "bad address 'www.google.com'". Searching the internet I found other people having similar problems with Raspberry Pi but none of the proposed solutions helped, either because they didn't seem to do anything or because the commands were not recognized by Yocto Linux. If someone knows how to solve my problem, please let me know. In the meantime I will continue hating Linux.

The lucky users that can connect their browser to the Edison module can try out the Arduino Wi-Fi examples. They compile and should work fine (as long as you set the "ssid" and "pass" strings correctly in the sketches), the Edison behaves like an Arduino with a Wi-Fi shield.

I did not find any Bluetooth Arduino examples. The tutorial I managed to find on the Intel website showing how to activate Bluetooth worked (except that I couldn't get it to connect to my PC), but it gives no clue of how to combine Arduino and Edison Bluetooth so this is for the moment reserved to Linux programmers.

Conclusion

The Intel Edison module is a great product that offers an awful lot of power, but how to unleash it? Without properly working Wi-Fi there isn't much you can do with the module. Without a BoB either. Missing documentation and easy-to-understand code examples aren't h4elpful for the newbie to get started. Edison is probably great for people who have high doses of Linux running through their veins and acutely aware of combining all kinds of packages into super applications. People intrepid about modifying the default Yocto *distro* (yeah, I learned a new word) to allow the packages they need to be added. People comfortable with a text editor like vi.

And then there is that detail that keeps nagging me: Edison is specified up to 40°C only. That is not much. I measured 33°C in my trouser pocket at an ambient temperature of 21°C. Keep your wearable devices cool.

(140439-I)

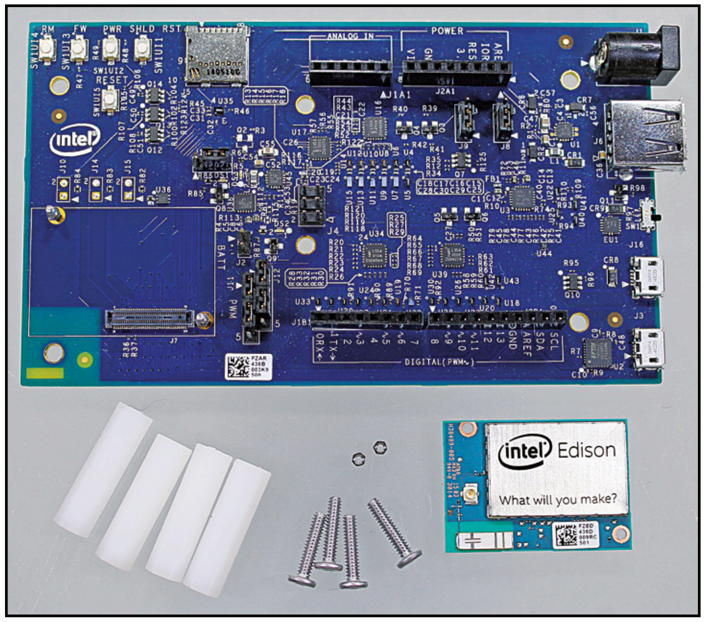


Figure 1. The Edison kit for Arduino unpacked. What will U make with it?

The Intel Edison module is a great product that offers an awful lot of power, but how to unleash it?

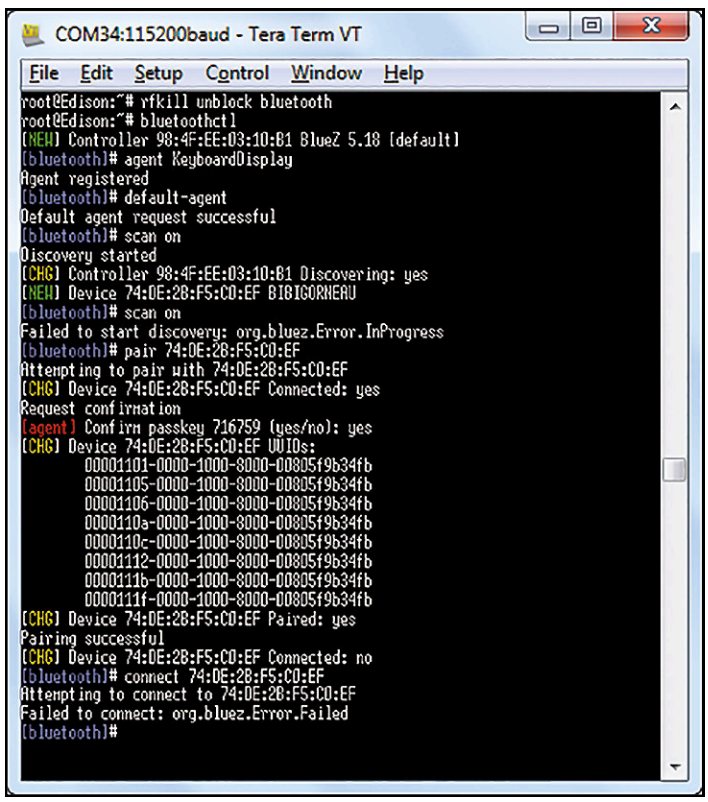


Figure 2. Activating Bluetooth on the Edison. All went well, except for the connection attempt at the end.

FUZEliers Support Young Coders at TNMOC

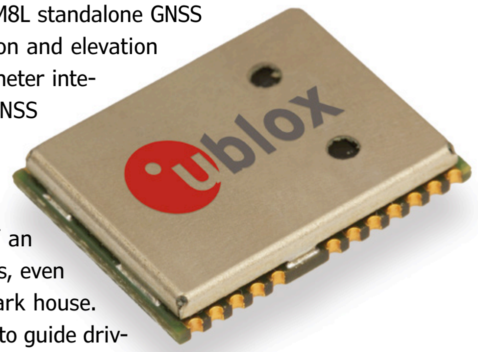
A team of *FUZEliers* sponsored by FUZE Technologies is ready every weekend to welcome young and budding computer programmers at The National Museum of Computing to show them how to code using the FUZE platform to explore the huge potential of computing.

The *FUZEliers* will be introducing young coders to the FUZE, a purpose-built, computer to make teaching and learning programming easy and fun. It is an electronics workstation powered by a Raspberry Pi and programmed using the ever-popular BASIC language. Youngsters will be encouraged to develop their logical thinking in a fun way by programming robots and other devices linked to the FUZE to give them a real sense of the power of computing.

Any young visitor can drop in and be guided by a team of student *FUZEliers*. FUZE, developed by UK-based FUZE Technologies Ltd, has been created by computer enthusiasts and has been designed to help schools meet elements of the National Curriculum.

(140549-IV)   www.tnmoc.org

Ultimate Indoor/outdoor Positioning Module



Alpha Micro Components' new NEO-M8L standalone GNSS module incorporates motion, direction and elevation sensors with a gyro and an accelerometer integrated into its leading u-blox M8 GNSS platform. In addition to accessing the integrated module's gyro and accelerometer data, accident reconstruction systems can provide the location of an accident to facilitate insurance claims, even if a collision occurs in a tunnel or park house. High-end navigation devices are able to guide drivers through tunnels several kilometers long thanks to the unsurpassed accuracy of u-blox' ADR system. Stolen vehicles can be located instantly due to continuous monitoring of sensor data and storage of location in non-volatile memory. The module is able to track all visible GNSS satellites including GPS, GLONASS, BeiDou, QZSS and all SBAS (European's Galileo in future firmware version). Concurrent reception of two GNSS systems is supported. The module can output a position up to 20 times per second.

(140525-II)   www.alphamicro.net/franchises/u-blox/neo-m8l.aspx

Field Analyzer with PIM and Line Sweep Testing

Anritsu Company breaks new ground in field wireless test with the introduction of the MW82119B PIM Master™, which combines a 40-W, battery-operated PIM analyzer with a 2 MHz to 3 GHz cable and antenna analyzer, eliminating the need to carry multiple instruments to measure the RF performance of a cell site. The MW82119B provides tower and maintenance contractors, network installers, and wireless service providers with the first handheld field passive intermodulation (PIM) analyzer with line sweep capability so they can fully certify cell site cable and antenna systems. The MW82119B PIM Master has also achieved an IP54 ingress protection rating, certifying its ability to operate without damage after exposure to blowing dust and water spray.

(140549-II)   www.anritsu.com

PicoTech to Launch 4824 8-Ch PC Scope @ Embedded World 2015 ... RedPitaya Scoops 2014 Frost & Sullivan Award ... Intersil: Industry's Smallest 3A/6A Step/Down Converter ... New AnywhereUSB from Amplicon ... Audio Precision new APx500 Audio Test Software ... Meet & Greet Editor Jens Nickel @ Embedded 2015 ... Silicon Labs to Keynote

EL-EnviroPad-TC Thermocouple Logger

The Lascar EL-ENVIROPAD-TC introduced by Saelig Company Inc. is a conveniently-sized thermocouple-based temperature meter for spot temperature readings with additional built-in data-logging and graphing functionalities. This robust and easy to use handheld device takes and records temperature readings via the supplied K-type thermocouple probe. Control and display is via a 2.8" color touch screen, which indicates current temperature to a resolution of 0.1 degC, maximum and minimum readings, and will produce an immediate graph of the data readings. Operation is user-selectable in Fahrenheit or Centigrade.

The EL-ENVIROPAD-TC's case is IP-66 rated, and a protective rubberized boot is also available. The instrument is compact (4 1/4" x 2 5/8" x 3/4") and the supplied 9.5" K-type thermocouple probe features a coiled lead wire that can stretch over 4 feet from probe tip to connector.

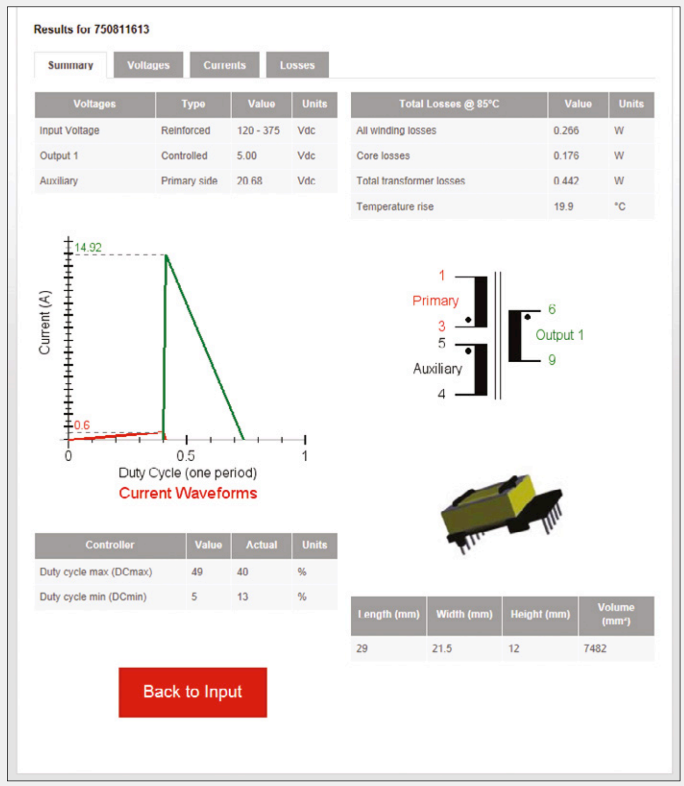
The EL-ENVIROPAD-TC is available now at $169.95.

(140525-IV) www.saelig.com

Design Tool for Discontinuous Mode Flyback Transformers

For discontinuous mode, flyback, SMPS designs, a clever online tool has been launched by Würth Electronics Midcom Inc. Entering as little information as input voltage range, switching frequency, output voltages and current, the Smart Transformer Selector, or STS, will return a list of prequalified, off-the-shelf transformers.

Taking the tedium out of the task, the STS searches Würth Electronics Midcom's database of hundreds of SMPS transformers designed for discontinuous mode, flyback operation. The STS uses power supply parameters and searches the database for parts that will not saturate, will provide the proper output voltages, and will not over stress your switch. The search results are returned in a table that lists many of the typical transformer parameters, such as inductance, turns ratio and saturation current along, with mechanical and safety parameters for the parts. Full specifications of the part are available for immediate download in pdf format right from the table. The STS finds all suitable transformers, even those that were not designed specifically with the given application in mind. The table itself can be sorted in various fashions, giving the ability to hone in on those designs that best suit the specified application. Selecting any part from the list will populate an analysis tab directly below the chart which summarizes the part and its performance in the application. It shows everything from current wave forms, voltage levels and power losses to mechanical dimensions, a schematic for the application and a 3D image showing what the part looks like. To get more detailed information one of three tabs available reveals the details of the currents, voltages and losses. As an added benefit to customers, the STS lists all parts which are available off the shelf. Customers eliminate a standard wait time of up to eight weeks for transformers.



There are no minimum order quantities on the products listed in the Smart Transformer Selector. Most of the parts found in the STS are stocked at Digikey or Mouser, and all are stocked directly at Wurth Electronics Midcom Inc.

(140525-I) www.we-online.com/sts

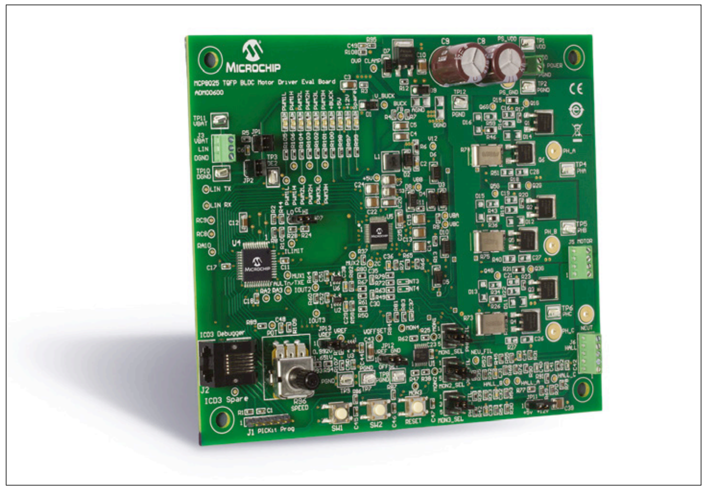
Embedded World ... Meet Freescale Experts @ Embedded World 2015 ... RS Components to Showcase DesignSpark 7.0 @ Embedded World 2015 ... Mr. Battery's Wintry Advice from Interstate's Batteries ... Veroshield 19-inch Rack for the BBC ... Successful PM-8QAM across Submarine Cable ... element14 Launches Atmel SAMA5D4 Explained Board ...

High Performance Vector Network Analyzers

Anritsu's VectorStar series offers extreme high performance suitable for use in leading-edge research establishments, as well as in the development of microwave devices used in communications, aerospace, military, security and industrial equipment. The single-sweep measurement capability of VectorStar VNAs offers a span of 70 kHz-145 GHz. High-performance pulse analysis is supported at a resolution of as little as 2.5 ns.
In a joint demonstration with RWTH Aachen, Anritsu recently showed how ShockLine VNAs can be used to perform near-field antenna measurements and measurements of material properties.

(140525-VI) www.anritsu.com

3-Phase BLDC Motor Gate Drivers ith LIN TX/RX

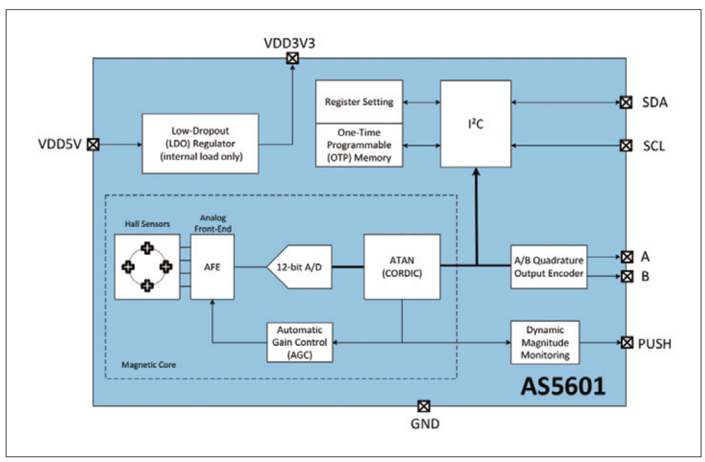


Microchip Technology Inc.'s MCP8025 and MCP8026 (MCP8025/6) are 3-phase Brushless DC (BLDC) motor gate drivers with integrated power module, LIN transceiver and Sleep mode. These devices can power a broad range of Microchip's dsPIC® Digital Signal Controllers (DSCs) and PIC® microcontrollers (MCUs), complementing their control algorithms by driving MOSFETs, sensing current, preventing short circuits, outputting zero crosses, controlling dead time and blanking time, and monitoring for fault conditions such as over/under-voltage, over-temperature and other thermal warnings. The MCP8025 driver is supported by Microchip's MCP8025 TQFP BLDC Motor Driver Evaluation Board (Part # ADM00600), which is available now for $149.99.

(140525-V) www.microchip.com/get/D3EG

Contactless Sensor Replacement for Rotary Encoder

In devices which use rotary knobs, ams' new AS5601 and its paired magnet may replace a rotary encoder without any change to the host microcontroller or its application software.
The AS5601 is based on magnetic Hall position-sensing technology, which is already widely used in the automotive, industrial, medical and consumer markets. Since the AS5601 performs contactless rotary position measurement, it suffers none of the reliability problems with which conventional rotary encoders are plagued. They are notoriously prone to interference and premature failure because of the effects of mechanical wear and contamination by dust, grease, dirt and humidity. The robust differential sensing circuit in the AS5601 also rejects interference from external magnetic stray fields.
The AS5601's operation, including its zero position, provides for easy configuration, since its register settings are accessed via an I²C interface and are saved in on-chip OTP memory.
The device's quadrature (A/B) output provides between 8 and 2048 positions, hence the AS5601 may be used by off-the-shelf rotary knob or encoder manufacturers, in multiple end products with different output requirements. Users of the AS5601 also have the choice of a 12-bit digital output, suitable for designs that are not directly replacing a conventional rotary encoder.



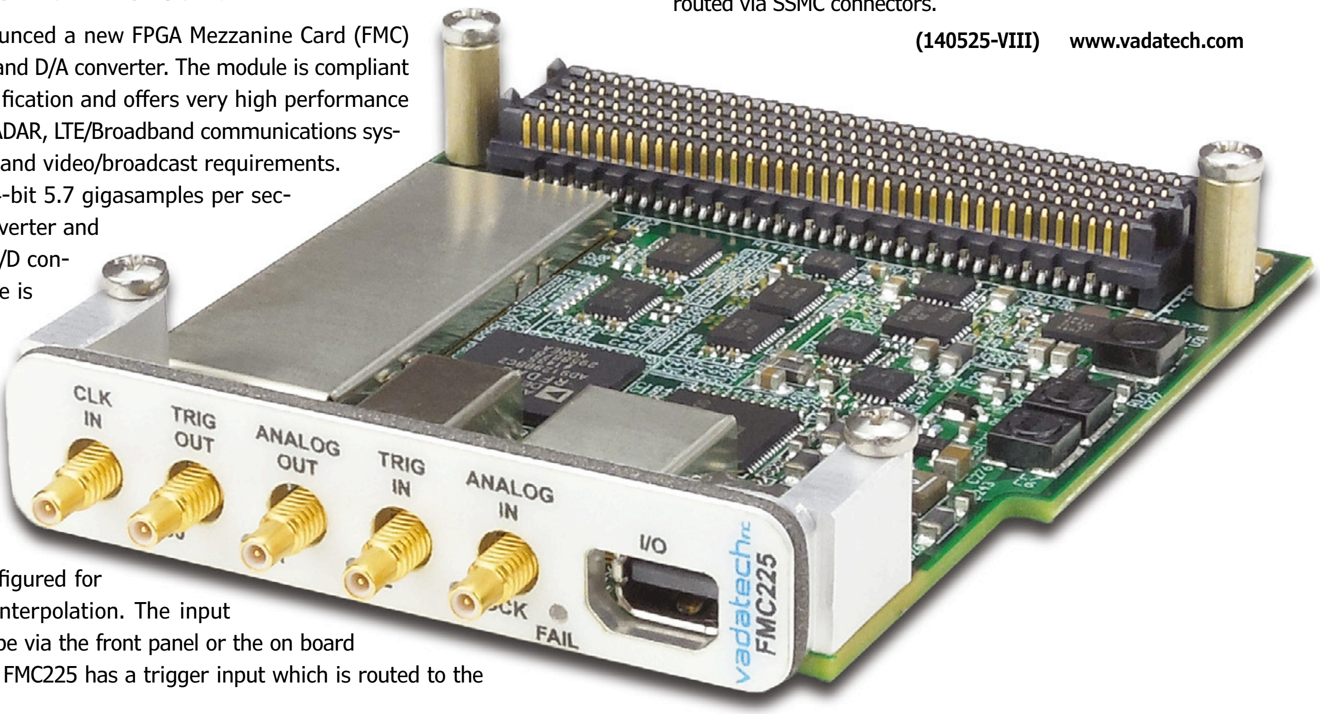
As well as measuring angular displacement, the AS5601 can detect button presses. Algorithms implemented in the AS5601 reliably detect a sudden significant reduction in the air gap between the IC and its paired magnet, generating a PUSH output signal. This push-button function is immune to variations in magnetic field strength due to temperature variations or ageing.
A demonstration board for the contactless rotary position sensor AS5601 is available online from ams.

(140525-III) www.ams.com/Rotary-Position-Sensor/AS5601

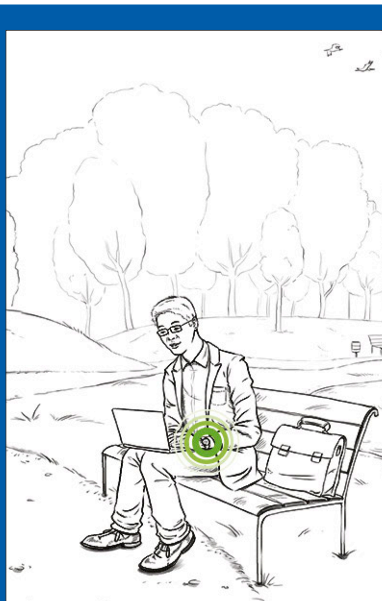
Ultra-Fast A/D /DA Converters on FPGA Mezzanine Card

VadaTech has announced a new FPGA Mezzanine Card (FMC) that is both an A/D and D/A converter. The module is compliant to the VITA 57 specification and offers very high performance targeted towards RADAR, LTE/Broadband communications systems, ATE, Physics, and video/broadcast requirements.
The FMC525 is a 14-bit 5.7 gigasamples per second (GSPS) D/A converter and a 12-bit 4.0 GSPS A/D converter. The DAC core is based on a quad-switch architecture that enables dual-edge clocking operation, effectively increasing the DAC update rate to 5.7 GSPS when configured for Mix-Mode™ or 2x interpolation. The input sampling clock can be via the front panel or the on board wide-band PLL. The FMC225 has a trigger input which is routed to the FMC connector. The analog input/output, clock input and trigger inputs are routed via SSMC connectors.

(140525-VIII) www.vadatech.com



Environmental Sensing for a Smarter Life



Like all Sensirion sensors, the new SHTW1 humidity and temperature sensor type SHTW1 is based on CMOSens® Technology, which integrates the sensor component and the evaluation circuit on a single semiconductor chip. The SHTW1's package is no larger than the CMOSens® Chip itself, yet the device has the same functionality as conventional digital humidity and temperature sensors. So when it comes to usability, space is no longer a concern.
The operating voltage of 1.8 V and the low power consumption of just 2 µW at 1 measurement per second provide an optimal basis for using the sensor in small, mobile devices. As the specifications demonstrate, performance was not sacrificed to achieve the tiny size: it measures relative humidity across a range of 0% to 100% RH, with a typical accuracy of +/-3% RH. The temperature measurement ranges from –30 °C to 100 °C with a typical accuracy of ±0.3 °C. The completely calibrated sensor features an I²C interface and is reflow solderable.

(140525-VII) www.smart.sensirion.com

PCB File Formats

compete to serve next-generation designs

By **Robert Huxel,** Technical Marketing Manager, Altium Europe GmbH

For an industry that regards itself as fast-moving and always pursuing the cutting edge of technology, the key step of preparing one of electronic systems' most fundamental components — the printed circuit board — for manufacture, is remarkably conservative. But it is an area in which there is currently an energetic discussion about the way forward.

The main point is that of providing data out of the design process to enable a fully-correct printed board to be manufactured. The production file exchange format evolved over time and perhaps unsurprisingly, there is no completely universal or "standard" specification of what constitutes a complete package of information handed to fabrication. There has been, and still is, a considerable amount of manual intervention in the process, with the fabricator routinely carrying out their own verification of the correctness and manufacturability of what they have been sent. A smooth and consistent workflow between design and manufacturing frequently evolves through personal contact, and changing contractor can be disruptive.

A growing number of voices declare that this paradigm is no longer sufficient, that the industry needs a new and comprehensive standard to define a PCB in an error-free and unambiguous manner that can be used to drive an automated manufacturing flow for the ever-more complex boards. As has so often been the case in the electronics industry, where there is a perceived need for a new standard, a number of competing candidates have emerged. In this case, the number (depending on how you categorize them and trace their antecedents) is three. As is also so often the case in the industry, there is a philosophical split between those who would build a new standard from the ground-up and those whose inclination is to continue the evolution of that which already works, adding features and fixing its shortcomings as required.

The Candidates

The faction that says, "If it (mostly) works then don't make wholesale changes — extend what we have, to accommodate the problem areas," — a doctrine of "minimum necessary change" — centers on the Gerber format. The format itself is now in the custody of Ucamco.

The most recent revision to the established order adds features to the widely-used extended-Gerber format to create "Gerber X2" (**Figure 1**). A key element of this progression is that the amount of information embedded in the basic geometry files is enhanced because the features they draw, now carry attributes. Those attributes can specify what type of object is depicted, and from that, aspects such as connectivity can be inferred automatically. Not the least claim made for this upgrade — playing to the sector's innate conservatism — is that it is backward-compat-

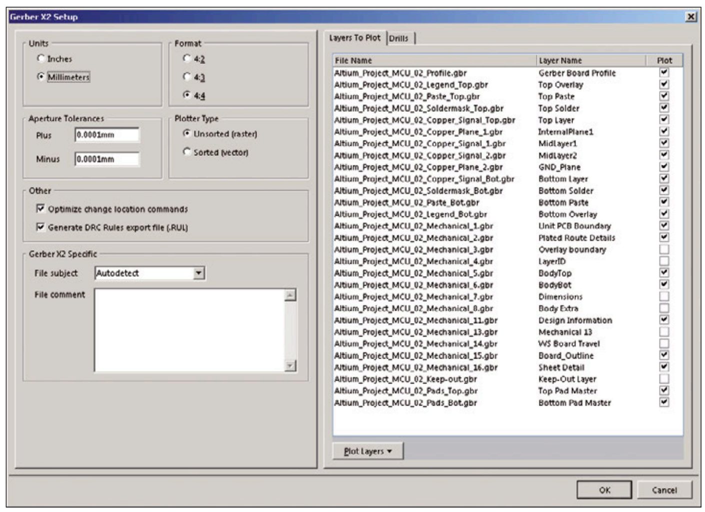


Figure 1. Gerber X2 Configuration Dialog in Altium Designer.

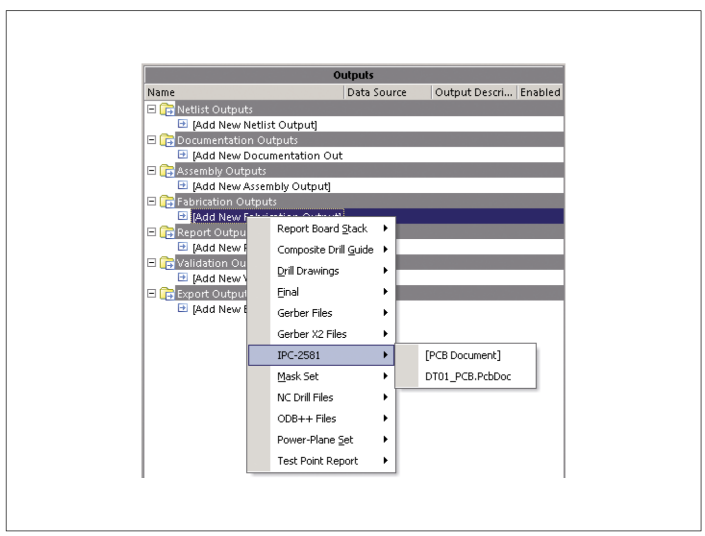


Figure 2. Easily make your choice between IPC-2581 and Gerber X2.

ible with widespread practice today; if you don't want to use the added information, your software simply ignores the added data and your system can continue as before.

For every engineer who champions progress by continuous refinement, there is one who advocates a clean-sheet approach and this sector is no exception. In general, the opposing camps propose a data-centric approach, deriving all needed information from a single data set, rather than the geometry-centric view of Gerber.

With its own history, one candidate is ODB++. The double-plus designation provides a clue to the fact that this format has gone through its own evolutionary process. The format was created by Valor, a company subsequently acquired by Mentor Graphics, and the format is now maintained, and promoted, by Mentor.

In the third camp are enthusiasts for the IPC-2581 format (**Figure 2**), a single description that is based on XML (eXtended Markup Language). As with other domains in which XML is used, its proponents observe that it was designed as a carrier of data, and as a consistent source format

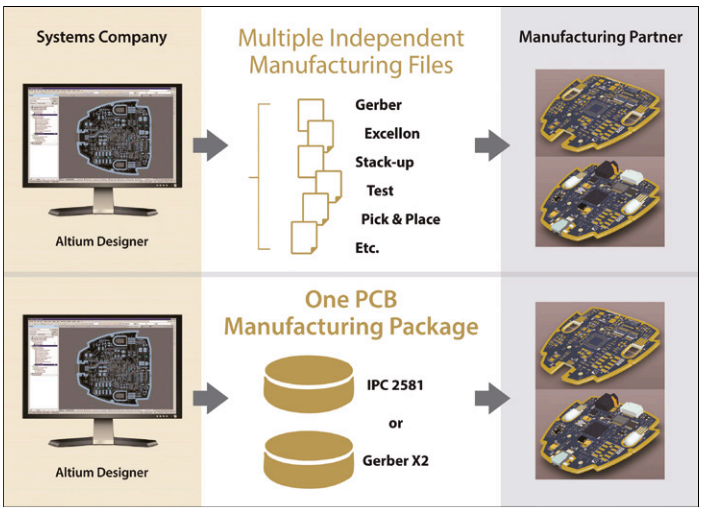


Figure 3. The PCB manufacturing file becomes a "one-click" event in the design process.

from which derivative data sets (in this case, for example, images and CNC files) can be generated. It is therefore an appropriate foundation on which to build a new standard for the PCB CAD/CAM interface. IPC-2581 is in the custody of a dedicated consortium, with Cadence Design Systems being a prime mover.

The Challenge of Adoption

Users are often wary of adopting formats that are seen as being proprietary, and of becoming embroiled in EDA-industry "standards wars", and scrutinize the role of the lead EDA companies in each standard. In the case of ODB++, Mentor has consistently positioned the standard as "open", with all aspects able to be supported by users — and competitors — from across the sector. Cadence emphasizes its position as just one of the lead players in the independent consortium.

All of these approaches are structured to encompass the parameters that characterize high-performance PCBs for today's and tomorrow's designs — for example, rigid/flex sections within the same board; and identification of tracks and routes as impedance-matched.

What is the likely impact in moving to an alternative strategy? Clearly, both designer and manufacturer must adopt the same standards and assimilate the costs of acquiring the necessary tools; are existing products in the marketplace sufficiently mature that "plug and play" is assured, or is a learning process on both sides inevitable?

Superficially, when equipped with a PCB design package that can output a given next-generation format, all the OEM will need to do is ensure that his board fabricator can accept that format, and begin using it to transfer board specifications to manufacture. Such changes in working practice are rarely so simple. Aside from any issues around the format itself, and interpretation of it by all parties, changes to the environment in which the PCB design is created also have a bearing.

Structures change

Today, the circuit designer can be using a powerful and comprehensive software package that takes his project all the way from architecture and schematic capture through to 3D physical design of the product, taking in the PCB design in all its aspects along the way. Indeed, with a high-speed, dense, and signal-integrity-critical design it is likely essential that the circuit designer controls the board geometry as it develops. In that flow, export of the PCB manufacturing data becomes a "one-click" stage in the process (**Figure 3**).

It follows that the level of standardization must be high, and confidence in the downstream flow to interpret the data package and execute on it, automatically, equally high. The same pressures on time-to-market that drive progress to more advanced standards mean that once the button is pressed to download the PCB data package, the engineer's attention will be directed elsewhere. The fabricator's "comfort zone" of a dedicated PCB engineer, immediately available to resolve issues, is likely to diminish. For such reasons it seems unlikely that the PCB design/manufacture user base will abandon its conservatism in a hurry.

It seems unlikely, too, that any one of the competing formats will become totally dominant. It may be that, as a purely format issue (neither data-centric nor shape-centric methodology should impose any restrictions on the design process or, ultimately, the fabrication flow) this comes to be resolved in the way that so many format "wars" have been. That is, a long-drawn-out popularity contest in which design-side tool vendors will have to support multiple variants, and a "winner" will emerge slowly — if ever.

(140525)

Welcome to Elektor Labs

Elektor Labs is the place where projects large, small, analog, digital, new and old skool are sketched, built, discussed, debugged and fine-tuned for replication by you.

Our History

The origins of Elektor Labs go back to the early 1970s when soldering and writing was a one-man, single-desk job. Ove the years Labs staff have not only witnessed the arrival of the transistor, the IC, the microcontroller, and the SMD, but actually jumped on these parts as soon as they were out of the professional-only woods. Once special branches, Audio Labs, Micro Labs, PCB Labs, and Mechanical have converged back again into a single activity.

Our offer: Become Famous

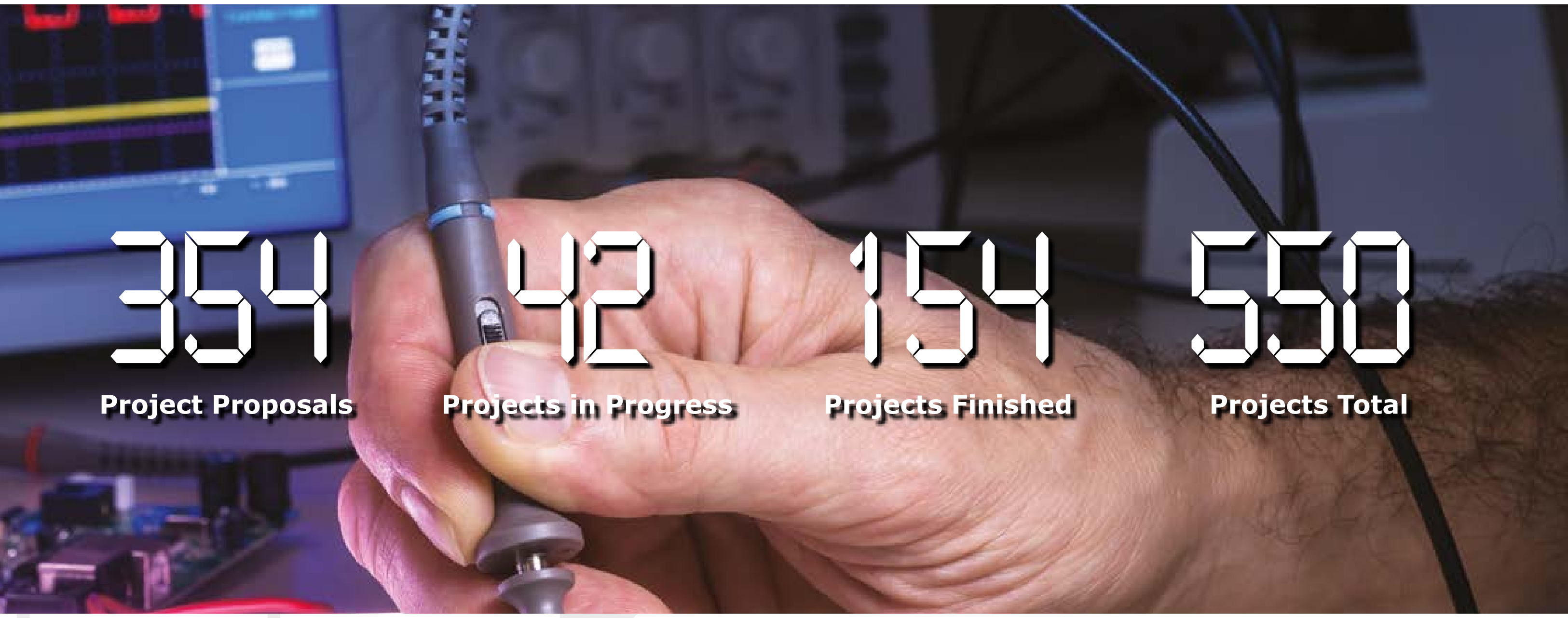
Most engineers and budding authors we come across are just too modest. If they do not see the attraction and beauty of a design idea scribbled on a coaster and worked out later at home, others may, and should. Let Elektor Labs help you hone your design to perfection, let the editors assist with text & graphics, and reap the rewards by seeing your name in print in Elektor magazine's celebrated LABS section. Sure, we are happy to negotiate payment, but the actual remuneration will be fame & glory in electronics land, and your name added to the long list of extremely successful e-authors. Our get-u-famous formula also applies to book authors, bloggers and video makers. Students and youngsters: being in publication is a current boost like no other in getting a job!

Our Facilities

We are sumptuously housed in three spacious rooms at Elektor House where we try unsuccessfully to keep our solder jobs and prototypes off our computer desks. We have water, 218 volts electricity and coffee nearby. PCB milling, prototype assembly, SMD reworking, audio testing, pizza cooking, and mechanical work are deferred to converted cellars in the basement.

Our .Labs Website

Use this online, highly bidirectional port to proclaim your project and follow the activity of others. Read about other cool projects, make contributions, and interact with fellow enthusiasts. Noticeably buzzed projects qualify for post-engineering in the real Elektor Labs and eventually, glorious publication in Elektor magazine. Soap-box your project on our website and hopefully see it in print.



Our Webinars

The more talkative of our Lab engineers do not stop at testing prototypes, they happily share technology related problems, insights, get-u-going information, and design skills on live camera at Elektot.tv. Labs' webinars are free to attend and extremely interactive. They are announced in Elektor.POST, and webcast live from Elektor House in Holland. Do plug in!

Our Products

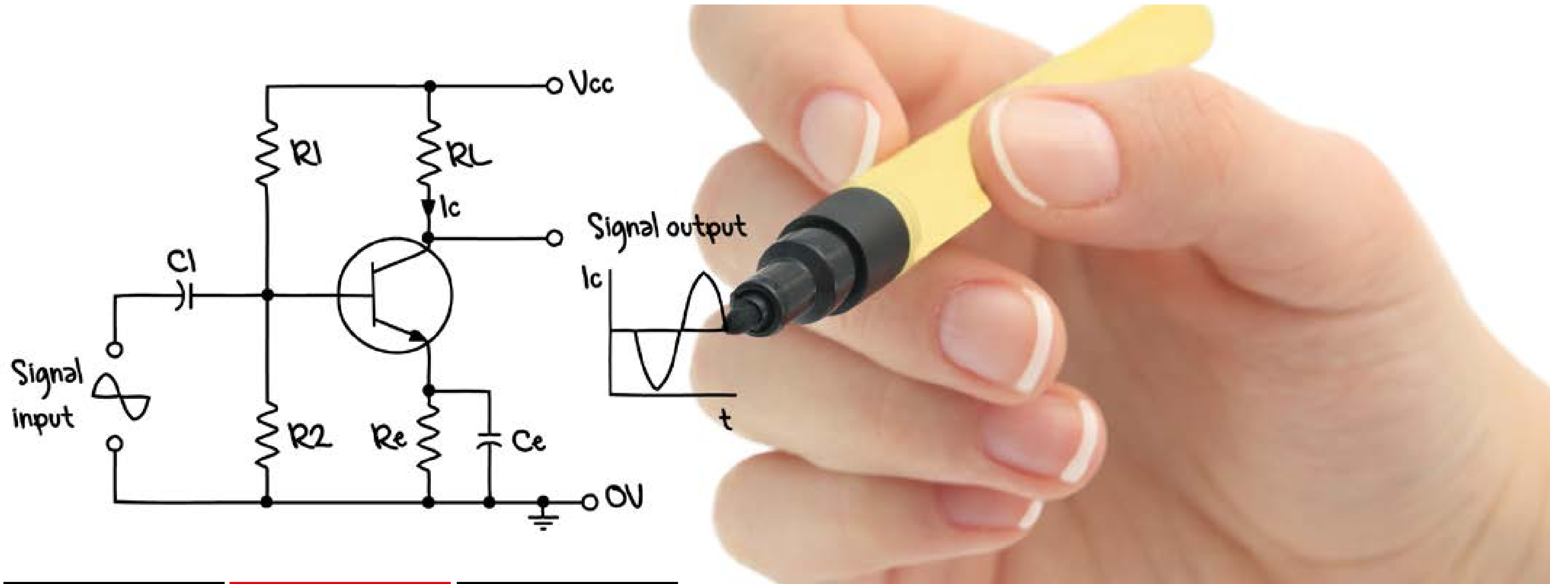
Our products are visible in Elektor magazine, as well as on the Elektor.Labs and Elektor Store websites. The range includes text write-ups for editors, prototype photography, PCBs including SMD-prestuffed, PCB files, project software, programmed devices, semi-kits, tools, modules, videos, and service desk information.

Our Experts and Designers

Besides experienced support staff and BSc, MSc qualified engineers with a total working life in electronics of about 200 years, the Lab has access to Elektor's vast network of experts for consultation, critical advice, and assistance with specialized assignments.

Our Standards

All projects and products going through the Labs pipeline are produced to high engineering standards. In practice, prototypes of projects labeled LABS in the magazine must be demonstrated to work to specification on certified, calibrated test equipment available locally. BOMs and schematics must match perfectly. Kits are sampled for completeness. We are ROHS compatible, lead free and comply with electrical safety standards applicable in our location. If engineering errors are found these will be put up for public notification.



LEARN DESIGN SHARE

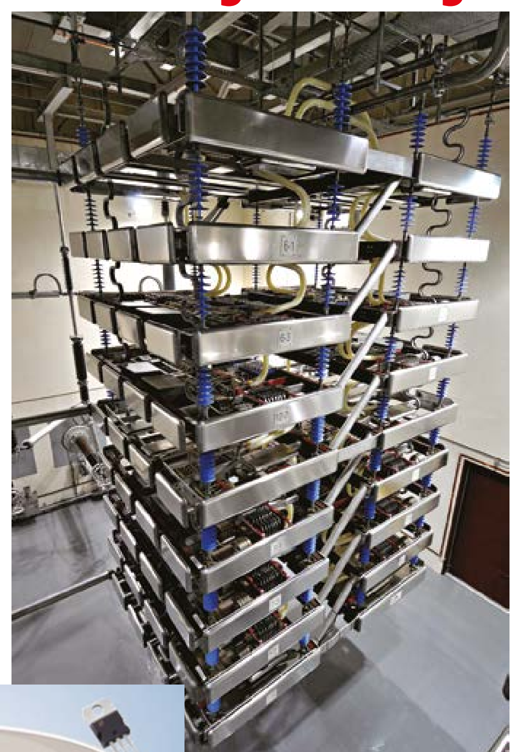
Inspired by design

The following pages have been grouped under the header "Design". Here you will find projects that can be built and understood without too much head scratching. It is about design in the sense of a set of worked-out plans featuring a description including building and testing instructions, a circuit diagram, a list of components, a drawing for a printed circuit board and, often, some software. The designs can also be used as a source of inspiration.

When you look up the word design in the dictionary you will find several definitions that all have wilfulness and purposefulness in common. Wilful has a negative connotation, purposeful doesn't. Design is like a polarized capacitor charged with ideas. It is a special kind of capacitor though, because it only keeps the good ideas, or, to be more precise, it stores the ideas that the designer considered worth keeping and so designs can be good or bad. Designs can also be ugly.

This reminds me of those spaghetti Westerns that were directed and produced by Italians. Design is a way of life for Italians. The popular rapid-prototyping platform Arduino was designed in Italy and meant for artists and designers. It is a design inspired by design. From an electronics and software point of view the platform is not a particularly good design, but it is a very successful one. Arduino is the Good, the Bad and the Ugly of electronics engineering that inspired thousands of people to have a go at spaghetti electronics. Ooh, now I feel inspired too; I'm gonna make an electronic harmonica.

This is what I call a 'Design Challenge'



This is a thyristor, or, to be more exact, a cluster of thyristor valves. Constructions like this are used to convert AC to DC and vice versa in so-called High Voltage Direct Current (HVDC) systems. Besides having efficiency advantages compared to HVAC systems, HVDC technology allows the connection of AC networks that work on different frequencies.

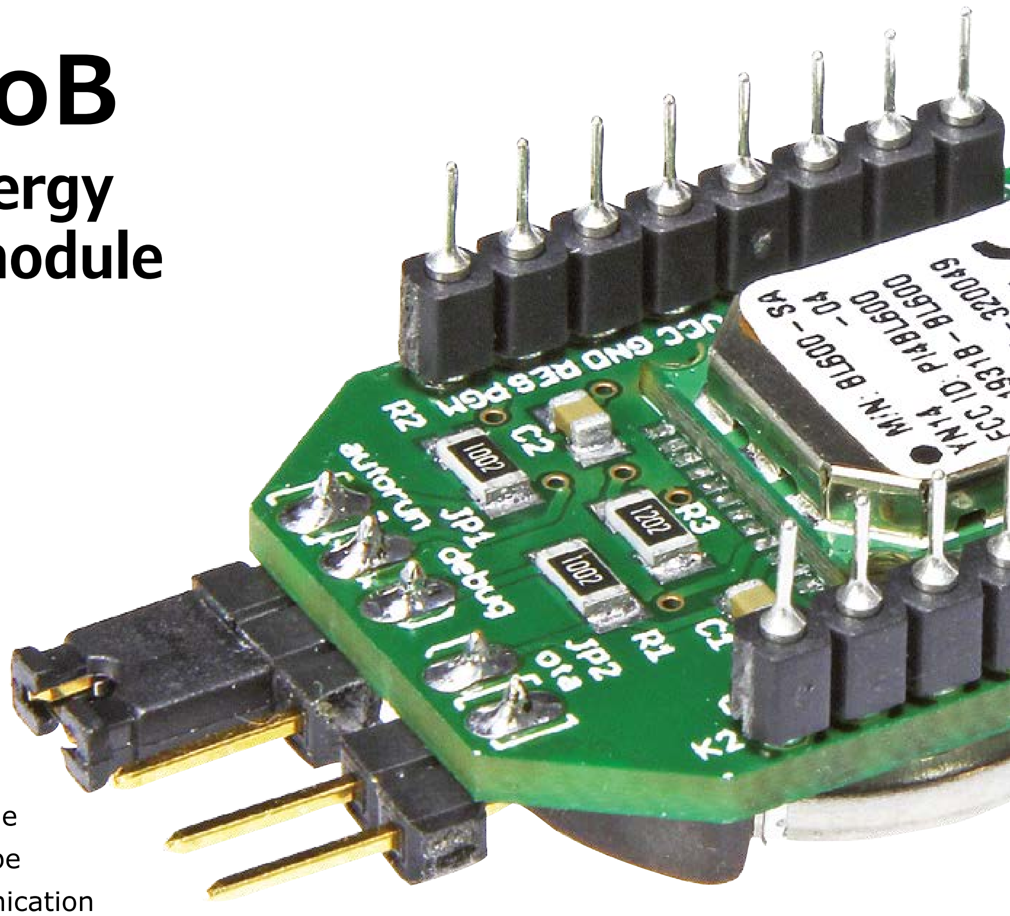
The photo shows an H400 valve built by Alstom Grid as part of a 400 MW power grid. It measures about 13 meters in height. Inside are several – typically twelve per valve tower – 150 mm (6") diameter thyristors wired in series that together can handle up to 400 kVDC. Snubber and grading capacitors and liquid-cooled resistors are also included inside the valves to improve the performance. Two of these clusters are connected in series for an 800 kV HVDC system.

The thyristor shown is a T1503NH from Inifineon, specified up to 8,000 V_{RRM} with an average on-state current of more than 2,000 A. The device can handle peak currents up to 57,000 A.

More on HVDC: www.sari-energy.org/PageFiles/What_We_Do/activities/HVDC_Training/Presentations/Day_7/ALSTOM_HVDC_for_Beginners_and_Beyond.pdf

BL600-eBoB

Bluetooth Low Energy communication module
Part 1



Wireless communication on a plate

By Jennifer Aubinais (France)
elektor@aubinais.net

For the network of connected objects (or Internet of things – IoT) to be able to thrive, certain conditions need to be met — in particular, wireless communication and low power consumption by the circuits used to connect these objects. Without long battery life, we'll soon lose interest. So the breakout board for the ultra-low consumption radio communication module presented in this article is going to be an ideal accessory for exploring IoT.

In the January & February 2015 edition, I presented a wireless outdoor thermometer [1], fitted with the Laird Technologies BL600 module (**Figure 1**). In association with iOS and Android applications, this project lets you read a remotely-measured temperature on your smartphone in Bluetooth Low Energy. In order to get to know this module, I strongly recommend you to read up that article; it shows just what you can get out of it.

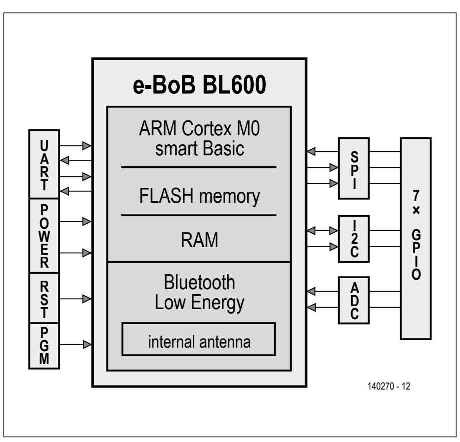


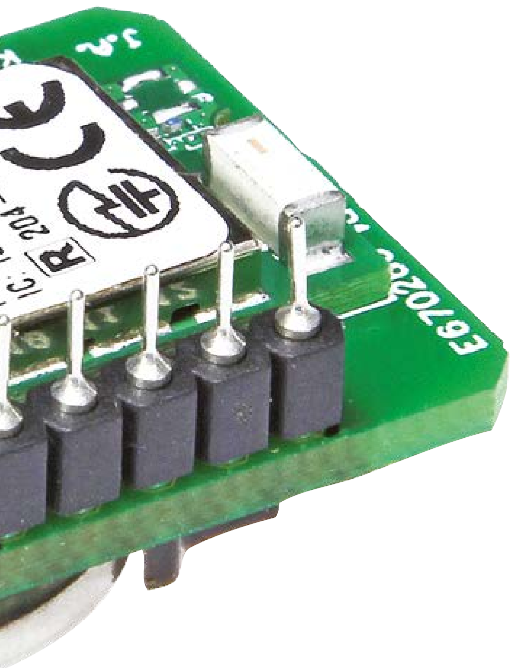
Figure 1. The BL600 Bluetooth Low Energy module is an ideal accessory for achieving wireless communication with connected objects.

The BL600 takes wireless communication for connected objects into a new era, thanks in particular to its low consumption and high degree of miniaturization; however, the latter comes with its own problems for anyone hoping to solder this component by hand. Aware of this drawback, the manufacturer offers a simple, effective trick for positioning the module with its miniaturized connections on a PCB to within 0.1 mm. This is also described in last month's article, but to spare our readers the pitfalls of this tricky operation, the PCB for this thermometer is available *with the module already fitted* from the e-shop [2]. It's jolly handy! On the same subject, I also recommend the video elektor.labs have posted on YouTube [3]; it shows how easy it is to establish BT communication between — in this case — a remote thermometer and a smartphone (Android in the video, but it's just as simple under iOS). The same principle can be used for countless other applications.

Breakout board

Given the universal interest of the BL600 module and of the other BLE applications envisaged by elektor.labs, we thought it would be useful to now offer you a breakout board (BoB). Despite the compact size of the Laird module (19 mm × 12.5 mm × 3 mm), this new e-BoB from Elektor will let you access the main signals on the BL600 module, even if you solder by hand.

The circuit (**Figure 2**) shows the two connectors K1 and K2 and the two jumpers JP1 and JP2 found around the edge of the board (**Figure 3**). As a choice had to be made out of all the module's pins, due to lack of space on the breakout board for connecting them all, certain of them have been left off (2, 6–8, 18–21, 24–26,



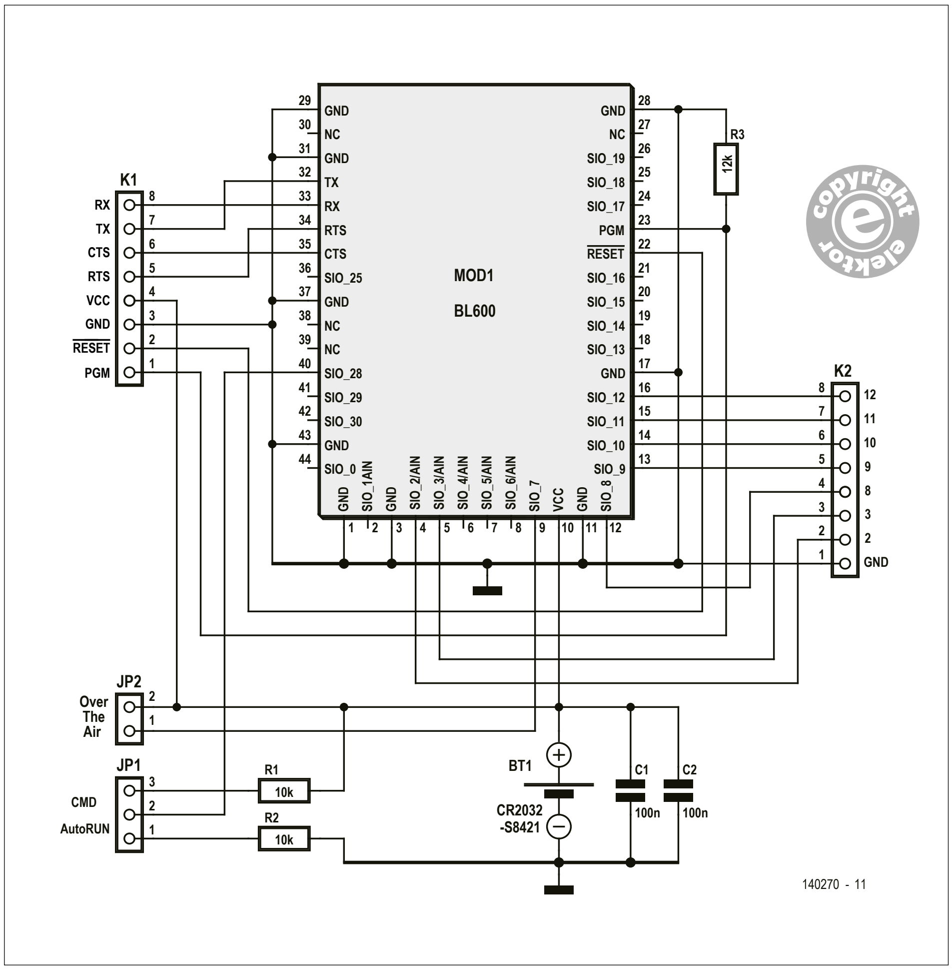


Figure 2. As is only fitting for a breakout board, the circuit of the e-BoB includes only very few components: the BL600 module itself and a few bias resistors and decoupling capacitors. Ultimately, the key elements are the 0.1″ pitch pin headers giving access to the module's main signals

36, 41, 42, 44) in favor the ADC, I²C, and SPI outputs, which are accessible (we'll come back to this in later articles devoted to the module on its BoB).

BoB connections

As is only fitting for a breakout board, the module presides between two rows of 0.1″ pitch pins and two jumpers. The MOD1 signals are grouped as follows:

- The serial port (K1) is used for loading the program into the BL600. It can also be used as a port for dialog between the module and a microcontroller. However, the BL600 has enough inputs/outputs, and its programming language SmartBASIC is powerful enough to allow the module to operate without the help of a microcontroller. Don't miss the next issues of the magazine, where you'll discover this language that certainly deserves its name!

smartBASIC

- Power pins (3.3 V) (K1). As the BL600 module only draws 5 µA (!) in stand-by, it can be powered by a

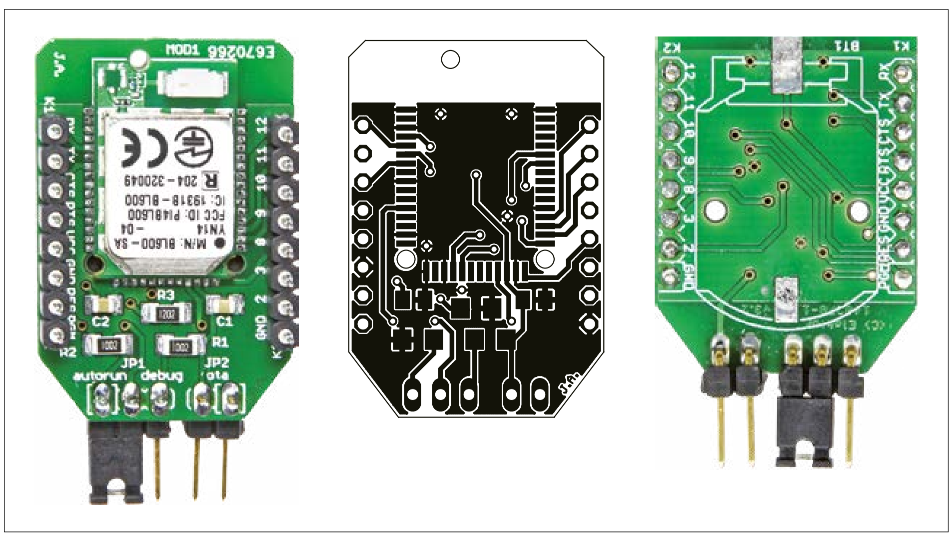
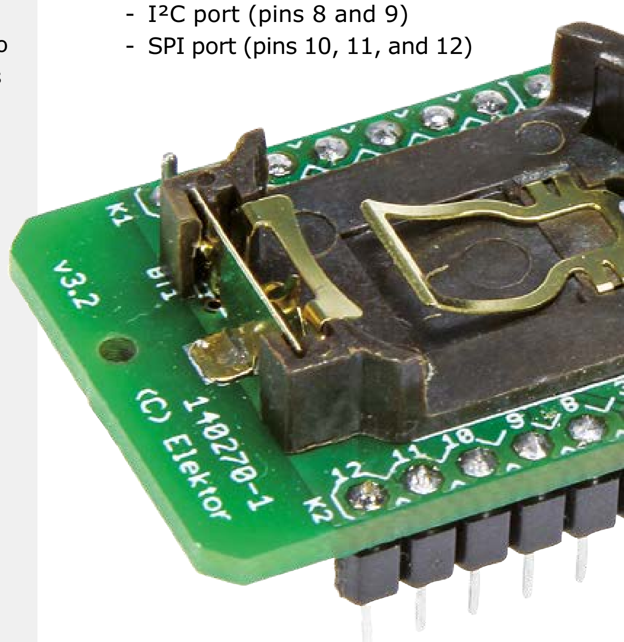


Figure 3. The module is available assembled and ready to use from the Elektor e-shop. K1 and K2 are not fitted but are supplied as loose parts. Depending on whether or not you will be using a battery holder and a button cell (optional, not supplied), you will have to fit K1 and K2 on one side or the other of the board.

CR2032 button cell (BT1), the holder for which is fitted underneath the board; pin-headers K1 and K2 will be fitted to the other side if the battery-holder is fitted.

- The RESET line must be connected to a mini push-button.
- The PGM pin (marked “Not Connected” in the Laird Technologies documentation [4]) can be used for any updates to the modules firmware (for which a J-LINK programmer is required).
- 7 logic inputs/outputs (K2), which can also be used as follows:
  - 2 × 10-bit analog inputs (pins 2 and 3)
  - I^{2}C port (pins 8 and 9)
  - SPI port (pins 10, 11, and 12)



- Jumper JP1 `autorun / cmd` (previously referred to by the author as `autorun / debug`) lets you select between the following modes:
  - AT commands (e.g. `AT&F 1` which performs a complete initialization of the module)
  - `autorun` which, at startup or following manual initialization (RESET), allows automatic execution of a program named `$autorun$`.............
- The `ota` (Over The Air) jumper (JP2) allows an already-compiled program to be downloaded *via a radio link* using a Laird Technologies application. We’ll discuss this further below.

Apart from MOD1, this e-BoB uses just three bias resistors and two decoupling capacitors.

Bluetooth Low Energy, Bluetooth Smart

Bluetooth is a standard for bidirectional radio (UHF) communication over short distances (10 m), principally between portable devices (computers, phones, etc.) and their peripherals: keyboards, mice, earpieces, headphones, etc. In Bluetooth version 4.0, known as BLE for Bluetooth Low Energy, the current consumption is much lower than for the previous 1.0 and 2.0 standards. The prevalence of BLE in the new generations of smartphones is encouraging a proliferation of new connected objects: watches, fitness and health accessories, remote controls, toys, home automation, alarms, etc. and the application of Bluetooth communication in new areas.

Bluetooth Low Energy is not trying to compete with its predecessors on speed: its data rate is 0.3 Mbps, as against 1 Mbps for Bluetooth 1.0. Its objective is low consumption when quiescent (5 µA) and during transmission (10 mA). The economical power reserve management facility for BLE modules allows them to be powered from AAA cells and even button cells (e.g. CR2032).
There are different functional profiles for BLE, particularly for medical applications, e.g. temperature, blood pressure (BL), heart-rate (HRP), etc. To do this, all BLE devices adopt a profile of generic attributes (GATT), which simplifies programming applications around consistent notions:

- ***Client***: a device capable of transmitting GATT requests and commands and receiving responses (e.g. a smartphone)
- ***Server***: a device that receives GATT commands and requests and sends responses (temperature sensor)
- ***Peripheral***: a peripheral can signal the presence of other devices.
- ***Central***: only a central can send a connection request and establish communication.

The notion of ***Service*** combines several characteristics specific to a function; e.g. the Health Thermometer service gives the characteristics of a temperature value as well as the interval between two successive measurements. The notion of ***characteristic*** is a value exchanged between client and server, e.g. the battery voltage. The ***descriptor*** gives information about a characteristic, e.g. the measurement units (degree Celsius). We shall see that all this makes programming easier.
The other services from the BLE protocol that the BL600 module knows are: **BPM** (blood pressure), **HRM** (heart rate), **HTM** (body temperature), ***Proximity***, ***Batch*** (*Send file*), ***Serial*** (UART or VSP interface) and **OTA** (Over The Air). We are using the last two here in our application example (see “Let’s connect up the e-BoB” paragraph).
BLE uses the 2.4 GHz band and works with iPhone 4S and iOS 5, Android 4.3 and Windows Phone 8, but does not communicate with Bluetooth 2.0. It has only 37 channels (against 79 for traditional Bluetooth) and contents itself with exploring three of them (against 32), noticeably speeding up connection. Modules from certain manufacturers combine both these technologies; Bluetooth Smart Ready indicates compatibility with both modes, Bluetooth Smart with the Bluetooth Low Energy mode only (**Table 1).**

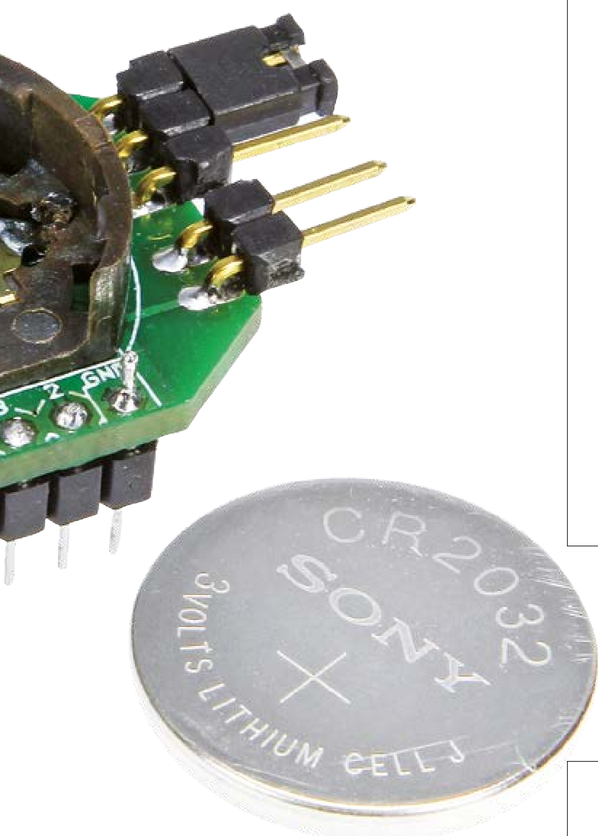
	version	
	2.0	**4.0**
Bluetooth	×	
Bluetooth SMART		×
Bluetooth SMART READY	×	×

To sum up, the advantages of BLE are:

- reduced consumption (battery life in months or even years)
- reduced size and price
- compatibility with recent smartphones
- easy programming

If you decide to build it yourself using the PCB design offered by Elektor, you'll find that the three holes around the edge of the BL600 let you temporary fit 1.6 mm screws that will wedge the module accurately for soldering (in a reflow oven). The procedure is described in my article on the wireless remote thermometer [1].

I also recommend reading this article because it gives an idea of just how easy it is to use the BL600 module, thanks to the SmartBASIC programming language offered by Laird technologies. Only certain aspects are



covered there; in the following articles on the application of the BL600 on its e-BoB, we'll be coming back to take a closer look at this language. While you're waiting, if you're curious, try studying the source code for the examples given in the manufacturer's documentation.

Let's connect up the e-BoB

To conclude, I'm suggesting an initial example of communication between your Android phone and a connected object using our e-BOB. Yes, but which object? A connected watch, perhaps? It's the fashion, of course — but to sim-

Bluetooth 4.0
the wireless detector revolution

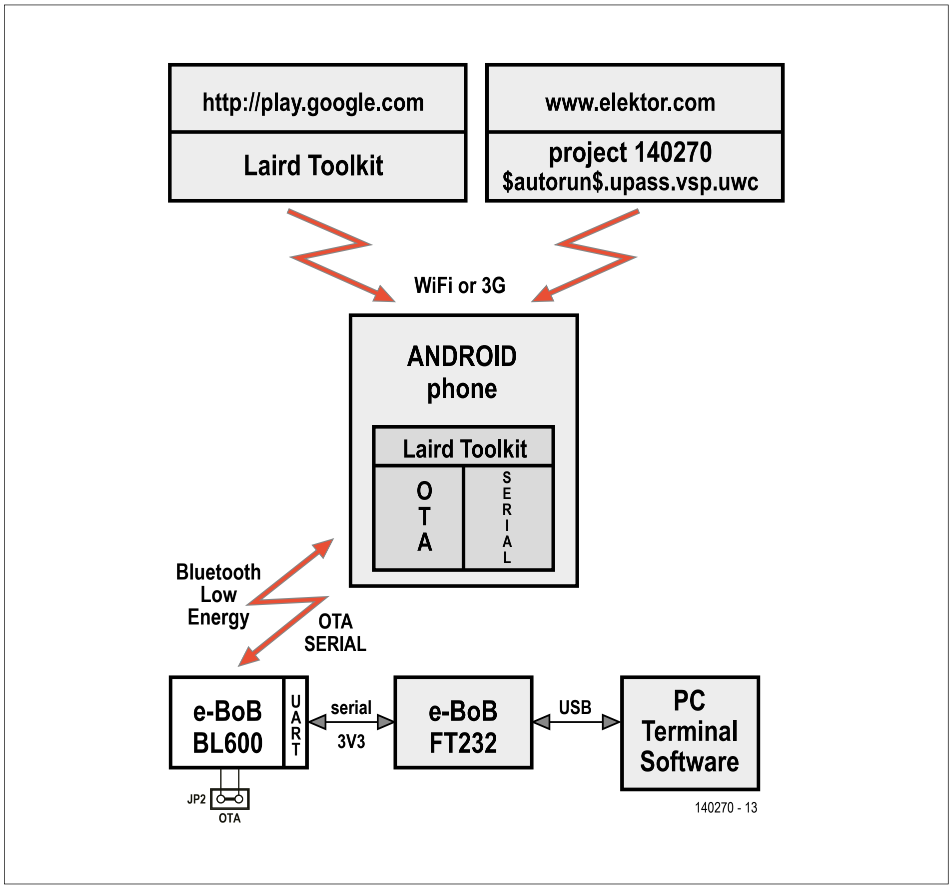


Figure 4. The application example I am suggesting consists of two steps. First of all, thanks to the OTA service, we send the program the e-BoB will be required to run at start-up to the BL600 by radio from an Android phone. Then this UART (or VSP) communication program lets you make your phone communicate with a BLE connected object — in this case, a PC.

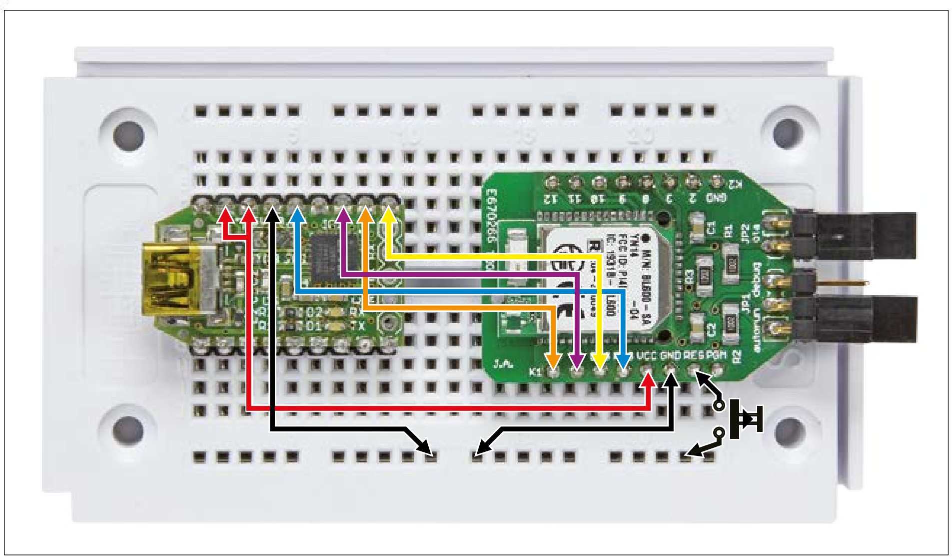


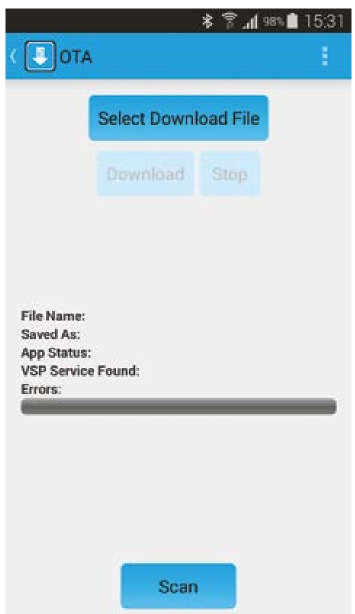
Figure 5. Wiring on the breadboard for connecting the BL600 e-BoB and the FT232 e-BoB for the application in Figure 4.

plify our first test, the connected object is going to be, quite simply… your PC! After all, isn't that the first connected object we all have at our fingertips?
The operation is performed in several steps, summarized in the illustration in **Figure 4**.
Thanks to the remarkable possibilities of Elektor's new BL600 e-BOB, we're first going to download the program for it, in wireless mode (!), and from our smartphone, thanks to the "Over The Air" service; OTA is a standard function of Bluetooth Low Energy that we're going to be taking advantage of.
Once the communication program has been downloaded from the phone to the e-BoB, we'll exchange some data between the Android phone and the PC via the UART interface.
To keep things simple, we're going to start off with an example provided by Laird technologies [5], namely their UART (or VSP) communication program `upass.vsp.sb`. Here, we're not going to be looking at either its source code or how to compile it, but will go directly to the already-compiled version, which you can find on the Elektor website [6] in the `$autorun$.upass.vsp.uwc` file, all ready to run. The fact this file is compiled and is named `$autorun$.xxx` actually offers two advantages: on the one hand, it can be transferred from the phone to the BL600 using BLE's "Over The Air" service (see **inset** about BLE); and on the other, once transferred into the e-BOB, this program will run automatically at start-up or following a manual reset.
To connect the BL600 e-Bob to my PC's USB, I've chosen the Elektor FT232 USB/serial bridge [7]. The experimental wiring will be done on a breadboard as in **Figure 5**.
All we need now is a program capable of operating the OTA and Serial services. Laird Technologies offers us this in the form of the `Laird BL600 Toolkit` application for Android smartphone.
Let's get going!

1st step: download the UART program to the e-BoB

For this first step, the BOB-FT232 is used only to power our BL600 e-Bob. On the latter, fit the OTA jumper JP2 and set jumper JP1 to the autoRUN position. Download the compiled file `$autorun$.upass.vsp.uwc` to your Android phone from the Elektor site [6]. Download the `Laird BL600 Toolkit` application from the Laird site [7], run it, and choose the option OTA (Over The Air), then click on Select Download File

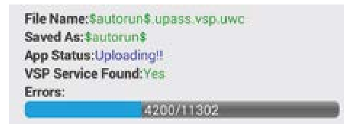
and search for the `$autorun$.upass.vsp.uwc` file on your phone. Now run Scan:



then choose the LAIRD BL600 service

If it is not displayed, reset the module (RESET). And then hit Download…

During the transfer of the file from the phone to the e-BoB, the progress bar will increment.



We're there — the BL600 is ready! All that remains is to terminate the OTA connection (Disconnect), reset the module manually, and quit the OTA application.

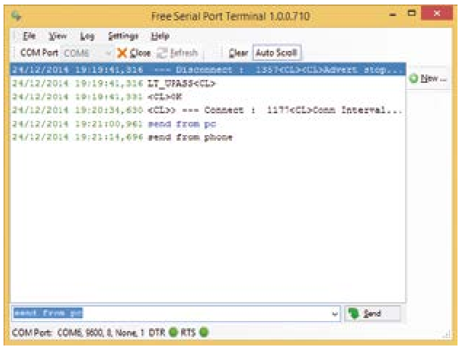
2nd step: testing the communication

I use the "Free Serial port Terminal" program, but any other terminal program will do. On the phone, you need to run the Laird BL600 Toolkit application, choose the Serial tool, then Scan. Then connect to the module.

Your Android phone is now ready to communicate with your PC via the BL600 module on its e-BoB.

Here's an example of exchanging text:

- - from the phone to the e-BOB and PC: send from phone
- - via the e-BoB to the PC, to the phone: send from pc



OK, there's maybe nothing very revolutionary about exchanging data between phone and PC — but isn't this simple application a convincing first demonstration of our e-BoB's capabilities? Over to you now to use it in your own projects! Elektor will be devoting other articles to the BL600 module in forthcoming issues. Stay connected!

I've posted online [8] a video of a remote-control application produced last summer using an earlier version of the e-BoB. I hope in this way I can encourage you too to think up Bluetooth Low Energy projects with Android or iOS phones. Thanks to Laird technologies who are putting online the sources for its Android and iOS programs (the Apple license is not free).

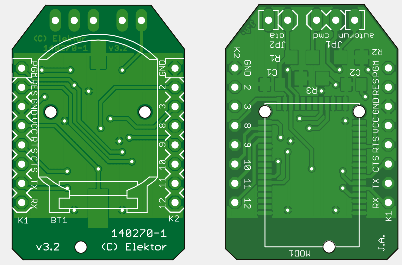
140270

Component List

Resistors:
(5% 250 mW 1206)
R1,R2 = 10kΩ
R3 = 12kΩ

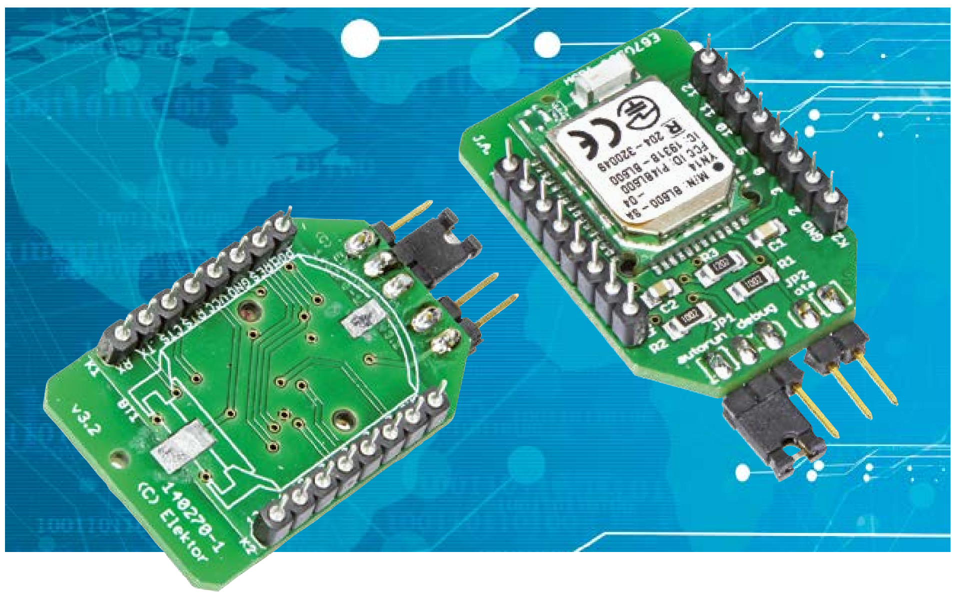
Capacitors
(25V 0805)
C1,C2 = 100nF

Miscellaneous
MOD1 = BL600-SA Bluetooth Lo-Energy module (Laird Technologies)
K1,K2 = 8-pin pinheader, 0.1" pitch
JP1 = 3-pin pinheader, 0.1" pitch
JP2 = 2-pin pinheader, 0.1" pitch
2 jumpers 0.1"
Battery holder S8421-45R (option) (2115305)
Battery type CR2032 (optional)
PCB, Elektor Store # 140270-1



BL600e-BoB, assembled, Elektor Store # 140270-91
(K1 & K2 included as loose parts)

(Farnell part numbers in parentheses)



Selection of topics to be covered in future episodes of this series on the BL600 e-BoB:

- inputs/outputs: a light-chaser
- handler, events
- the Red Green Blue program
- Low Energy, 5 µA
- the I²C port
- the SPI port
- Bluetooth communication
- explanation of the remote wireless thermometer program
- writing a program for Android
- writing a program for iOS (hmm… the Apple license is not free)

Web Links

[1] Wireless thermometer using the BL600 (Elektor January & February 2015)
www.elektor-magazine.com/140190

[2] Wireless thermometer using the BL600 in the Elektor Store
http://www.elektor.com/bluetooth-thermometer

[3] elektor.labs video on the thermometer using the BL600
http://youtu.be/WZSQZGUgJXI

[4] Laird Technologies documentation on the BL600
http://www.lairdtech.com/Products/Embedded-Wireless-Solutions/Bluetooth-Radio-Modules/BL600-Series/

[5] source code
https://laird-ews-support.desk.com/?b_id=1945#software

[6] www.elektor.com/140270

[7] USB/Serial bridge: Elektor BOB-FT232R
http://www.elektor.com/ft232r-usb-serial-bridge-bob-110553-91

[8] Another example of a BL600 application
https://www.youtube.com/watch?v=SxwaVI0Kkk8

[9] Author's website
www.aubinais.net

Guitar Overdrive

With genuine 'germanium-sound'

By A.J. Köhler (Germany)

The project described here is an effects device for a guitar, where the distortion effects are generated with the aid of paired silicon and germanium diodes. Very beautiful sound effects are generated because the diodes are driven by controlled current sources (OTAs).

There are countless sound effect circuits available for guitars with all kinds of features and varying in price from cheap to very expensive. The author of the circuit described here has already spent a lot of time on this subject. He has analyzed a large number of these devices and subsequently developed his own version. He is still very busy with modifications and extensions to his project, which is updated regularly on the Elektor.Labs website. Since this is a project where you could continue *ad infinitum* making further improvements and optimizations, we decided to publish a particular version of the project now. Elektor Lab have made some small changes to this circuit in, equipping it with readily available components, construct it and test it. So you can copy the circuit with the components described here and everything should work just fine. But after you have finished this and would like to experiment with other configurations or components, then you will find and abundance of tips, modifications and component variations in the description on the Labs website [1]. We therefore limit ourselves here to the technical description of the version of the circuit as it was built by us.

Current control with OTAs

Many effects circuits use diodes to distort the signal from a guitar. These diodes round the signal from the guitar and are usually connected in anti-parallel in order to obtain a symmetrical signal waveform. A great many variations of this concept have already been invented. The author has designed a configuration where the diodes in his circuit are not controlled with a voltage, but with a current. This results in much more beautiful effects, according to him. In addition, he has also experimented extensively with different types of diodes. These experiments include sil-

icon diodes, LEDs and of course different types of germanium diodes. The latter are usually popular with guitar players. This circuit is suitable for all the aforementioned types and in the default configuration you can already choose between silicon and germanium using a switch.
The diodes are driven completely symmetrically with the aid of OTAs. An OTA (Operational Transconductance Amplifier) looks very similar to an ordinary opamp, but the big difference is that the output supplies a current instead of a voltage. The voltage difference between the two inputs determines the output current. Furthermore, an OTA has a few additional connections which set the transconductance (the ratio between output current and input voltage, or in other words, the gain) and the input bias current.
Since about 1970 a number of different OTA ICs have become available and these have been reasonably popular, but in recent years they have become less fashionable and most manufacturers have stopped making them. One type that is still readily available is the dual OTA LM13700 made by TI. This is therefore the one that is used in this design. The author has also experimented with other types of OTA, but the problem with these is that they are either completely unavailable or can only be sporadically obtained.

The diodes are driven completely symmetrically with the aid of OTAs.

The circuit

The circuit which is shown in its entirety in **Figure 1**, comprises roughly three parts. The first is the input section with a buffer and a pre-amplifier, the second is the overdrive stage with the two OTAs and the diodes, and the third is the output section with a tone control stage.
The buffer at the input is built from an N-channel JFET type BF545A (T1). This ensures that there is almost no load on the signal from the guitar (input impedance of 1 MΩ). This is followed by an amplifier stage which takes the form of low-power opamp IC4 (a TLC271). This amplifies the buffered signal by about 2 times; if necessary, you can change R6 to obtain a different gain. This opamp has a bias-pin (pin 8), which you can use to change the characteristics. Here it is set to give a medium bias, which combines a reasonable speed with a low power consumption.
The output signal goes via foot switch S1.C (*Effect on/off*) to two OTAs IC3.A and B which drive the distortion diodes. When you operate switch S1, the output signal from IC4 goes, via R49, directly to the output buffer IC5.B so that the entire distortion section is bypassed.
The push-pull circuit around IC3.A/B is driven via R13 and R18. You can adjust the symmetry of the signal with potentiometer P1 (*Symmetry*) which, via R11 and R17, applies a DC voltage to the other inputs of the OTAs. The purpose of trimpots P7 and P8 is to set the zero bias point of the OTAs.
Between the OTA-outputs are the two pairs of anti-parallel connected diodes in series with resistors, a silicon pair (D3/D4) and a germanium pair (D1/D2). These must be matched pairs for optimum results — we'll say more about that later. With the aid of switch S2 (*Soft/Hard*) you can switch between the two diode pairs (it is also possible to use red LEDs instead of the silicon diodes). You can adjust the (symmetrical) output voltage between the two OTA outputs — and with that also the operating point of the diodes — with P5 and P6 respectively.
The gain of the OTAs — and with that the amount of 'overdrive' — is adjusted with a current into the amp-bias-input pins 1 and 16 of the OTAs. This current effectively sets the 'gain'. This current is supplied by opamp IC1.B. In order to make

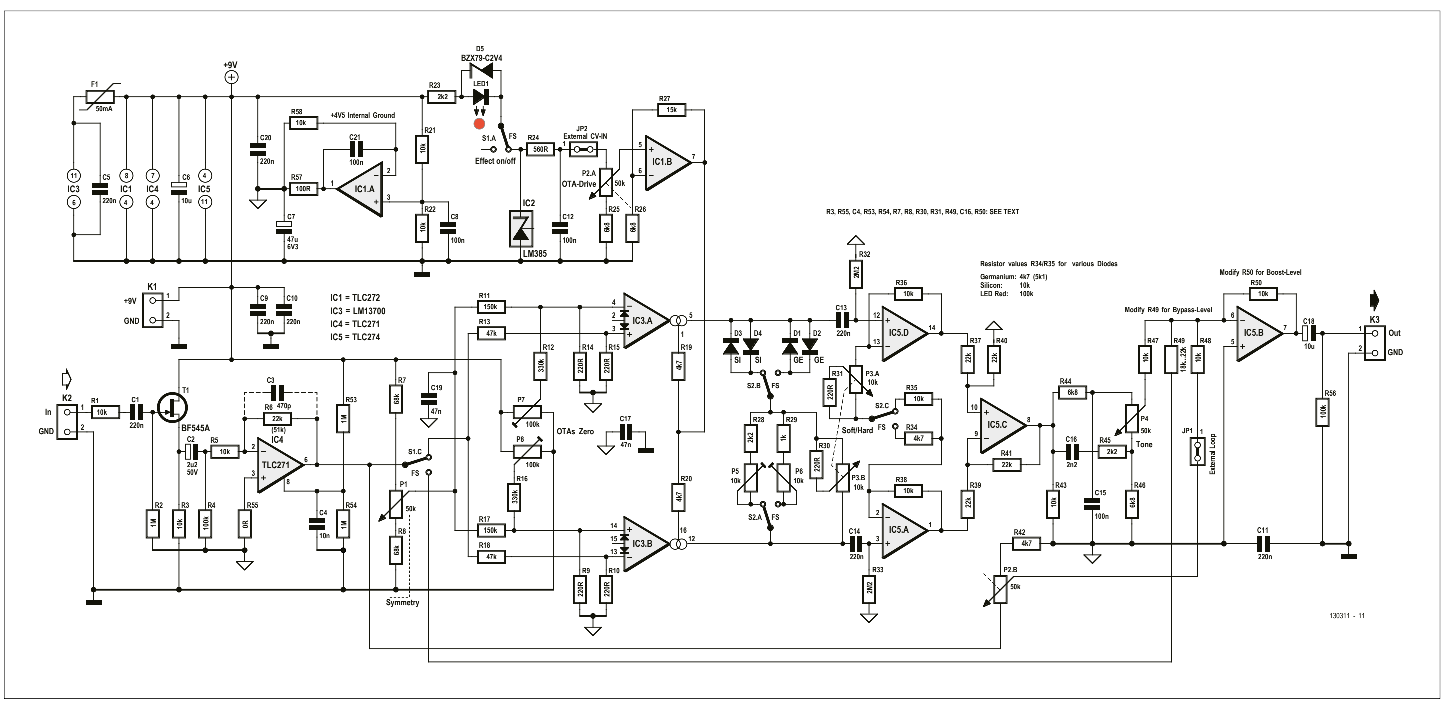


Figure 1. The entire schematic for the guitar distorter. The main players are two OTAs and a few diodes.

sure that this opamp supplies a very stable voltage it uses a voltage-reference IC (IC2) which is connected via R23 and LED1 to the power supply (a 9-V battery). This supplies a stable DC voltage of 2.5 V. Depending in the position of potentiometer P2.A (*OTA-Drive*), this opamp generates a DC voltage which will result in a certain amount of current through R19 and R20. LED1, incidentally, functions as the power-supply-on indicator. Zener diode D5 that is in parallel serves as a bypass, should the LED become faulty (this LED is, after all, mounted on the outside of the effects enclosure and could suffer a knock every now and then during use).

When the bias-current is changed with P2.A, the amount that is contributed by the original signal to the output stage is changed with P2.B at the same time.

Component List

Resistors

Default: SMD1206, 0.25W 1%
R1,R3,R5,R21,R22,R35,R36,R38,R43,R47,R48,R50,R58 = 10kΩ
R2,R53,R54 = 1MΩ
R4,R56 = 100kΩ
R6,R37,R39,R40,R41,R49 = 22kΩ
R7,R8 = 68kΩ
R9,R10,R14,R15,R30,R31 = 220Ω
R11,R17 = 150kΩ
R12,R16 = 330kΩ
R13,R18 = 47kΩ
R19,R20,R34,R42 = 4.7kΩ
R23,R28,R45 = 2.2kΩ
R24 = 560Ω
R25,R26,R44,R46 = 6.8kΩ
R27 = 15kΩ
R29 = 1kΩ
R32,R33 = 2.2MΩ
R51,R52 = not fitted
R55 = 0Ω
R57 = 100Ω
P1,P4 = 50kΩ linear potentiometer, 5mm pitch
P2 = 50kΩ stereo, linear potentiometer, 5mm pitch
P3 = 10kΩ stereo, linear potentiometer, 5mm pitch
P5,P6 = 10kΩ trimpot, 0.1" pitch, Vishay Sfernice T73YP103KT20
P7,P8 = 100kΩ trimpot, 0.1" pitch, Bourns 3362P-1-104LF

Capacitors

C1,C5,C9,C10,C11,C13,C14,C20 = 220nF 50V, 10%, X7R, SMD1206
C2 = 2.2µF 100V, 20%, 2.5mm pitch, diam. 6.3 mm max.
C3 = 470pF 50V, 10%, X7R, SMD1206
C4 = 10nF 50V, 10%, X7R, SMD1206
C6,C18 = 10µF 100V, 20%, 2.5mm pitch, diam. 6.3 mm max.
C7 = 47µF 35V, 20%, 2.5mm pitch, diam. 6,3 mm max.
C8,C12,C15,C21 = 100nF 50V, 10%, X7R, SMD1206
C16 = 2.2nF 50V, 10%, X7R, SMD1206
C17,C19 = 47nF 50V, 10%, X7R, SMD1206

Semiconductors

D1,D2 = OA90, alternatively, D9B/OA191/MD276/D311/1N60/OA1161
D3,D4 = 1N914A, DO-35
D5 = BZX79-C2V4, DO-35
LED1 = LED, red, 3mm, wired
T1 = BF545A (SMD SOT-23)
IC1 = TLC272CD (SMD SOIC-8)
IC2 = LM385Z-2.5 (TO-92)
IC3 = LM13700M/NOPB (SMD SOIC-16)
IC4 = TLC271CD (SMD SOIC-8)
IC5 = TLC274CD (SMD SOIC-14)

Miscellaneous

K1,K2,K3 = 2-way PCB screw terminal block, 0.2" pitch
JP1,JP2,LED1 = 2-pin pinheader, 0.1" pitch
S1,S2 = foot switch, 3 C/O contacts (3PDT), PCB mount, 4 x 5 mm pitch (www.UK-electronic.de, no. 102-000 with solder eyes, or 102-000-1 with solder pins)
Clip with wires for 9-V battery
¼" (6.3mm) mono jack connector, panel mount
F1 = 50 mA Multifuse, radial (Bourns MF-R005-0)
PCB # 130311-1 [2]

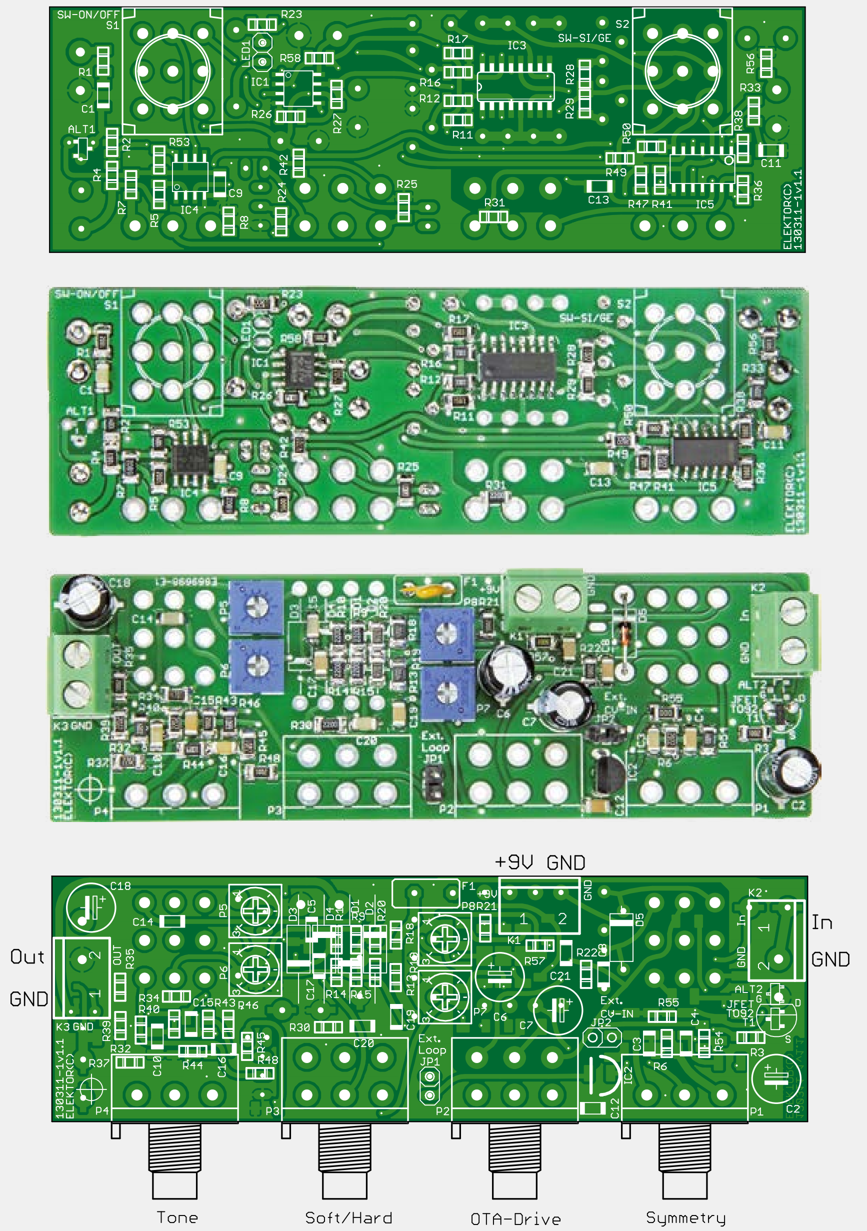


Figure 2. The circuit board has been designed for surface mount components and has components on both sides.

The signal across the diode-resistor circuit is buffered by IC5.D and A, where with the aid of another pole on switch S2 (S2.C), a different gain is selected depending on which diode pair is used. The output signals from these two opamps are then subtracted from each other by IC5.C, so that an asymmetric output signal remains.
This signal then passes though a simple treble tone-control section which is built around R43 through R46, C15, C16 and P4 (*Tone*).
Finally this is summed with the (by P2.B) attenuated original guitar signal, which originates from the output of IC4. When switch S1 is in the other position, the overdrive-section will not contribute an output signal and the output buffer IC5.B receives only the undistorted signal from IC4.
This concludes the description of the entire signal path. What we haven't dealt with yet is opamp IC1.A. This provides an artificial ground at half of the power supply voltage. After all, the power supply is only a single 9-V battery, and in this way it is possible to power the opamps from a pseudo-symmetrical power supply. The current consumption of the entire circuit amounts to only a few mAs, so such a small battery will last quite a while.

Paired diodes

For the optimal functioning of the circuit it is important that the characteristics of the two anti-parallel connected diodes are as identical as is possible. For this purpose the author has tested hundreds of OA91s at an identical ambient temperature. In a test setup the voltage drop across the diodes was measured at a current of about 1 mA and again at a current of about 50 μA (ratio 1:20). You can do that yourself with the aid of a regulated power supply (for example 15 V) and two resistors (for example 15 k and 270 k). The objective is to find two diodes that have an identical voltage drop (or as close as possible) at the first test current, and an identical voltage drop (which will be different from the first) at the other test current.

Construction

A compact circuit board has been designed for the guitar-overdrive, which also accommodates all the potentiometers and foot switches. In practice, it will usually work out that it is more convenient to fit the potentiometers directly on the circuit board and connect the switches via short wires because it will then be much easier to fit everything into an enclosure.
This circuit board shown in **Figure 2** contains, besides a few through-hole components, mainly SMD components. These are on both sides of the circuit board. You will need to have some soldering experience with these kinds of components in order to mount these on the circuit board reliably.

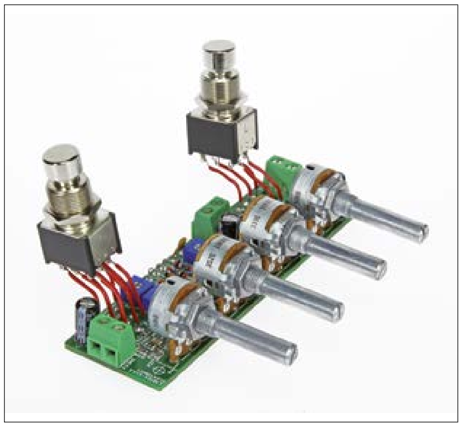


Figure 3. With this prototype you can clearly see how the switches are connected.

Figure 4. This is how the circuit is mounted into a sturdy enclosure.

Figure 3 shows the completed prototype, which differs a little from the final version. Start by fitting the components with multiple legs, these are always the hardest. Once that has been successfully completed you can continue with the remaining parts. After that, mount the potentiometers and the foot switches.
We originally designed the circuit board in such a way that these switches could be soldered directly onto it, but that turned out not to be so handy after all. The photo in **Figure 3** shows how we solved this. The switches are connected to the circuit board with insulated wires a few centimeters long (preferably as short as is possible), which gives you some wiggle room when mounting into a (metal) enclosure by first mounting the switches in the correct position and then sliding the circuit board so that the potentiometers fit properly in their appropriate mounting holes. In this way everything will end up mounted robustly in the enclosure.
Now we'll briefly return to the circuit board, because the diodes must be fitted last. This has to be done with a certain amount of care, because the germanium diodes (a point-contact diode!) in particular, change considerably if they become too hot. Mount the diodes about a centimeter above the circuit board and wrap the diode body in a damp cloth while soldering to ensure that it will not become too hot. Another solution is to hold the leg that is soldered with some small pliers which will conduct the heat away.
Once this is all done you can connect the battery and adjust the circuit before putting it in the enclosure. Set P1 to the center position and adjust P7 and P8 (without an input signal) so that the output voltage at the two OTAs is 4.5 V. The settings for P5 and P6 are done by ear; set these so that the soft/hard changeover with S2 result in an effect that you find sounds the best.
For the enclosure we have selected a sturdy aluminum enclose made by Hammond, type 1590TBK. Everything fits neatly inside it and you have a robust unit that can be placed on the floor without any problems and be operated with a foot. Attach the 9-V battery with a small piece of double-sided tape in the inside of the lid (which here functions as the bottom of the enclosure). The two ¼" (6.3-mm) jack connectors for the connections to the guitar and the amplifier can be mounted on the sides of the enclosure. So, now you can try all the effects using the available knobs and switches. Enjoy your experiments!

(130311)

Web Links

[1] www.elektor-labs.com/9130703401

[2] www.elektor-magazine.com/130311

By Sunil Malekar (Elektor Labs India)

A proportional radio control is ideal for controlling the speed and direction of model cars, planes or boats. Unfortunately, there are often very few facilities for switches on such controllers. With the help of this circuit you can control five switches using just a single channel. The switching is effected by relays at the receiver side.

Model planes, boats and cars are usually controlled with a proportional remote control. The control signals from the transmitter are processed by the receiver in the model. This then outputs a number of signals (depending on the number of available channels), which are used to drive servos, etc. These signals are encoded using pulse-width modulation. Each signal is repeated every 20 ms, where the width of the pulse is between 1 and 2 ms; 1.5 ms corresponds to the central position of a servo. The position of each joystick in the transmitter (in the X or Y direction) produces a specific pulse width.

This type of proportional servo controller is of course perfect for a continuously in collaboration with DESIGNSPARK

Radio Controlled Multi-Switch

Control 5 switches via 1 channel

adjustable speed or direction of a model, but it is less easy to implement switching functions. This requires extra channels, which may not always be available.

Model boats and trucks in particular often have other functions that you'd like to be able to turn on and off remotely. A siren, water cannon and lights come to mind. The circuit described here has been designed with that purpose in mind: it can control up to five functions using just a single channel. The position of a joystick or slider determines which function is turned on. You can only have one of the five functions on at any time.

The circuit consists of a single-gate oscillator, a decade counter and two buffers. The principle is very straightforward. As soon as a pulse is received, a counter with ten outputs is started. When the pulse is finished, one of the outputs will be active, depending on the length of the pulse. The relays have been connected to the counter outputs in such a way that there always is some dead space between each pair of switch points.

Circuit diagram

The circuit diagram for the multi-switch is shown in **Figure 1**. As we said earlier, the pulse coming from the receiver lasts between 1 and 2 ms, and is repeated at intervals of about 20 ms. This pulse enters the circuit via connector K7 and is then inverted/buffered by gate IC1.A. It then continues along two different paths. The positive edge of the pulse (which is right at the start) creates a short pulse (by differentiator C2/R2). This is buffered by IC1.B and resets the counter (IC2). At the same time, the clock oscillator built around IC1.D is started via IC1.C. This oscillator generates a square wave of 5 kHz (set by P1). This causes the decade counter (IC2) to be incremented every

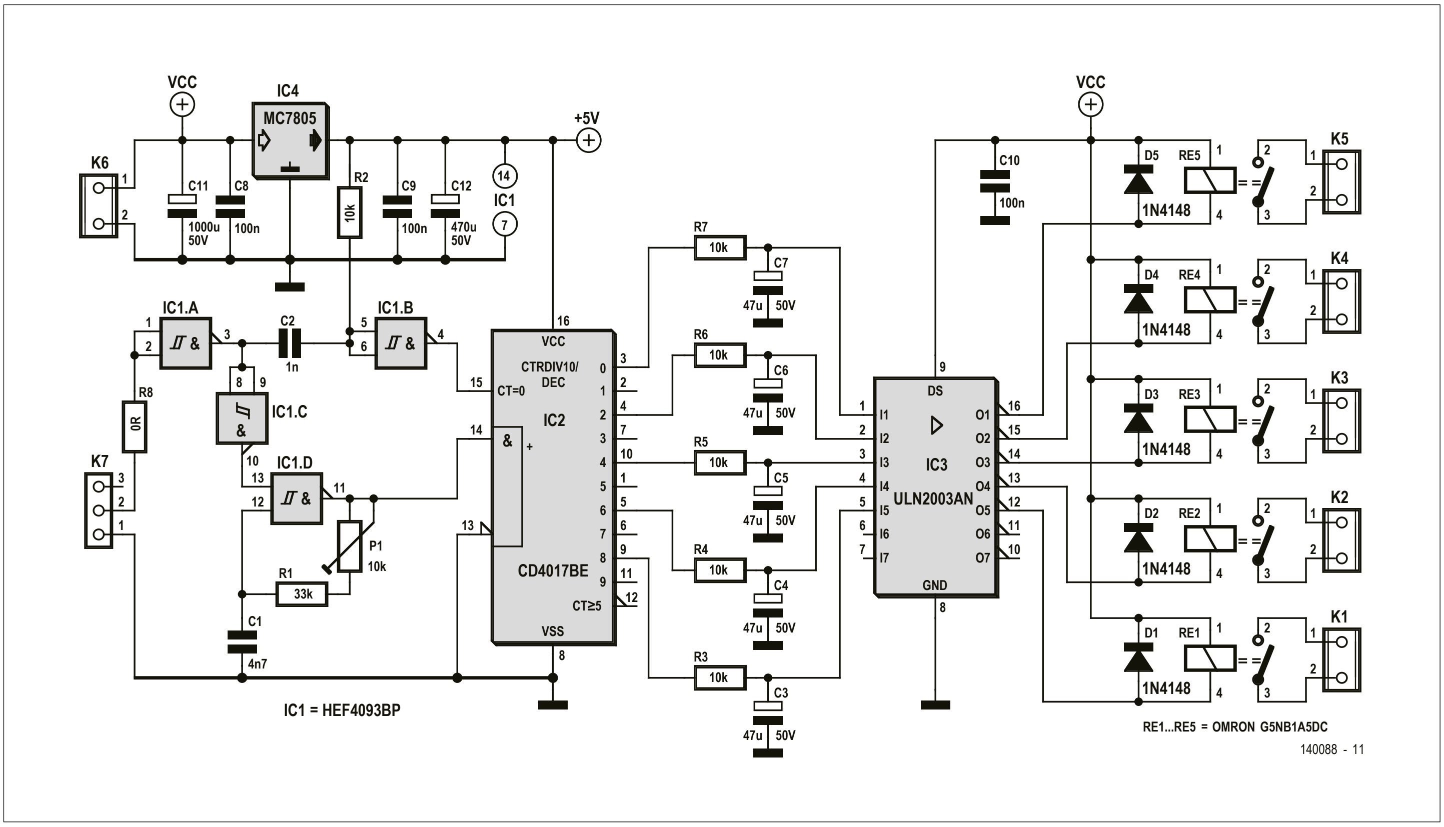


Figure 1. The multi-switch uses a decade counter to ‘measure’ the length of the control pulse.

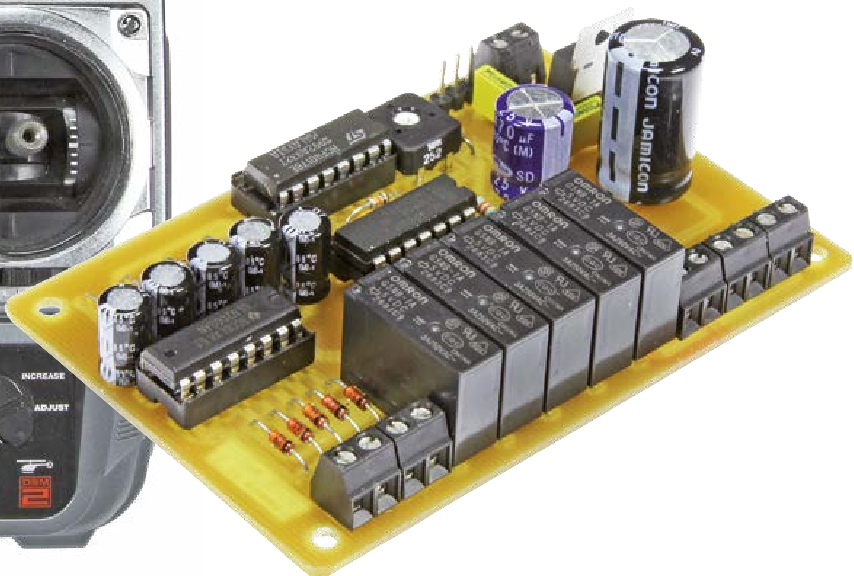
0.2 ms, as long as the oscillator is on.

IC2 now counts through the outputs at 0.2 ms intervals, starting with output zero. After 0 ms the first output (output ‘0’) will be high, after 0.4 ms it will be output ‘2’, etc. In this way the clock pulses generated by IC1.D will make all outputs go high in turn. In theory, we could use this method to create ten switched outputs. In this instance, we have only used five of the outputs. The intermediate (unconnected) outputs serve to create a dead space where no relay is activated.

IC1.D will only generate pulses as long as the input signal at K7 is high. In other words, it will do so for the duration of the pulse. When the input signal returns to zero the oscillator stops and the counter output that was active at that time will stay in that state until the next input pulse arrives some 20 ms later. If this pulse has the same width, as well as those following it, then that particular output will (almost) remain active continually. There will just be a short glitch every 20 ms while the counting takes place.

The output signal is averaged with the help of integrator networks connected to the counter outputs (R3/C3 to R7/C7), which removes these glitches.

The outputs of the integrator networks are connected to the



inputs of driver IC ULN2003, which contains 7 power Darlington stages. These are used to switch relays RE1 to RE5 on or off. Since the power stages are inverting, the relays are connected between the positive supply voltage and the outputs of the ULN2003. A flyback diode is connected in parallel with each relay coil to suppress the reverse emf generated by the coil when it is turned off. The lamps, motors, etc. that need to be switched should be connected to connectors K1 to K5.

This concludes the explanation of the circuit operation. There is also a voltage regulator on board (7805) along with its required capacitors. The supply voltage for the circuit should be between 7 and 20 V. The current consumption of the complete circuit is only a few mA (when no relay is active).

Construction

The board for the multi-switch (**Figure 2**) has a spacious layout, which makes the construction quite easy. This board will normally be added to a model car or boat that has sufficient room for all the extras that you want to control. In that case you should be able to easily add this board as well. The construction will be standard, since only through-hole mounted components have been used. If you prefer, you can mount all the ICs in sockets.

Connecting the circuit to the receiver is done in the same way as for a standard servo. K6 is connected to the receiver battery and the connector with the control signal is plugged into K7 (you should first verify that the order of the connections is the same). The supply lines for the devices you want to control should be connected to K1 to K5.

The calibration of the circuit is also a simple matter. Preset P1 has to be adjusted so that all channels switch reliably when the joystick is moved from one extreme to the other. Enough space is left between each relay switch position for a dead space, where none of the relays is active. To make things easier, you can always add a few markings next to the slider or joystick to indicate where each of the relays becomes active.

(140088)

Web Link

[1] www.elektor-magazine.com/140088

Component List

Resistors
R1 = 33kΩ
R2–R7 = 10kΩ
R8 = 0Ω (jumper)
P1 = 10kΩ trimpot

Capacitors
C1 = 4.7nF, 0.1” pitch
C2 = 1nF, 0.1” pitch
C3–C7 = 47µF 50V, 0.1” pitch
C8,C9,C10 = 100nF, 0.2” pitch
C11 = 1000µF 50V, 7.5mm pitch
C12 = 470µF 50V, 0.2” pitch

Semiconductors
D1–D5 = 1N4148
IC1 = HEF4093BP
IC2 = CD4017BE
IC3 = ULN2003AN
IC4 = MC7805

Miscellaneous
K1–K6 = 2-way PCB terminal block, 3.5mm pitch
K7 = 3-pin pinheader, 0.1” pitch
RE1–RE5 = relay, SPST, 5V coil, 5A contact, e.g. Omron G5NB1A5DC
PCB, Elektor Store # 140088-1

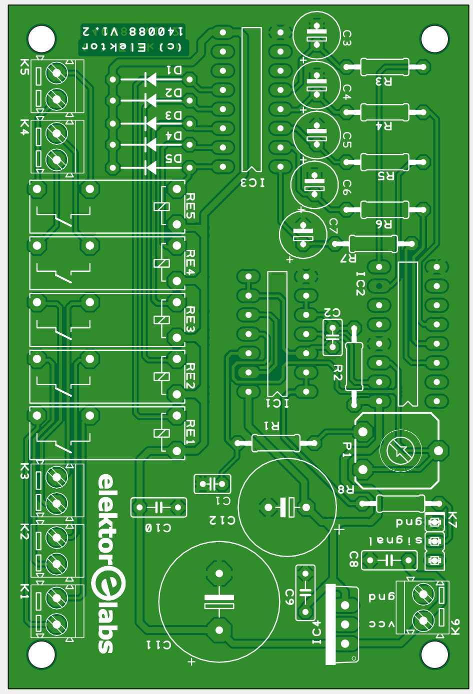


Figure 2. The board for this circuit has a spacious layout and uses only through-hole mounted components.

Experimenter's Transistor Tester

For go/non-go and device gain testing

By **Sunil Malekar** (Elektor Labs India) and **Luc Lemmens** (Elektor.Labs)

This simple Platino-based Transistor Tester is designed to subject dodgy and unmarked transistors to OK/RIP and gain tests in a simple way. Transistor matching is also within reach. The instrument handles NPN, PNP bipolar transistors as well as n- and p-channel MOSFETs.

Some say it's tedious to read the markings on transistor devices and look up their characteristics in datasheets. Others claim that's the only proper way of going about in the land of three-legged silicon. Either way, a handy transistor tester (and the Internet) can be helpful, especially for quick go/non-go tests, thermal/mechanical shock tests, and device matching (much hyped in high-end audio circles).

In line with the two other instruments in this series the Experimenter's Transistor Tester is based on the 'Platino' AVR microcontroller board [1]. The instrument not only identifies the leads of the device under test (base, collector, emitter; gate, source, drain) but also the structure (PNP/NPN; n-channel/p-channel), the technology (bipolar/MOSFET), and of course that quintessential electrical characteristic called H_{fe} expressing the device gain. Platino has everything on board to interface an inherently analog test circuit for the DUT to a readout (LCD) and a pushbutton. The rest is firmware being executed by an AVR micro.

Specifications & Features

- Tests bipolar NPN/PNP; MOSFET N/P channel
- Device gain range: 5 – 999 (approx.)
- Auto identification of bipolar/MOSFET, P/N type, c/b/e/ leads, MOSFET gate lead
- Indicates faulty or no transistor
- ATMega32 microcontroller on Elektor Platino board
- Bascom AVR software (free)
- 20 x 4 LC Display
- Optional control and readout via serial terminal
- DC Input: 12 – 18 VDC

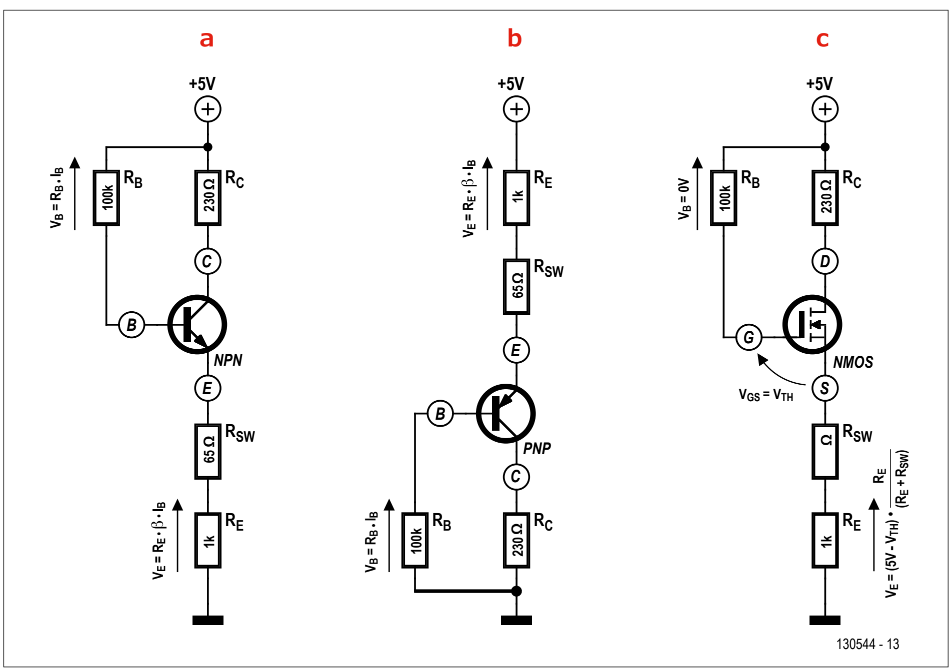
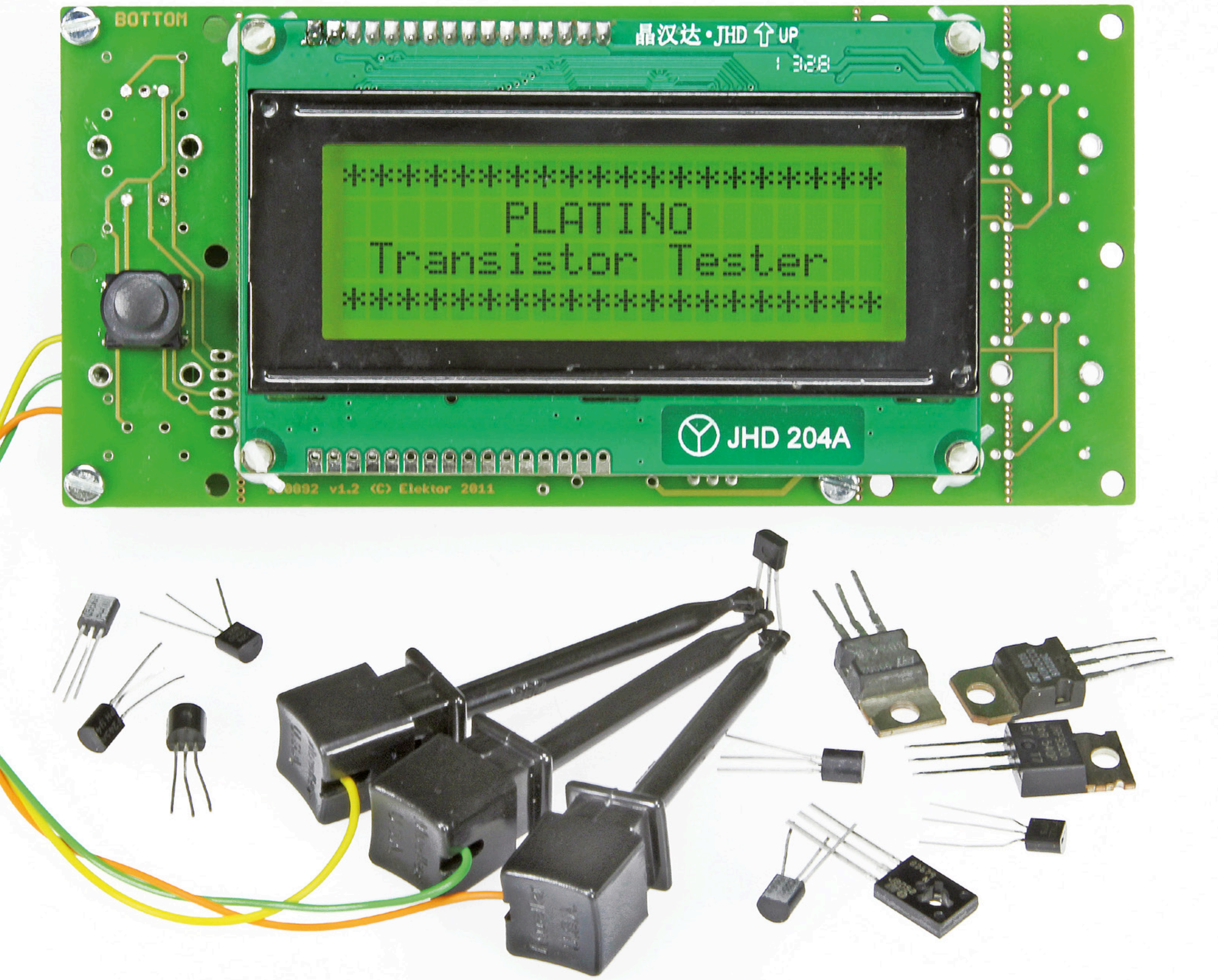


Figure 1. Elementary configurations for measuring the value of HFE (b) for a transistor and the threshold voltage of a MOSFET.

A few years back

The Experimenter's Transistor tester was inspired by, and designed around, an earlier Elektor publication, Michel Waleczek's *SC Analyzer 2005* [2]. Those of you interested in the theoretical model of 'blind' transistor testing should download and digest that article (it's free to all Elektor members). If on the other hand, you are more of a consumer at this point, then read the following digest. By the way, by 'blind' testing we mean just dropping the trannie in the test socket without any prior intelligence on it being bipolar or MOSFET, PNP or NPN, P- or N-channel, ADHD, or even what ball park the gain is in. Blind testing should be non-destructive to the DUT as well as the test circuit, of course.



How do they do it?

The descriptions below apply to properly working transistors and should be interpreted as generic. They do not cater for Darlingtons, UJTs, thyristors, some RF germanium transistors, logic MOSFETs, diode-protected transistors, and a galaxy of other special devices out there. In practice, the pin determining and test method described works for the majority of run of the mill transistors out there. In case of doubt, assume a bipolar silicon NPN transistor type BC547B or a 2N2222 is being tested.

NPN or PNP?

Two of the three transistor leads e(mitter), b(ase), c(ollector) are virtually grounded via 100-Ω resistors while the other is pulled to +5 V via 5.6 kΩ. The voltage drop across the resistor is measured and stored. Next, a sequence of different ground/5.6-kΩ lead configurations is arranged by test circuit, and the voltage drop is measured for each configuration. Three values are obtained in this way. **Table 1** shows the values that should theoretically be measured for NPN and PNP transistors. Here the minus sign corresponds to a connection to ground via a 100-Ω resistor, and the plus sign, to a connection to +5 V via the 5.6-kΩ resistor. An NPN transistor gives two values of approximately 5 V and one of approximately 0.7 V, while a PNP transistor gives a single value of 5 V and two values of 0.7 V.

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

Gain?

Since the NPN/PNP test does not establish the positions of e and c, the current gain is measured for each of the two possible combinations. The ultimate value is taken to be the greater of the two measured values.

If the measured voltages do not match any of the combinations in Table 1, the

Table 1. Initial measurements

Type	E	B	C	Measured value
NPN	–	–	+	5 V
	+	–	–	5 V
	–	+	–	0.7 V
PNP	–	–	+	0.7 V
	+	–	–	0.7 V
	–	+	–	5 V

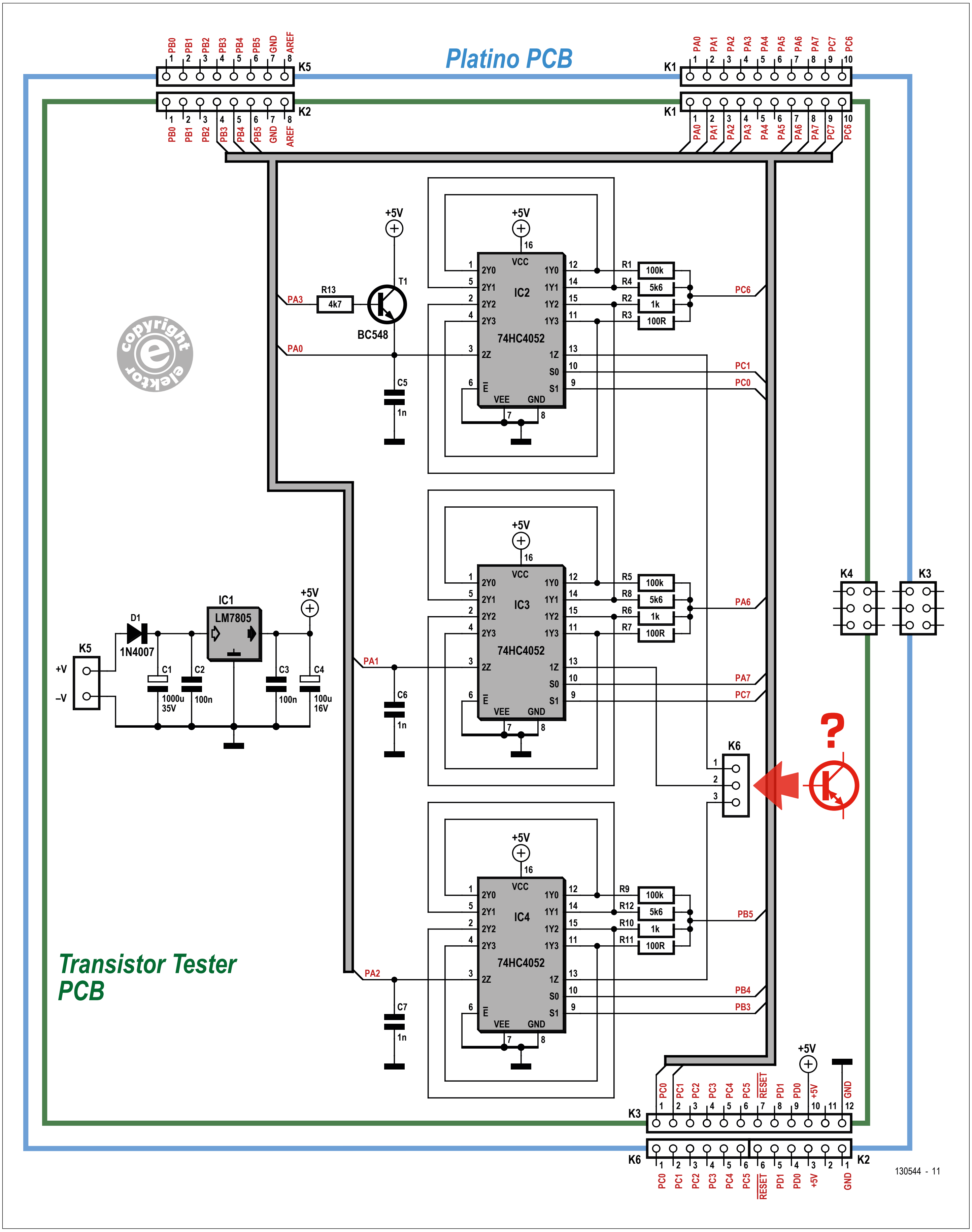


Figure 2. Schematic of the Experimenters’ Transistor Tester. If by chance you are missing the Platino control board, it’s represented by the blue perimeter!

device is subjected to special tests for other types of components (MOSFET, diode). To test whether the device is a MOSFET, the current gain is measured in a similar manner for all six of the possible lead arrangements.

The above measurements have identified the base lead of the transistor. Next, the other two leads are identified by connecting the DUT in a common-collector configuration if it is a bipolar transistor, or in a source-follower configuration if it is a MOSFET, see **Figures 1a, b, c**.

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

$$V_E = R_E \times (H_{FE} + 1) \times (V_B \div R_B)$$
$$H_{FE} = [(V_E \times R_B) \div (V_B \times R_E)] - 1$$

Note that in some literature the symbol □ is used in lieu of H_{FE}. The practical range in terms of gain that can be measured is about 5 to 999. You can tell an N-channel MOSFET (**Figure 1c**) from a bipolar transistor by its almost-zero gate current. In this case, the threshold voltage corresponds to $V_{CC} - V_E$ (assuming NMOS). Unfortunately J-FETS are not supported by the tester.

Circuit description

Our instrument has a hardware and a software component with strong interaction. We'll deal with the hardware first, specifically an outline of its operation, see **Figure 2**. You are looking at a circuit that's plugged onto Platino.

The section around IC2, IC3, IC4 effectively reads the voltage across each of the three leads of the transistor under test (DUT) connected on K6. CMOS analog 4-to-1 multiplexer ICs type 74HC4052 are used to juggle the connections of the leads to produce the required lead configurations. The resistor configurations required for measuring the various parameters of the transistor device

Listing 1. BASCOM AVR program (excerpt only)

```
cls
locate 1,1
'Transistor type and base/gate pin determined
'Now calculate transistor parameter and display results
Select case T_type
   case 1: 'NPN
           HFE_cal ' (T_type, base_pin, C_pin, E_pin,beta)

           lcd "Type : N-P-N"
           Locate 2,1
           LCD "C = ";c_pin; " B = ";base_pin;" E = "; e_pin
           Locate 3,1
           LCD "Hfe=";beta
   case 2: 'PNP
           HFE_cal ' (T_type, base_pin, C_pin, E_pin,beta)
           lcd "Type : P-N-P"
           Locate 2,1
           LCD "C = ";c_pin; " B = ";base_pin;" E = "; e_pin
           Locate 3,1
           LCD "Hfe=";beta
  Case 5: 'NMOS
           call cal_Vgs(Vt)
           lcd "Type : NMOS"
           Locate 2,1
           LCD "D = ";c_pin; " G = ";base_pin;" S = "; e_pin
           Locate 3,1

           lcd "Vth = ";fusing(Vt,"#.#");" V"

   CASE 6: 'PMOS
           Call cal_Vgs(Vt)
           lcd "Type : PMOS"
           Locate 2,1
           LCD "D = ";c_pin; " G = ";base_pin;" S = "; e_pin
           Locate 3,1
           LCD "Vth = -";fusing(Vt,"#.#");" V"
end select
```

Table 2. Platino jumper settings

JP3:	PC5
JP4:	PB0
JP5:	PB1
JP6:	PB2
JP14:	PC7

Table 3. Platino-to-Tester board interfacing

Pin Designation	Function
PA0, PA1, PA2:	ADC pins connected to transistor leads to read voltages
PA7, PB3, PB4, PC0, PC1, PC7:	Multiplexer select lines
PA6,PB5,PC6:	Digital output used for pull-up, pull-down configuration for resistors
PB0–PB2:	Encoder with pushbutton
PC5:	LCD backlight control

are set up (under software control) at the MUXs' paralleled inputs 1Y*n*/2Y*n*. The resultant voltage drop is read by connecting the lead to V_{CC} via a 5.6-kΩ resistor (R4, R8, R12) while the other two leads are connected to ground via 100-Ω resistors (R3, R7, R11). After reading the voltages across each lead through the multiplexer outputs *n*Z, the b(ase) lead of the DUT is identified by comparing the voltages developed on the leads, and using the fact that the voltage across the base pin (w.r.t. 'ground') is always different from the voltages across the other two pins. The NPN/PNP/NMOS/PMOS determination of the DUT is then done in software on the basis of Table 1 discussed above.
The '4052 MUX outputs are connected to Platino's ADCs via extension bus lines PA0, PA1, and PA2. Transistor T1 is the calibration switchover governor.
The power supply section on the add-on board is conventional. The 12 to 18 VDC raw input voltage from a wall wart or similar is applied to screw terminal block K5, with polarity protection afforded by diode D1. The LM7805 regulator supplies a regulated +5-volts to the microcontroller and other components that operate from the +5-V rail.

Platino

The Platino contraption has been discussed extensively in various past publications. Some say it's a platform, others, a board. In practical terms it's an ATmega32 microcontroller on a small board, all optimized for connectivity and ease of programming using all the famous AVR tools. Platino by default sports an LCD, a rotary control and pushbuttons. Here, it has to be configured as summarized in **Table 2**. Do follow the jumpering carefully! A summary of I/O, control and ADC pin usage is given in **Table 3**. A 20 x 4 LCD display along with a single pushbutton allow for easy user interaction. Only in case you need serial readout from the tester as well as a degree of interaction, you can interface Platino to Elektor's FTDI USB-to-Serial BOB module and see DUT readings in a terminal program. From there, it's an easy road to compiling spreadsheets and automated test reports, right?
The DUT interface is plugged onto Platino and electrically linked through connectors K1, K5, K2/K6.

Software

The firmware for the project was written in BASCOM AVR for the ATMega32 microcontroller. The Platino board itself was used as a development tool in the process of this project publication. If you wish to deepen your knowledge of BAS-

Component List

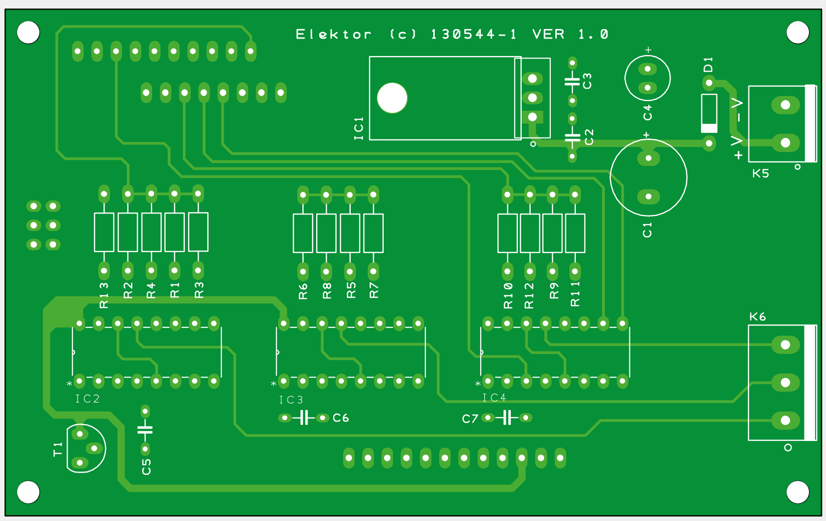
Resistors
(5%, 250mW)
R1,R5,R9 = 100kΩ
R2,R6,R10 = 1kΩ
R3,R7,R11 = 100Ω
R4,R8,R12 = 5.6kΩ
R13 = 4.7kΩ

Capacitors
C1 = 1000µF 35V radial
C2,C3 = 100nF
C4 = 100µF 16V
C5,C6,C7 = 1nF

Semiconductors
IC1 = MC7805
IC2,IC3,IC4 = 74HC4052
D1 = 1N4007
T1 = BC548

Miscellaneous
K1,K2,K3 = single-row pinheader socket, cut to size from 36-way part
K4 = dual-row pinheader socket, cut to size from 72-way part
K5 = 2-way PCB screw terminal block, 0.2" pitch
K6 = 3-way PCB screw terminal block, 0.2" pitch
3 pcs DIP-16 IC socket
PCB, Elektor Store # 130544-1

(Component descriptions from ELPP)



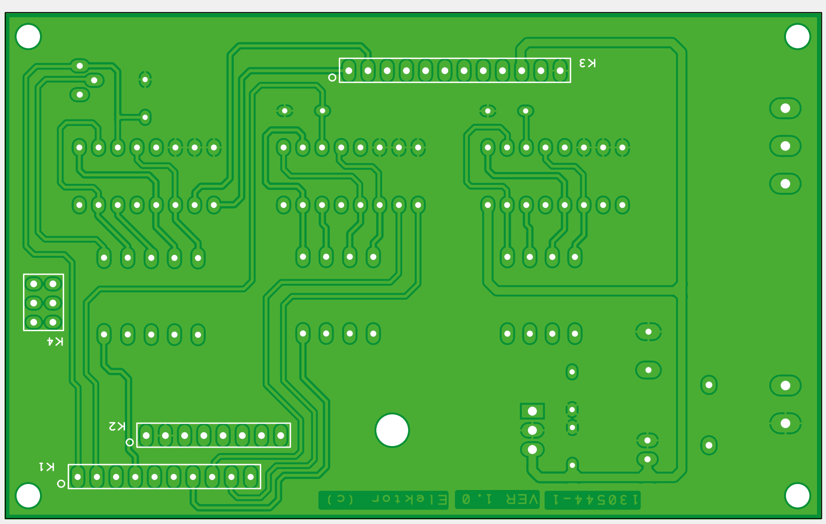


Figure 3. Printed circuit board for the Experimenter's Transistor Tester. Once populated the board can be plugged onto the Platino microcontroller board.

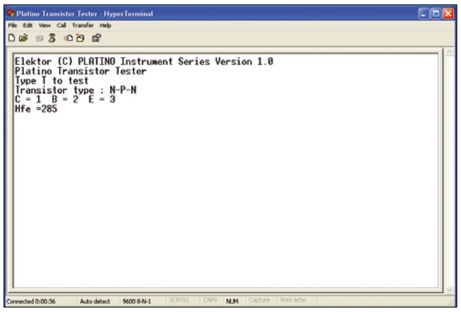


Figure 4. The tester can communicate with you and your PC through a terminal.

COM AVR, do consider spending half an hour daily on Elektor's fab *Microcontroller BootCamp* series started back in April 2014 [3].

Get the archive download at [4], a snippet of the code is given in **Listing 1**. The software is made up of three main modules.

Display section

The Display section is written to serve up the type of the transistor on the display, its CBE pinout and H_{FE} value in common-collector configuration.

Test section

This section reads the voltage developed across the resistance in each lead to the transistor. The multiplexer ICs are used to switch the lead constellations. After reading the voltages across each lead, PNP/NPN/NMOS/PMOS is established.

Gain calculation section

The gain is measured using Platino's ADCs. Having identified PNP/NPN and the base lead of the DUT, the identification of the other two leads remains unknown. This identification is done by assuming any one lead as the collector and connecting the transistor in common-collector configuration, then calculate the H_{FE}. By switching the MUXs we reverse the connections and calculate H_{FE} again. The greater value is considered the correct one, which firmly establishes the c and e leads.

Construction

First build the Platino board with its LCD and single pushbutton, ATMEGA32 MCU and all other components, then 'jumper' it as shown in Table 2.

The add-on board is all through-hole technology, spaciously laid out and designed to secure to the back of Platino with the help of connectors. Construction should be straightforward using the **Component List**, the component mounting plan in **Figure 3**, and the photographs in this article. For a professional look, build the board assembly into a case and fit it with a ZIF socket for the DUT on the front.

Connect the Elektor USB to Serial BOB module if you wish to see the output on your serial terminal (**Figure 4**). The serial baud rate is set to 9600.

Those of you wishing to program their own microcontroller for the project will find the Fuse Settings shown in **Figure 5** indispensable.

Testing 1–2–2N...

Power up the device by connecting a 12 – 18 VDC supply to connector K5. Connect the transistor to be checked to connector K6 **using any lead order you find convenient**. When all the connections are done, press the pushbutton. Alternatively, if the output is to be viewed on a serial terminal (optionally) then input 'T' from there. Some results obtained using two unwary transistors rudely awakened and plucked from the *Tran* drawers @ Elektor Labs are pictured in **Figure 6**.

(130544)

Web Links

[1] Platino: Elektor October 2011, www.elektor-magazine.com/100892

[2] SC Analyser2005: Elektor April 2005, www.elektor-magazine.com/030451

[3] Microcontroller BootCamp: Elektor April through November 2014. Part 1: www.elektor-magazine.com/120574

[4] Project software: www.elektor-magazine.com/130544

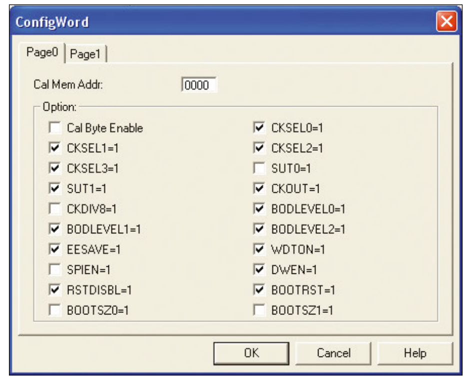


Figure 5. Fuse settings for the ATMega32 on the Platino board.

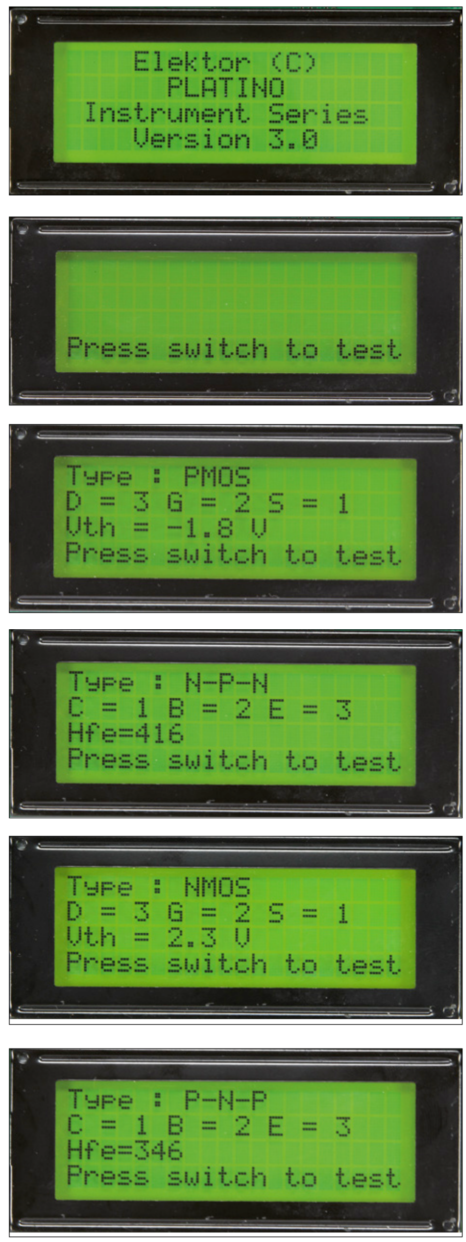


Figure 6. We grilled a bipolar trannie and a MOSFET, both "alien to us" we swear ... with plausible results.

CMOS IR Transmitter

Generate infrared signals the easy way

By Viacheslav Gromov (Germany)

Just about every control process can be handled elegantly with a microcontroller — even generating infrared signals. Sometimes there's an even easier method, using the time-honored CMOS logic. A case in point is the circuit offered here for an IR jamming transmitter, which can knock out (paralyze) popular remote controls using (not only) the widely used RC5 code protocol. An 'IR afterburner' makes longer range operation possible. And with an IR add-on for microcontrollers we can launch IR signals on their way with ease.

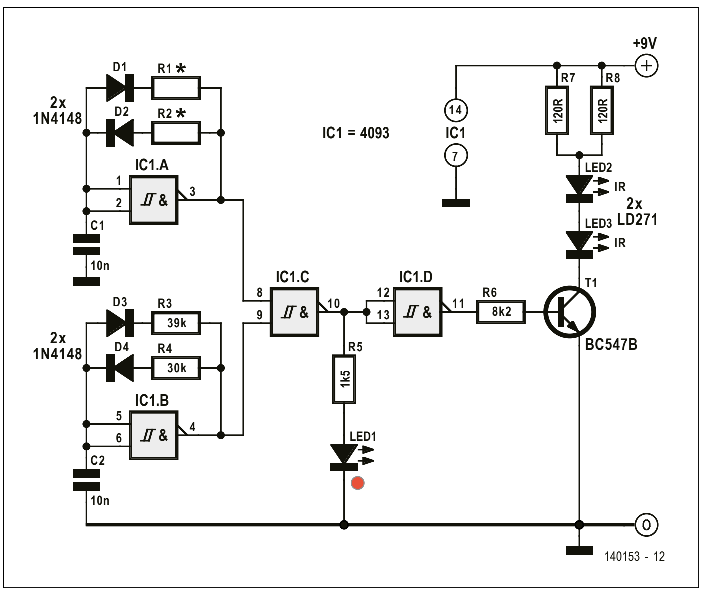


Figure 1. An IR jamming transmitter in CMOS technology.

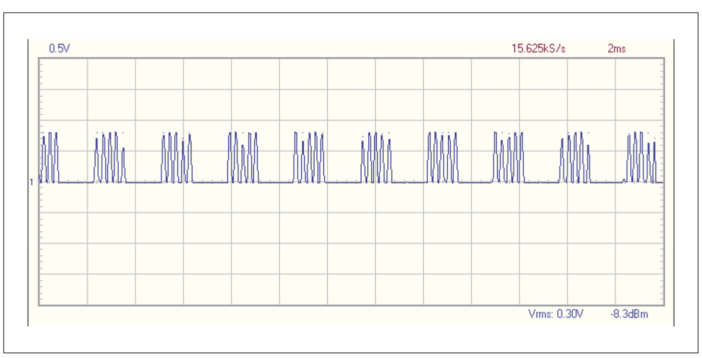


Figure 2. Oscilloscope screenshot of the modulated output signal (measured on the base of T1).

The RC5 code for IR remote control gadgets is straightforward and very widely used. The 14 bits provided per data word (Start, Toggle, Address and Command bits) are transferred at intervals of 113.778 ms (start of two data words). The length of a bit amounts to 1.778 ms. This data is modulated onto a squarewave carrier signal with a frequency of typically 36, 38, 40 or 42 kHz. In order, for instance, to jam the remote control of a TV, all you need do is modulate the relevant carrier frequency with a squarewave signal (frequency 1/1.778 ms ≈ 560 Hz). The receiver in the TV cannot filter the signal of the remote control zapper from the mixed IR messages arising now. Of course this assumes the jamming transmitter has sufficient IR power output and that's exactly what the circuit described here does. It can hold its own against an IR zapper up to a distance of about 3 m (10 feet) so far as transmit power is concerned.

The second circuit is the IR add-on for microcontrollers already mentioned. This produces a carrier signal that is controlled via an input port by a microcontroller (i.e. can be interrupted or blanked out). In this way the connected microcontroller can transmit any code of your choice (not only RC5), without needing a supplementary timer; this saves resources and code. A further option is an afterburner for extended range, which applies particularly high current pulses to the IR LEDs.

Jamming transmitter

The concept just mentioned for jamming data transfer of the RC5 IR kind can be implemented without further ado using a CMOS logic IC — a 4093 that contains four NAND gates with Schmitt trigger inputs. The circuit in **Figure 1** is very simple, comprising two independent R-C squarewave oscillators. The partial circuit involving IC1.A generates the carrier frequency, with the oscillator based on IC1.B producing the pseudo data in the form of a squarewave signal of around 560 Hz. Both oscillators use two diodes to compensate for differing resistances in charging and discharging the capacitor caused by unsymmetrical states of the hysteresis switching points of the Schmitt trigger inputs, resulting in a symmetrical pulse/pause relationship of about 50 %. Experience indicates this can be critical for many receiver modules. Both squarewave signals are mixed digitally in a third gate (IC1.C). The combined signal is also applied to LED1 to indicate correct operation. As this loads the output of IC1.C, the signal is regenerated in the remaining gate IC1.D. This 'cleaned up' squarewave signal is then passed to T1 via R6. The two IR transmit LEDs (LED2 and LED3) are connected in series via R7 and R8, driven with a peak current flow of around 100 mA. Because the duty cycle of 50 % * 50 % amounts to 25 %, the power dissipation at R7 + R8 is less than 200 mW. Consequently normal resistors of ¼W power rating are adequate for R7 and R8. The resulting transmit signal at the base of T1 should correspond to the oscillogram in **Figure 2**.

IR afterburner

If a greater transmit range — and consequently greater transmit power — is required, you can replace the IR final stage of Figure 1 using T1 with the IR LED driver circuitry of Figure **3**. T1 and T2 form a Darlington pair that will cause a healthy 500 mA of current to flow via R8 through the IR diodes. According to the data sheet [1] these LEDs can withstand a maximum of 130 mA standing current. With 500 mA and a duty cycle of 25 % they will not be pushed to their limits. In this set-up a 9V alkaline battery will be exhausted in four hours.

The oscillator for the MCU

Generating IR signals with microcontrollers is very straightforward but also resource-consuming, since a separate timer is necessary for timing the RC5 codes and another for producing the carrier frequency. On simple controllers this will 'use up' (fully occupy) both timers. However, producing the carrier frequency externally in hardware will free up at least one timer and avoids the need of code for the carrier frequency. **Figure 4** illustrates a supplementary circuit block of this kind, using a CMOS gate to generate the carrier. More precisely this is a sub-element of Figure 1. The oscillator for the carrier frequency is blanked out simply using T2. A High signal of 3.3 – 5 V at the input of the circuit disables production of the carrier frequency. IC1.B serves purely to invert the signal, as otherwise when the carrier is disabled, the current would flow continually through the transmitting LEDs, which will not do the 9-V battery supplying them any good and could also unsettle many an IR receiver.
Using a microcontroller means you can modulate the carrier frequency with any transmission protocol you like. If required, you can even construct two circuits of this kind with a single 4093 chip, since each uses only half the IC per circuit.

Carrier frequencies

Table 1 contains the individual values of R1 for commonly used carrier frequencies. Using these you can match the frequency of the transmitter to that of the receiver. R2 is generally 2.2 kΩ. The duty cycle varies somewhat in the process, according to the carrier frequency, but any minor deviation in the carrier from the desired 50 % is not of great significance. It's only with frequencies not in this table that you should select a correspondingly smaller or larger value of R2. What's significant is the tolerance of the components that determine the frequencies. In particular you should choose low-tolerance types for C1 (and C2 in Figure 1). Resistors R1 and R2 (together with R3 and R4 in Figure 1) must be of the metal film type.
Furthermore it's a good idea to check the carrier frequency with an oscilloscope or a frequency counter, to ensure that you have hit your desired frequency with reasonable accuracy. Most IR receivers are not particularly selective but even so, you should aim for a frequency divergence of below 1 kHz.

Implementation

The three circuits are so simple that you can build them without problem on a scrap of perf board. For once no SMD components are involved ;-).
When you power up the circuit in Figure 1 the red LED will tell you its status; if the red LED illuminates only weakly or not at all, there's something wrong with at least one of the two oscillators. In contrast, if it lights up bright and continuously, then

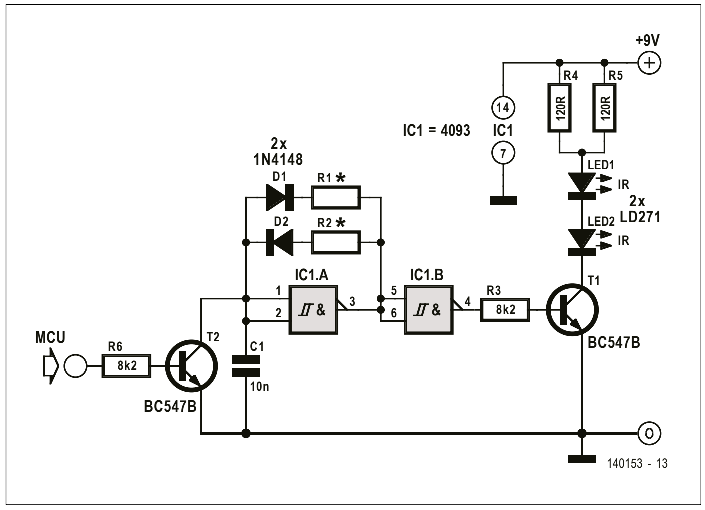


Figure 3. This driver circuit uses more current to produce greater IR transmit power. Caution: the LEDs and R8 get quite warm!

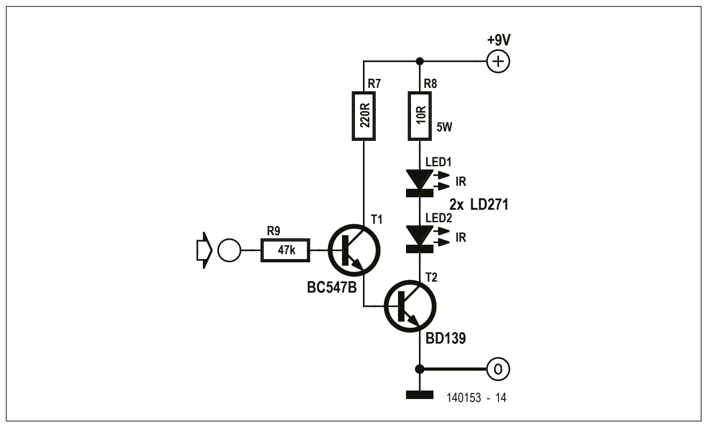


Figure 4. The IR add-on for microcontrollers enables them to transmit IR signals without sacrificing a timer for the carrier frequency.

everything is in order so far.
If you are unsure which carrier frequency the IR receiver in your TV uses, you can try out 36 kHz or 38 kHz first, as these are the two most widely used carrier frequencies. By the way, the circuit frequently works quite well even when the frequency is a full 2 kHz off what it should be.
If you have any comments or suggestions for improvements to this project please share them on Elektor-Labs.com [2].

(140153)

Web Links and Literature

[1] LD271 data sheet (Osram): http://goo.gl/vKDR2Y

[2] www.elektor-labs.com/node/3835 (in German)

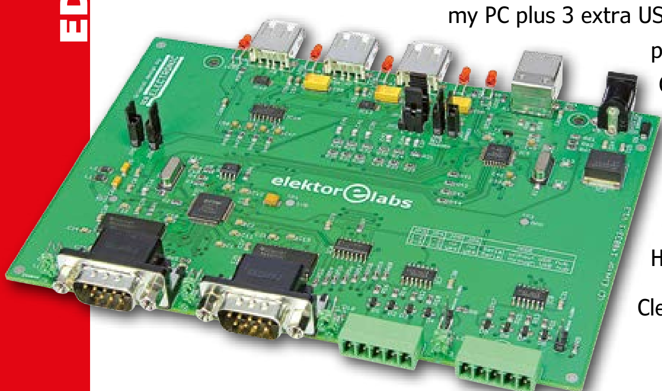
Table 1.
Frequency-dependent values for R1 when R2 = 2.2 kΩ

Carrier Frequency	Resistor R1
36 kHz	5.1 kΩ
38 kHz	4.7 kΩ
40 kHz	4.3 kΩ
42 kHz	3.9 kΩ

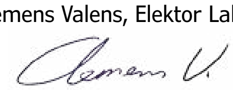
welcome in your ONLINE STORE

EDITOR'S CHOICE

At Elektor Labs I work with lots of different microcontrollers. The best way to communicate with these devices is still over Ye Olde Serial Port. Unfortunately, the computer I am using to write firmware has no serial ports, so typically I have to resort to USB-to-Serial converter cables. Often applications use more than one serial port and cable spaghetti builds up on my desk. Even sadder, my PC has insufficient USB ports. That's why the Elektor USB Hub with Legacy Serial Ports got developed. Now I have 4 (!) fully configurable serial ports (RS-232 and RS-485) on my PC plus 3 extra USB ports because it's a USB hub too. Only one cable between my PC and the project! If you too are tired of juggling with USB-to-Serial converter cables, do as I did, get yourself an Elektor USB Hub with Legacy Serial Ports.



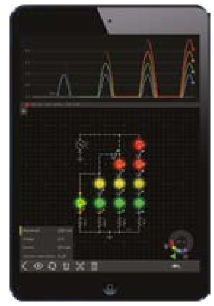
Clemens Valens, Elektor Labs



www.elektor.com/usb-hub

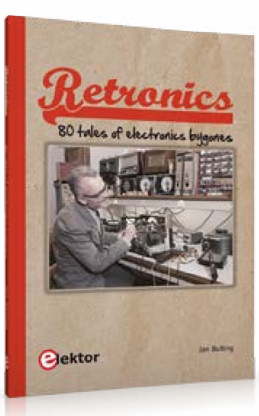
Elektor Bestsellers

1. EveryCircuit App
   www.elektor.com/everycircuit



2. DVD Elektor 1980 through 1989
   www.elektor.com/eighties
3. RPi Advanced Programming
   www.elektor.com/rpi-book
4. Advanced Control Robotics
   www.elektor.com/robotics-book
5. Raspberry Pi Model A+
   www.elektor.com/rpi-a-plus
6. DVD Elektor 2000 through 2009
   www.elektor.com/zeroes
7. Trilogy of Magnetics
   www.elektor.com/trilogy-magnetics
8. Android Breakout Board
   www.elektor.com/android-bob

Retronics

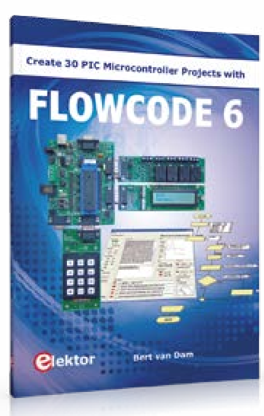


This book is a compilation of 80 Retronics installments published in Elektor magazine between 2004 and 2012. Since launching his Retronics columns, Jan Buiting has never been short of copy to print, or vintage equipment to marvel at. The stories cover vintage test equipment, prehistoric computers, long forgotten components, and Elektor blockbuster projects, all aiming to make engineers smile, sit up, drool, or experience a whiff of nostalgia.

member price: £23.95 • €26.96 • US $37.00

www.elektor.com/retronics

Create 30 PIC Microcontroller Projects with Flowcode 6

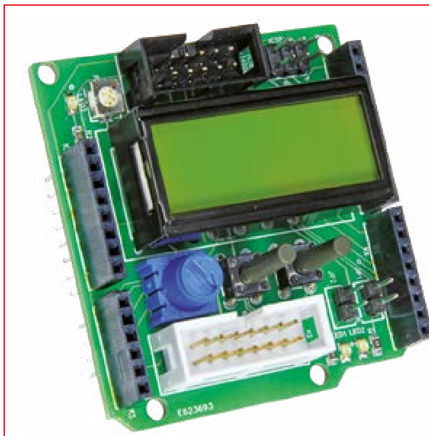


This book covers the use of Flowcode version 6, a state-of-the-art, all-graphical based code development tool, for the purpose of developing PIC microcontroller applications at speed and with unprecedented ease. Starting off with down-to-earth tutorial projects and step-by-step instructions, the book soon progresses and deepens with more ambitious projects. Without exception, the 30 projects in the book are fun to build and use!

member price: £27.95 • €31.46 • US $43.00

www.elektor.com/flowcode-6-book

Arduino Extension Shield



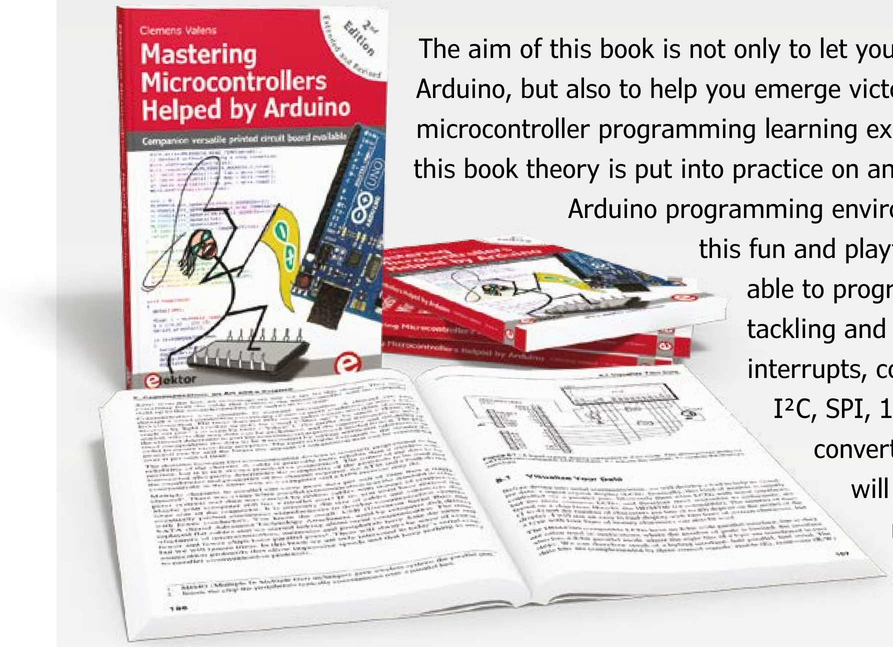
This shield, intended to augment the Arduino Uno, offers starters a text display, LEDs and pushbuttons to provide a good basis for their projects. For experienced users two extension connectors are provided which can be used to connect relay modules, wireless modules and many other devices. This shield thus ameliorates the Arduino Uno's most significant shortcomings on on-board peripherals.

member price: £23.95 • €26.46 • US $37.00

www.elektor.com/arduinoshield

Mastering Microcontrollers Helped by Arduino

Second, extended and revised edition. Extra chapter included!



The aim of this book is not only to let you enter the World of Arduino, but also to help you emerge victorious and continue your microcontroller programming learning experience by yourself. In this book theory is put into practice on an Arduino board using the Arduino programming environment. Having completed this fun and playful course, you will be able to program any microcontroller, tackling and mastering I/O, memory, interrupts, communication (serial, I²C, SPI, 1-wire, SMBus), A/D converter, and more. This book will be your first book about microcontrollers with a happy ending!

MEMBER PRICE: £31.95 • €36.50 • US $50.00

www.elektor.com/mastering-microcontrollers

Mastering Microcontrollers

Limited time offer for GREEN and GOLD members:
15% Discount plus free shipping!

20% OFF CDs and DVDs

For a limited time only Elektor offers a 20% discount on all CDs and DVDs!

J²B Synthesizer

An open-minded digital music platform

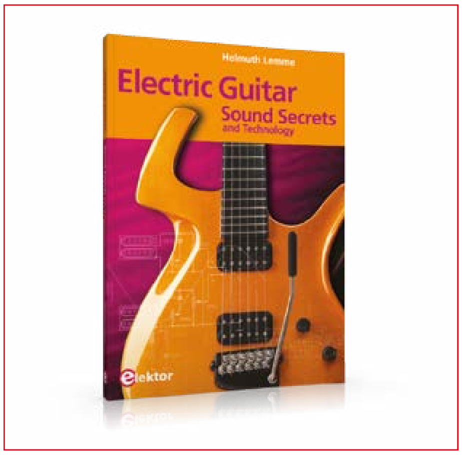
Intel Edison Breakout Board Kit

The Intel® Edison breakout board is designed to expose the native 1.8 V I/O of the Intel® Edison module. The board consists of power supply, battery recharger, USB OTG power switch, UART to USB bridge, USB OTG port, and I/O header. This kit consists of both the Intel® Edison breakout board and module but also includes two compute module screws and a regulatory information document.

member price: £70.95 • €80.96 • US $110.00

www.elektor.com/edison-bob

Electric Guitar



What would today's rock and pop music be without electric lead and bass guitars? These instruments have been setting the tone for more than sixty years. Their underlying sound is determined largely by their electrical components. But, how do they actually work? Almost no one is able to explain this to the true musician with no technical background. This book answers many questions simply, in an easily-understandable manner.

member price: £27.95 • €31.05 • US $42.00

www.elektor.com/electric-guitar

Raspberry Pi Maker Kit

If you already own a Raspberry Pi and have done your first electronic project then you are ready for this kit. It has all you need to bring 20 simple projects to life, including 62 parts and a 160 page handbook. You'll learn how to display your song titles on the LCD, how to build your own binary clock and much more. With the graphic programming language Scratch this all can be done in a short time. Don't worry. Programming skills are not needed.

member price: £69.95 • €79.95 • US $108.00

www.elektor.com/rpi-maker-kit

REVIEW OF THE MONTH

BY WIGGERT PEERDEMAN

I have purchased various products from the Elektor webshop, including a bat detector, several issues of the magazine as downloads and a trial membership, but the most useful of all in my opinion is the TV Simulator. It is also what I use the most. I like to go on holiday, and because my home is left unattended when I am away I was looking for something that would give the impression that someone was at home. That is why I bought the TV Simulator from Elektor. It proved to be a good product and very easy to build. The SMD parts were already mounted on the PCB, so I only had to mount the LEDs and a few wired components on the bottom of the PCB. After that it didn't take long to fit the board in a suitable case and fold the small piece of transparent paper in place, and the simulator was ready to use. After switching it on I naturally tried it out in the room with the television set, to see what it looked like from outside. It really is just like having the TV on! All in all, this device has come in handy for me several times already. It's a real boon for the holiday period and for evenings when there is nobody at home.

Read this review and more about this product at www.elektor.com/tv-simulator

When you send us a review of your favorite Elektor product, you have a chance of winning a 100-euro voucher that can be redeemed in the Elektor Shop.

Like to know more? Check out www.elektor.com/rotm

If you want to complete your collection of annual or decade DVDs or just want to get one of the Audio Masterclass DVDs from our experts such as Jan Didden or Menno van der Veen, we have a special offer for you in March in our shop.

Visit the CD/DVD section at Elektor.com for a an overview of our full collection. Your discount is directly visible.

More good news: this special offer is not limited to the recently released 2014 Annual DVD, but also covers all magazine volumes from previous years released on CD-ROM.

Note: This special offer is only valid until March 31, 2015 and is subject to availability.

www.elektor.com/cds-dvds

Mastering Microcontrollers

Limited time offer for GREEN and GOLD members:
15% Discount plus free shipping!

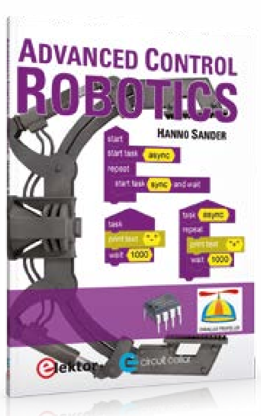
20% OFF CDs and DVDs

For a limited time only Elektor offers a 20% discount on all CDs and DVDs!

J²B Synthesizer

An open-minded digital music platform

Advanced Control Robotics



Hanno Sander's Advanced Control Robotics simplifies the theory and best practices of advanced robot technologies. You're taught basic embedded design theory and presented handy code samples, essential schematics, and valuable design tips (from construction to debugging). The book is intended to help roboticists of various skill levels take their designs to the next level with microcontrollers and the know-how to implement them effectively.

member price: £31.95 • €35.96 • US $49.00

www.elektor.com/robotics-book

Mystery Product

We dare you ...

www.elektor.com/mystery

Raspberry Pi Model A+

The Model A+ is the low-cost variant of the Raspberry Pi. Model A+ uses the BCM2835 application processor and has 256MB RAM, but it is significantly smaller (65mm in length), consumes less power, and inherits the many improvements that we made to the Model B+: more GPIO, micro SD and better audio. We recommend the Model A+ for for embedded projects and projects which require very low power, and which do not require Ethernet or multiple USB ports.

member price: £17.95 • €20.25 • US $28.00

www.elektor.com/rpi-a-plus

THIS MONTH'S SHOPPING LIST

- **BL600-eBoB**
  PCB: 140270-1
  Module: 140270-91
- **Guitar Overdrive**
  PCB: 130311-1
- **Radio Controlled Multi-Switch**
  PCB: 140088-1
- **Experimenter's Transistor Tester**
  PCB: 130544-1
- **Mole Repellent**
  PCB: 140078-1
  Progr. microcontroller: 140078-41
- **Elektor ZigTexter Network**
  PCB: 130556-1
  Progr. microcontroller: 130556-41
- **Ultrasonic Tape Measure**
  PCB: 130546-1
  Progr. microcontroller: 130546-41

Get it all @ www.elektor.com/out-now

elektor•post : DON'T MISS OUT!

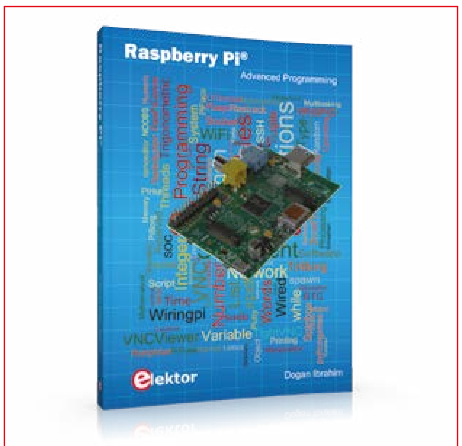
Over 100,000 people worldwide already receive our free Elektor newsletter in their inbox every Friday morning. It is packed full with the latest news items, tips and trends from the world of electronics. If you do not already have a .POST membership, sign up now and receive these extras:

- **A free Elektor project every second week**
- **Special offers in the Elektor Store**
- **A 5 Euro discount on your next purchase from Elektor**

REGISTER TODAY, IT'S FREE

www.elektor.com/newsletter

Raspberry Pi Advanced Programming

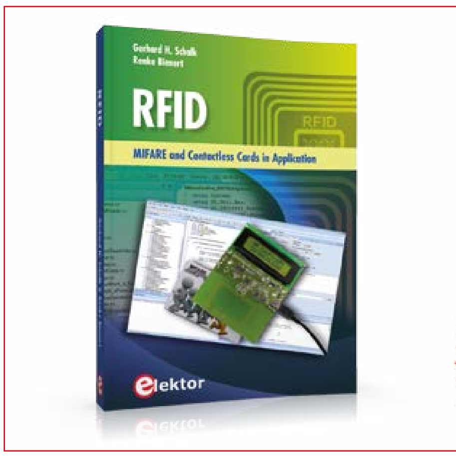


This book is about advanced programming of the Raspberry Pi computer using the Python programming language. The book explains in simple terms and with examples: how to configure the Raspberry Pi computer; how to install and use the Linux operating system and the desktop; how to write advanced programs using the Python programming language; how to use graphics in our programs and how to develop hardware based projects using the Raspberry Pi.

member price: £35.95 • €40.46 • US $55.00

www.elektor.com/rpi-book

RFID – MIFARE and Contactless Cards in Application



MIFARE is the most widely used RFID technology, and this book provides a practical and comprehensive introduction to it. Among other things, the initial chapters cover physical fundamentals, relevant standards, RFID antenna design, security considerations and cryptography. What's more, the complete design of a reader's hardware and software is described in detail. Finally, the major smart card reader API standards are introduced.

member price: £39.95 • €44.91 • US $61.00

www.elektor.com/rfid

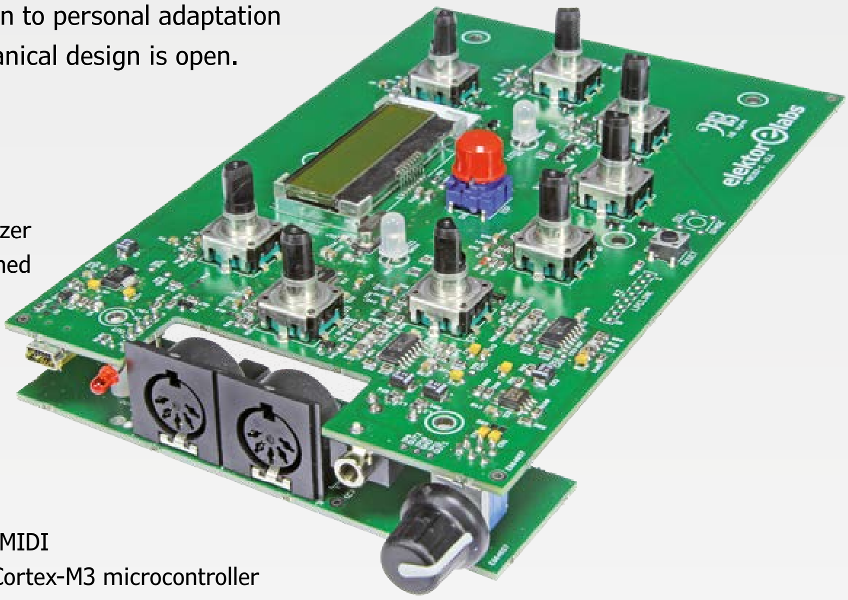
MICROCHIP DM160218 HILLSTAR DEVELOPMENT KIT

This exclusive product bundle consists of the MGC3130 Hillstar Single Zone Development Kit, and the 3D Touchpad. The dev kit in the bundle serves the microcontroller fans among you, the 3D Touchpad, those of you into PC programming. By special arrangement with Microchip the price of the bundle is 43% lower than that of the two individual parts from Microchip direct.

member price: £97.95 • €112.50 • US $152.00

www.elektor.com/microchip-technology-inc.

The J²B Synthesizer employs a powerful 32-bit Cortex M3 processor to produce two channels of sound. The novel sound engine offers a large choice of filter and distortion algorithms resulting in a unique sound with a bite.
The platform is 100% open to personal adaptation and taste. Even the mechanical design is open.
Headphones on!



Specifications

- Monophonic 9-bit synthesizer
- 32 waveforms + user defined
- 15 filter types
- 2 envelope generators
- LFO with 16 waveforms
- 15-pattern arpeggiator
- 16 patch memories
- 6 live controls
- MIDI
- Patch saving/loading over MIDI
- NXP LPC1347 32-bit ARM Cortex-M3 microcontroller
- 2 output channels
- Open Source & Open Hardware design

MEMBER PRICE: £97.95 • €112.46 • US $152.00
www.elektor.com/j2b-synthesizer

Mastering Microcontrollers

Limited time offer for GREEN and GOLD members:
15% Discount plus free shipping!

20% OFF CDs and DVDs

For a limited time only Elektor offers a 20% discount on all CDs and DVDs!

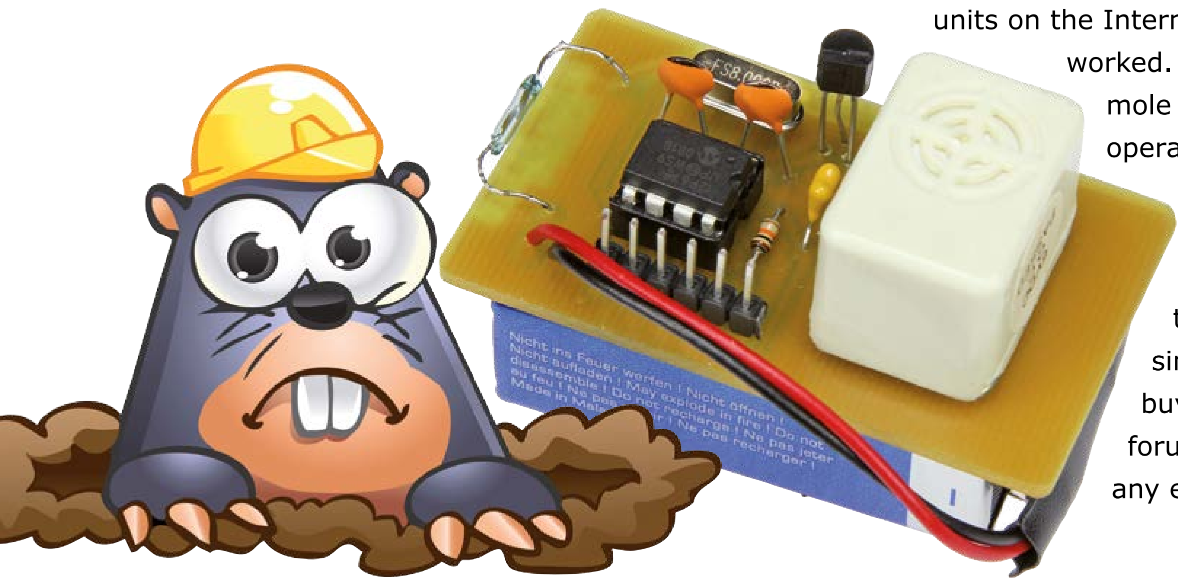
J²B Synthesizer

An open-minded digital music platform

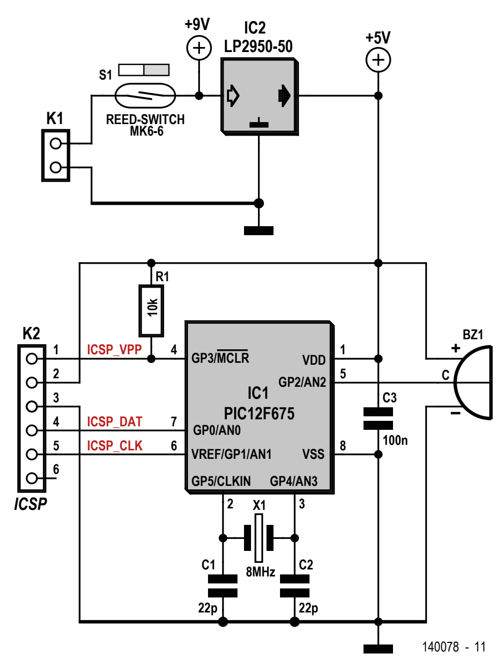
Mole Repellent

By **Jaap van Rijswijk** (South Africa)

A while ago, the author was approached by an acquaintance asking whether he could build an electronic device to repel those moles that were constantly digging up the lawn. Since he didn't know much about the behavior of moles, he first searched for some commercial units on the Internet to find out how they worked. Most of the 'electronic' mole repellents appear to operate by producing a 400-Hz signal generated at 30 to 40 second intervals. However, he didn't have that much faith in these, since about half of the buyers reported in various forums that they didn't have any effect.



To start off, a number of anti-mole circuits from the Internet were examined. Most of these were designed around the old faithful 555 timer and the author really wanted to make something more contemporary, with low energy consumption, so that the circuit could last for several months on a battery. After some deliberations, he chose a design that includes a PIC microcontroller, which drives an active three-wire mechanical buzzer. The built-in electronics of the buzzer produces a tone of about 400 Hz with reasonable sound pressure, perfectly suited for this application. His choice for driving this buzzer was a PIC12F675, which he happened to have on hand.



The program written for the PIC ensures that the buzzer is turned on for about 1 second every 40 seconds. That should be sufficient to repel the moles in the vicinity of the circuit. Using a 9-V battery (capacity about 550 mAh) the circuit should be able to operate for nearly two months.

The schematic is quickly explained. In addition to the PIC, buzzer and voltage regulator, it contains only a few other components. Quartz crystal X1, together with C1 and C2, provide the clock frequency of 8 MHz. The internal oscillator in the PIC is not used here because its current consumption is too high for this application. Resistor R1 ensures that the reset connection (MCLR) of the PIC is pulled High after the power supply is turned on. The control line of the buzzer is connected to pin 5 (GP2) of the controller. Finally there is a decoupling capacitor connected across the power supply lines to the PIC. A 6-pin header is present for in-circuit programming of the PIC using, for example, a PICkit3 programmer.

The 5-V supply voltage for the PIC comes from a frugal voltage regulator type LP2950, which has a quiescent current of only

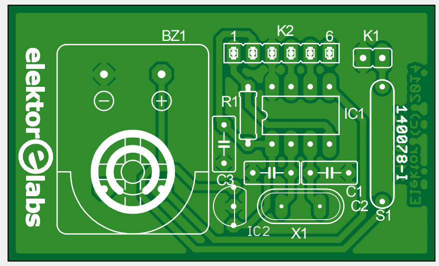
Component List

Resistor
R1 = 10kΩ

Capacitors
C1,C2 = 22pF
C3 = 100nF

Semiconductors
IC1 = PIC12F675-I/P, programmed, Elektor Store # 140078-41
IC2 = LP2950-50

Miscellaneous
X1 = 8MHz quartz crystal
K1 = 2-pin pinheader, 0.1” pitch
K2 = 6-pin pinheader, 0.1” pitch
S1 = reed switch, normally open
BZ1 = 3-pin active buzzer, 400 Hz, e.g. EMS-06-4TP, Reichelt # SUMMER 6V P
9-V-battery with clip-on leads
Small magnet
PCB # 140078-1, see [1]



In various forums half of the buyers of ‘ready made’ electronic mole repellents reported that they didn’t have any effect.

75 µA. The power supply to the mole repellent is switched on with the aid of a reed relay (S1). This has been done so that the circuit can be built into a watertight enclosure, which can then be put into the ground (for example a jam jar with screw lid). After putting the jar in place, the circuit can be turned on by a taping a small magnet on the outside of the jar in the vicinity of the reed relay.

The operation of the circuit is as follows. After powering on, the buzzer will be switched on for 1 s. After that there is a pause of 2.5 minutes, which gives the user time to insert the jar into the ground. After that, the buzzer will sound every 40 seconds producing a 400-Hz tone for nearly a second.

The program uses the built-in watchdog timer (WDT) to briefly wake up the PIC12F every 2.3 s form its sleep state (current consumption about 150 µA) using a timeout-interrupt. The WDT has been configured with a 1:128 prescaler, so that the timeout will occur every 2.3 seconds. A counter keeps track of the number of interrupts that have been generated by the WDT. After 16 interrupts the output GPA2 (pin 5) is pulled High for 800 ms.

A small circuit board was designed for the circuit, which contains all the components including the buzzer. We don’t need to say much about its assembly as only through-hole components have been used. The reed relay is fitted a little above the circuit board, so that it can be bent towards the outside and ends up near the edge of the board.

The microcontroller is available ready-programmed from Elektor Store, but you can, of course, program it yourself as well (software and PCB layout are available as free downloads from [1]).

As already noted, it is best if the circuit and battery are fitted together inside a jar made from glass or plastic that can be closed tightly. The reed relay has to be close to the outside so that it will react immediately when a magnet is held near it from the outside of the jar. By the way, you could also use a reed relay which has normally-closed contacts. You then stick a magnet to the outside of the jar when you are not using the circuit.

It is best to bury the mole repellent in a tunnel, for example in a molehill. But don’t forget where you have buried it, otherwise it will be hard to replace the battery in due course.

And the result? After three months the acquaintance phoned to say that it actually worked and the moles had abandoned their work area! And that for only 10 dollars worth of parts.

(140078-I)

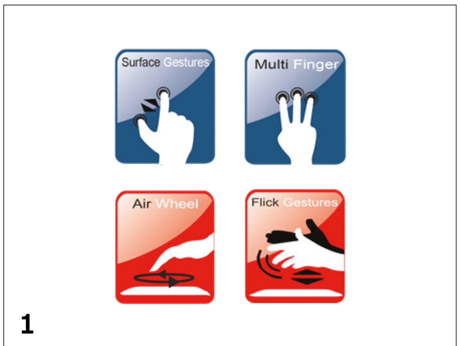
Web Link

[1] www.elektor-magazine.com/140078

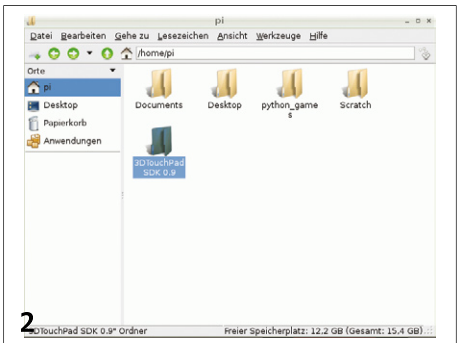
GestIC & 3D Touchpad Workbook (3)

Explore the RPi with the 3D Touchpad

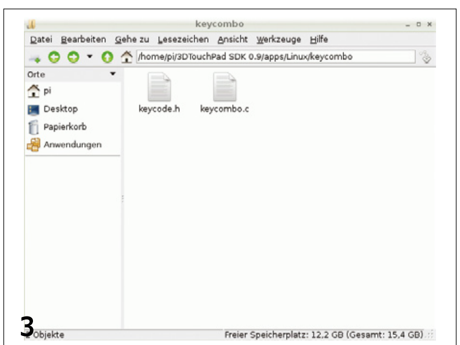
By **Thomas Lindner** and **Hung Nguyen** (Microchip GestIC® Team, Germany), also feat. **Tux the Penguin**



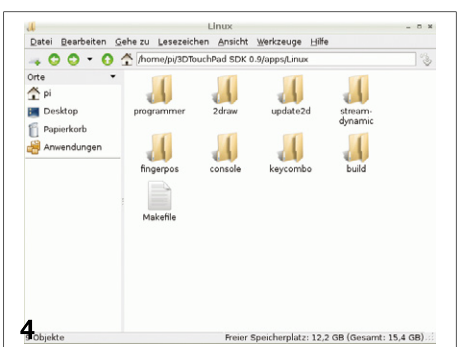
1



2



3



4

In the third and final installment we explore the ready-made 3D Touchpad in conjunction with the RPi minicomputer. We install the Software Development Kit (SDK) and finally get back to playing the '2048' game — now with a tuned 3D Touchpad.

Connect 3D Touchpad, start exploring the RPi

The 3D Touchpad in the product bundle exclusively available from Elektor [1] combines 2D and 3D functions in one PC peripheral device. It is recognized by the PC as a standard HID device and can be used without the installation of a driver. Supported operating systems (OS) are Windows 7/8, MAC OS, and several Linux distributions.
In our case we use the Raspbian OS which has the necessary HID stack integrated.

Let's go!

Connect the 3D Touchpad to the RPi's USB port and use it as a standard mouse-ish device. You can track the mouse pointer and simulate a left click by tapping on the surface. A two finger tap equals right-clicking, a double tap starts applications — but there are extras to discover, like these (also pictured in **Figure 1**):

- using two fingers to scroll (e.g. in the browser);
- use pinch gestures to zoom (e.g. in a maps application);
- Tap & slide to drag and drop files or to mark text in a document.

Moreover, there are 3D gestures included:

- flicking left/right to simulate the left/right arrow key;
- using AirWheel for continuous 3D scrolling (e.g. in the browser);
- using a Double Flick gesture from North to South to close applications (not in Linux).

We want to use the 3D touchpad to play the 2048 game and therefore we need to tune its functionality and map all four flicking gestures to the arrow keys.

Almost forgot... the Windows PC

In terms of software the 3D Touchpad is supported by the 3DTouchPad GUI and the 3DTouchpad Software Development Kit (SDK). Both suites can be downloaded as Windows installers (Win7/8) [2] and will unpack the software to a Windows folder.

The **3DTouchPad GUI** is a Windows program; it visualizes 2D/3D signals and allows the 3DTouchpad firmware to be updated.
The **3DTouchPad SDK** provides a framework to control the 3DTouchPad by an API interface and includes example applications for Windows and Linux.
The folder structure for the installed SDK is like this:

3DTouchPad SDK 0.9
→ api
→ apps
→ doc
→ utilities
→ readme.html

We copy the complete folder to the home directory of our Raspberry Pi, as pictured in **Figure 2**.

Build the Application examples

If we step through the folders, we see a Linux section under the 'apps' folder (**Figure 3**). That is what we are interested in. It contains a predefined Make file (**Figure 4**) and with the right Linux packages the complete apps can be built. To install the necessary packages on the RPi, make sure that it has an Internet connection and then type following lines in the Terminal:

5

```
sudo apt-get install build-essential
libncurses5-dev
sudo apt-get install libudev-dev
```

Then change the directory to apps/Linux and run “make”.

```
cd apps/Linux
make
```

The screen should look like in **Figure 5**. You may continue as `root` but we propose to grant all users read and write access to 3DTouchPad device files. The necessary command lines look like this (the full set is at [3]):

```
sudo sh -c ‘echo
   “SUBSYSTEMS==\”usb\”,
   ATTRS{idVendor}==\”04d8\”,
   ATTRS{idProduct}==\”09d3\”,
   SYMLINK+=\”hmi2d\”,
   GROUP=\”plugdev\”, MODE=\”0666\””
   >> /etc/udev/rules.d/99-hmi.
   rules’
```

To refresh the new rules restart the Linux system and disconnect and reconnect the 3DTouchPad to the USB port to make sure the settings are updated.
Now you can execute the 3DTouchPad example applications in then following directory:

```
cd apps/Linux/build/bin
./nameofapp
```

Try them out – they are described in the html documentation with the SDK package. In our *Workbook* we continue with the keycombo app.

Play 2048! Or anything else!

As said in the beginning, the 2048 game requires the mapping of Flick gestures to all four arrow keys. That can be done by using the *keycombo* app shown in action in **Figure 6**.
To map flicks only to the arrow keys, modify the `key_combo_map` at lines 35-54 in the file “keycombo.c” (for more commands refer to “keycode.h”), thus:

```
key_combo_entry_t key_combo_map[] = {
    { hmi2d_param_outkeyFlickL_send, { hmi2d_event_on_single,
                                       key_left } },
    { hmi2d_param_outkeyFlickR_send, { hmi2d_event_on_single,
                                       key_right } },
    { hmi2d_param_outkeyFlickU_send, { hmi2d_event_on_single,
                                       key_up } },
    { hmi2d_param_outkeyFlickD_send, { hmi2d_event_on_single,
                                       key_down } },
    { hmi2d_param_outkeySwipe1FL_send, { hmi2d_event_on_single,
                                         0 } },
    { hmi2d_param_outkeySwipe1FR_send, { hmi2d_event_on_single,
                                         0} },
    { hmi2d_param_outkeySwipe1FU_send, { hmi2d_event_on_single,
                                         0 } },
    { hmi2d_param_outkeySwipe1FD_send, { hmi2d_event_on_single,
                                         0 } },
    { hmi2d_param_outkeyApproach_send, { hmi2d_event_on_single,
                                         0 } }
};
```

keycombo.c can be downloaded at [3]. To compile the app go back to the console, change to the “\apps\Linux” directory and enter “make” to build the apps again. Start the app in “\apps\Linux\build\bin” like so:

```
./keycombo
```

Now run the 2048 game from the previous installment — it can be played with the GPIO inputs or with arrow keys. And thanks to our new gesture mapping, we can play it with 3D hand gestures. It’s embedded choreography, **Figure 7** tells the story.
Alternatively, run any other application which can be controlled be arrow keys and from now on happily use it with 3D gestures.
When finished go back to the console and enter “quit” (by typing! Duh!). The 3D TouchPad “standard” mapping will be restored.

Conclusion

Having explored and successfully implemented full-blown touch and gesture control on a microcontroller system (RPi) running a nerdy game (2048) it’s time to close our GestIC Workbook. The small team is grateful for all feedback received along the journey and calls on all owners of the Elektor/Microchip Hillstar product bundle to report on their projects and share relevant information.

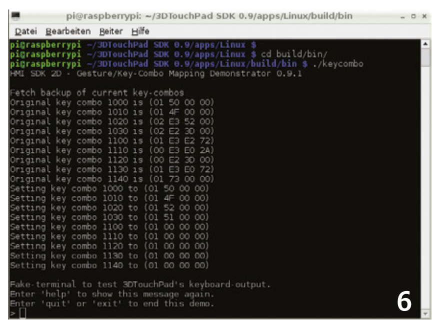
(140536)

Web Links

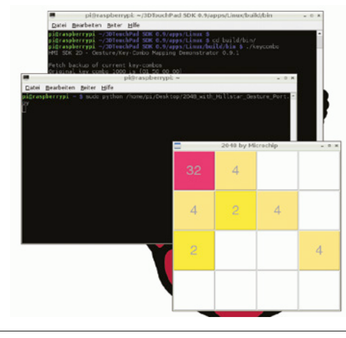
[1] Microchip Hillstar GestIC dev kit and 3D Touchpad product bundle: www.elektor.com/hillstar

[2] 3D TouchPad GUI and SDK: www.microchip.com/3DTouchPad

[3] Keycombo C code; Grant Write Access command file: www.elektor-magazine.com/140536



6



7

Elektor ZigTexter Network

Short-messaging over Zigbee

Specifications

- Zigbee based point-to-multipoint short messaging system
- 4x40 liquid crystal display
- Supply voltage: 8 to 12 V DC
- ATtiny2313 microcontroller software written in C
- Transmitter software for Windows written in C#
- Instant messaging
- Message scheduling

By **Elektor Labs India, Luc Lemmens** & **Clemens Valens** (Elektor.Labs)

Here's a flexible system to broadcast text messages wirelessly to family members, neighbors, clients, visitors, colleagues, lazy staff, head cooks and bottle washers — all via Zigbee, with a handy message scheduling function built in.

Broadcasting is the art of transmitting a message to many receivers at once. TV and radio stations have a broadcast permit to disseminate their program material to as large an audience as possible. Broadcasting in a more technical sense is also widely used on computer networks, for instance to announce status information that all nodes need to know. The antonym is *narrowcasting*.

Imagine you run a restaurant where you serve a Chef's Special. You'd prepare a limited number of servings, enough for, say, a hundred clients, before opening your restaurant to let the hungry crowd in. Instead of writing the Chef's Special on a blackboard where only a few people will (or can) see it, why not install a display on every table to show what's for lunch. You could then enhance these displays with a counter to show the number of Chef's Specials Basil Fawlty or the Swedish Chef can cook before running out of ingredients (*supplier-induced stock anxiety*). Not only would it save you and the serving staff a lot of hassle, it's also a nice gimmick that clients may spread the word on. That's just one example of a case where a simple text message broadcasting system (textcaster) can be useful, and we are sure you can think of many others.

No wires

Building such a text message broadcasting — or point-to-multipoint — system is not overly complicated thanks to the multitude of cheap wireless modules on the market today. All you need is one transmitter and a bunch of receivers listening to the same radio channel. The transmit power, modulation and frequency band should be legal to use in the country where the system is deployed, while the protocol is not too important. If the transmitter and receiver modules can form a wireless serial connection, then they are good to go. In this project we use Zigbee modules operating in the 2.4 GHz range, but 868/915 MHz radio modems (wireless UARTs) for instance should work equally well.

Zigbee, XBee, ZB, ItripleE

A few years ago Zigbee was hyped and then faded away quietly. That's not to say Zigbee is dead — on the contrary, it found its way into many Smart Metering and Home Automation applications. Practically all major microcontroller manufacturers support Zigbee. Using Zigbee is interesting for several reasons:

- low power; quintessential for battery operated devices;
- mesh networking capabilities, allows for intelligent networking and extended range;
- robust protocol for reliable communication;
- global approval saves paperwork.

The radio module chosen for our project, the XB24-Z7WIT-004 is a member of the popular XBee ZB family from Digi. The family offers a bunch of fine features but we decided against using all of them. However, future expansion is always lurking so you might as well prepare for it.

Therefore, in this project we did not connect the four ADC inputs to any sensor, nor did we connect the I/O pins to LEDs or relays. That's not to say though we forbid you to do so if you please.
Another advantage of using an XBee module is that it can be replaced by a pin-compatible member to extend the range of the network if necessary. For us the specified 40-m (120-ft.) indoor range or 120-m (350-ft.) outdoor range (line-of-sight) was more than sufficient. However, if you decide to replace the specified Zigbee module by some other XBee module, be aware that they are not all compatible with each other. There are two families, XBee (Series 1) and XBee ZB (Series 2 and 2B), that implement slightly different communication stacks. The Series 1 modules talk IEEE 802.15.4 which is similar but not identical to Zigbee which is an extension of IEEE 802.15.4.
Another subtlety that you may run into is that Zigbee does not necessarily operate in the 2.4 GHz ISM band, but may also use other ISM bands like 868 MHz or 915 MHz. So, before ordering modules to integrate in your existing Zigbee network, check the frequency they are on.
One last fact to know about Zigbee before we dive into the circuit diagrams and software of our ZigTexter Network: you need at least two modules. As meek as the point may seem, you will need a so-called coordinator and one or more routers or end points. Luckily the XBee module can play all these roles so although you don't need different modules, you should be aware of the fact that this project requires **two different XBee configuration files** (profiles) to make it all work together. You got cautioned, so please don't send us questions about it.

System overview

The system overview in **Figure 1** shows one transmitter (Tx) serving three receivers (Rx), but you can have as many Rx's as you like. Not infinitely though, but a heap! The more nodes you have the more traffic your network has to support due to protocol overhead. Obviously that puts a practical limit to the maximum number of nodes. It is hard to give an exact figure, but 1,000 nodes should not be a problem., and enough for even the largest restaurants or your bank account, whichever has precedence.

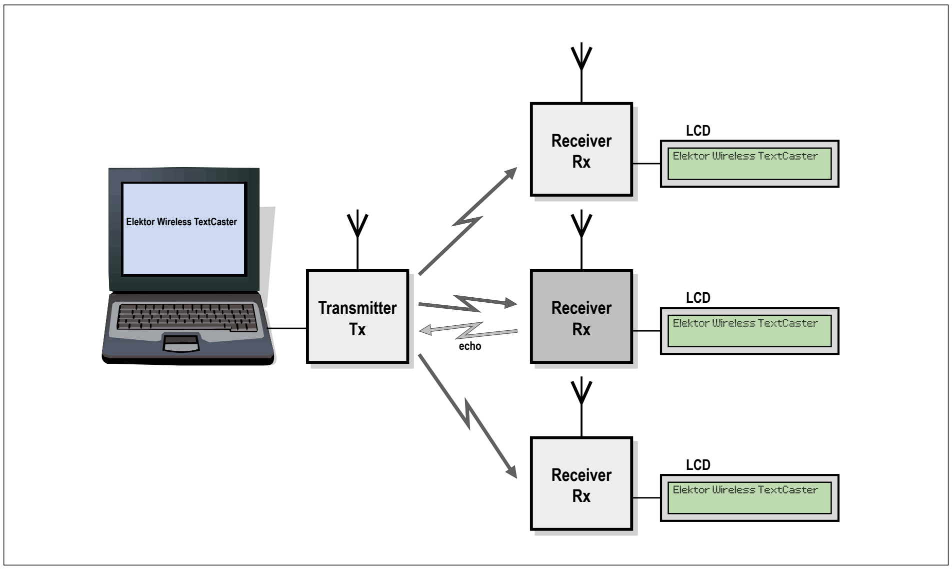


Figure 1. System overview of the Elektor Wireless TextCaster. Note the single echoing receiver (Rx) node in the middle.

In Figure 1 one node has a different color as well as an arrow labeled "echo" pointing back to the transmitter. This node echoes the received data to the transmitter allowing the user to verify that at least one node received the message. Install this node in the worst spot as far as signal reception is concerned. If it can "hear" your broadcasts, then the other nodes can too.

The Elektor ZigTexter Network supports one transmitter only and one echoing receiver only, irrespective of the number of nodes. In fact the echoing receiver can be left out at the cost of not getting any acknowledgement from your network. But it won't save any money as this node is identical to the others except for a jumper.

Receiver

That very jumper is JP1 in **Figure 2**, the schematic of the Receiver node (Rx). Fit the jumper on pins 2 and 3 for an echoing receiver and on pins 1 and 2 for a non-echoing receiver. Microcontroller IC1 will read the level on jumper pin 2

ZigBee and Wireless Radio Frequency (friendly) Coexistence

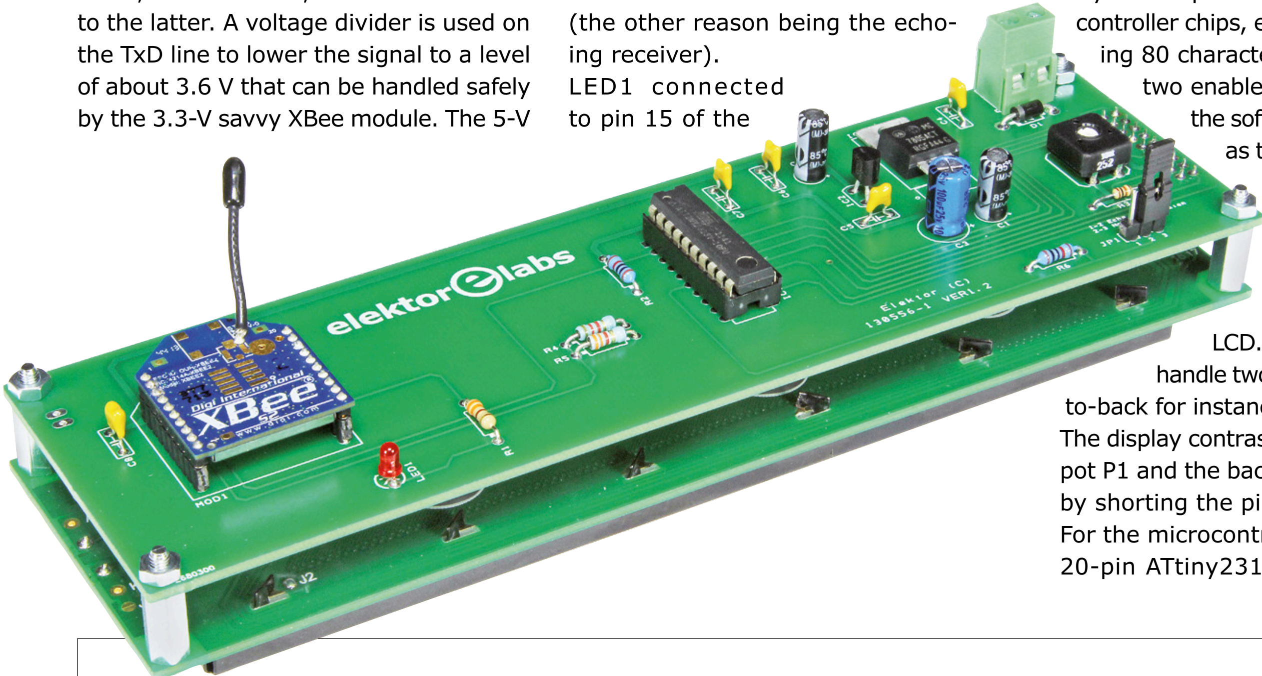
As you probably know, the 2.4 GHz band is also used by (to name a few) Wi-Fi networks (IEEE 802.11) and Bluetooth connections (IEEE 802.15.1). All these technologies are part of the IEEE 802 standards family supervised by the IEEE 802 group. From the Zigbee Alliance documentation:

"The IEEE 802 group continually evaluates its standards to identify areas of ambiguity or concern and works to improve its standards to ensure robustness and long-term success. To be approved as an IEEE 802 standard, IEEE 802 wireless standards must develop a Coexistence Assurance Document and implement a plan as part of the standard that ensures that all 802 wireless standards can operate and coexist in the same space."

Of course this is no guarantee that everything will always work perfectly fine when setting up wireless networks, but it allows for some optimism.

Source: https://docs.zigbee.org/zigbee-docs/dcn/07/docs-07-5219-02-0mcg-zigbee-and-wireless-radio-frequency-coexistence.pdf

at start-up, and configure its operating mode accordingly.
The microcontroller has one side connected to an LCD for displaying the received text messages and the other side to the XBee Zigbee module. Only two wires, RxD and TxD, are needed to talk to the latter. A voltage divider is used on the TxD line to lower the signal to a level of about 3.6 V that can be handled safely by the 3.3-V savvy XBee module. The 5-V powered microcontroller has no problems with the lowish level of the RxD signal. Besides receiving data, the Rx node also has to control the XBee module, notably at power-up. That's one of the reasons for the TxD and RxD signals traveling between the MCU and the module (the other reason being the echoing receiver).
LED1 connected to pin 15 of the XBee module provides information about the network. It is supposed to blink at 1 Hz or so when it has successfully joined a network. It will be on constantly if it hasn't.
The LCD used can display four lines of 40 characters (i.e. 160 characters). It's actually made up of two HD44780-compatible controller chips, each capable of displaying 80 characters, which explains the two enable lines EN1 and EN2. In the software the LCD is treated as two separate 2 x 40 displays. Consequently, if you insist, you could connect two 80 character displays instead of one 160 character LCD. One receiver thus can handle two smaller displays, back-to-back for instance.
The display contrast is adjusted with trimpot P1 and the backlight can be activated by shorting the pins of K2.
For the microcontroller we opted for the 20-pin ATtiny2313. A larger controller



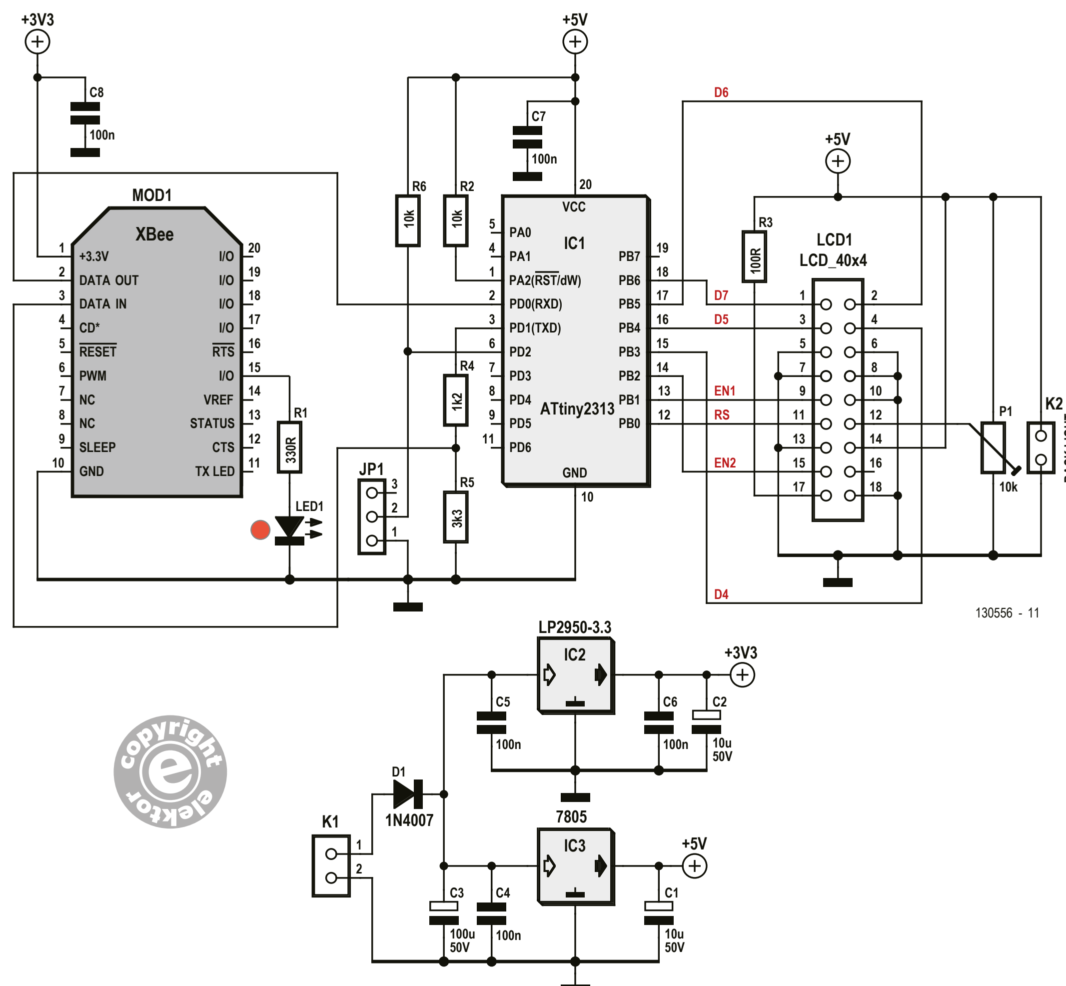
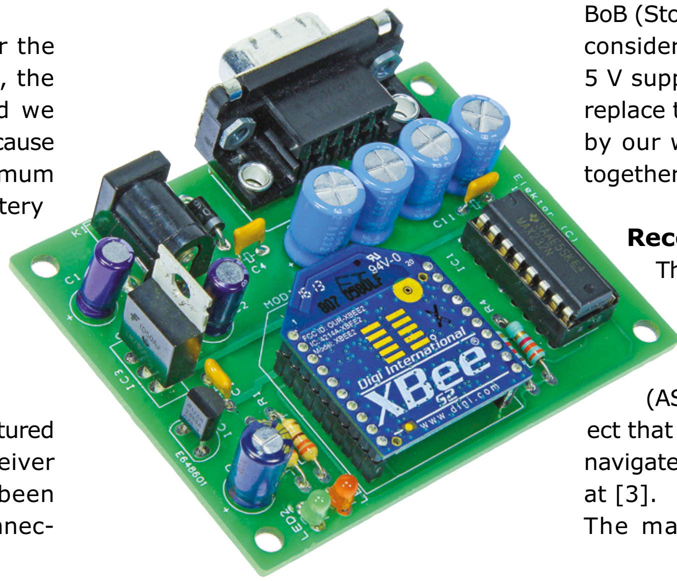


Figure 2. Schematic of a receiver node (Rx).

in terms of size or processing power, is superfluous really. Smaller is possible as long as enough I/O ports are available. Contrary to the smaller ATtiny's around the 2313 has an integrated UART peripheral, which keeps programming life simple. It happily runs from its internal oscillator making an external crystal unnecessary.
The receiver node needs 3.3 V for the XBee module and 5 V for the LCD, the microcontroller accepts both and we chose to wire it to the 5 V supply. Because of the 7805 (IC3) and D1, the minimum supply voltage is 8 V. A 9-V dry battery works fine. If you manage to find a 3.3 V LCD then the whole unit can be powered from a 5 V phone charger from the thrift shop.

Transmitter

The transmitter (Tx) schematic is pictured in **Figure 3**. Compared to the receiver one could say that the LCD has been replaced by K2, a 9-pin sub-D connector (DE-9, not DB-9 as it is commonly called) and that the microcontroller got replaced by an RS-232 level converter (IC1) wired up as usual. See the **RSSI inset** on the function of LED1.
The transmitter's power supply section is identical to that in the receiver. Here the 5 V is only needed to power the good old MAX232N. If you replace it by a MAX-3232EEPE+ in a 16-pin DIP package you can power the whole circuit from 3.3 V. In case you are thinking of replacing IC1 by a USB-serial converter cable, make sure that all the signals (including the supply pin!) use 3.3 V levels. The Elektor Serial BoB (Store item 110553) is a good choice considering the 3.3-V FTDI cable has a 5 V supply pin! Yet another option is to replace the transmitter board altogether by our wireless T-board no. 140374-1 together with a 3.3-V FTDI cable.



Receiver software

The firmware for the receiver module was written in C as an Atmel Studio 6.2 (free) project without Atmel Software Framework (ASF) support. This results in a project that is easy to understand as well as navigate. You can download the software at [3].
The main loop is located in the file

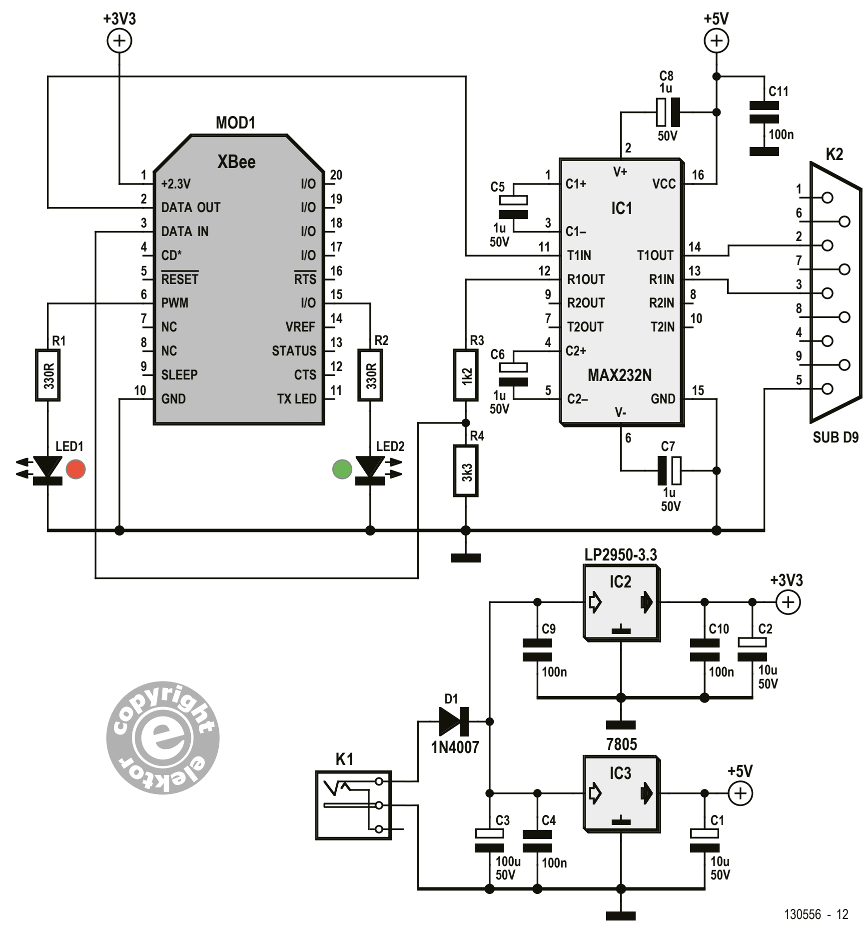


Figure 3. Schematic of the transmitter node (Tx).

RSSI Readout

Compared to the receiver module, the Tx'ing XBee module (Figure 3) has an extra LED on its pin 6 that shows the relative signal strength of the last received packet. It is controlled by a PWM signal with a duty cycle that's a function of the received signal strength indicator (RSSI) value, as follows:

RSSI = (PWM + 5928) / 41

where PWM represents the duty cycle or On time in steps of 200 µs with a maximum of 2400 (12-MHz period). As an example, let's calculate the RSSI when the PWM signal has a duty-cycle of 50%, i.e. PWM = 1200.

RSSI = (1200 + 5928) / 41 ≈ 174

Convert this to dBm as follows: 174 decimal equals 0xAE hexadecimal, interpreting that as a two's-complement signed byte gives a decimal value of –82 dBm.

That was easy and now you understand why we used an LED.

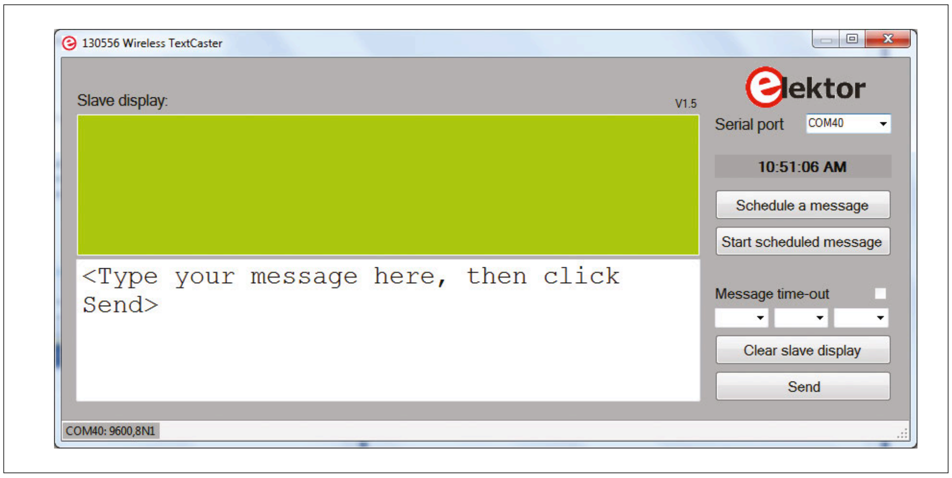


Figure 4. The ZigTexter Network PC application for instant over-air messaging and message scheduling.

"130556-wireless-textcaster.c". It starts by initializing the microcontroller ports and peripherals. The baud rate is set to 9600 bits/s. Once the UART is operational the MCU can initialize the XBee module. This module starts up in data transmission mode. To get it in Command mode send it the string "+++" (without carriage return and linefeed) — this is a standard method to get (wireless) modems in Command mode. Now you can send AT commands to the module. We only have to send one command: "DL", to set the destination address of the module to broadcast. The 64-bit broadcast address is:

```
0x0000 0000 0000 ffff
```

With the DL command you can set the lower 32 bits; the high part, set with the DH command, defaults to 0 so we don't have to change it. The module supports a large number of commands; you can find them all at Digi's website [1]. Configuring the other XBee parameters will be done only once with the utility XCTU (see section **XBee configuration** below). Nothing needs to be done to return to Data Transmission mode; the module will do that automatically when it hasn't received any AT commands for about three seconds. While waiting for this to happen the firmware shows a friendly welcome message.

If you see "(echo)" somewhere on the display, then your module is an echoing receiver. Remember that you should not have more than one of these on your system. Power-cycle the module if you decided to change the position of JP1.

After three seconds the module enters its main loop where it will wait for data to arrive. The welcome message remains visible until it is overwritten by the first message from the server.

The format of a message is very simple: a string of up to 160 ASCII characters with a value of 0x20 (32 decimal, space) or higher. Sending a character with a value lower than 0x20 will clear the display and move the cursor to the upper-left corner. This is the only command available; positioning the characters at the right position on the screen is done by the sender. If the message is longer than 160 characters the display will simply continue printing at the display's home position — it will not scroll. If the receiver has to echo, it will echo only the bytes that it displays; the command byte is not echoed.

The LCD driver is written such that the highest-level functions (`lcd_puts`, `lcd_clear`, `lcd_goto_xy`) present it to the user as a display of four lines of 40 characters. One level down the LCD is treated as two separate displays. This means that a call to `lcd_putc` to print a character at the current cursor position must specify which display (1 or 2) it wants to use.

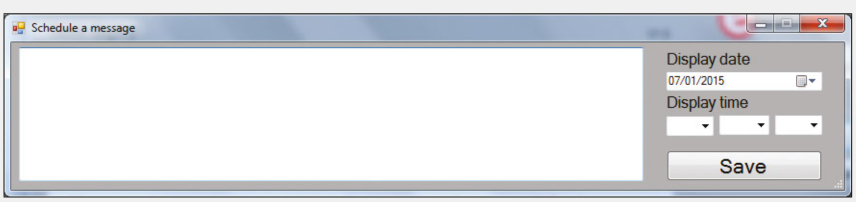
The UART driver does not use interrupts or data buffers, so the main loop must not take too long to complete to avoid missing data. The LCD is wired in 4-bit read-only mode which implies two write cycles to send one byte to it, and the use of delays to wait for the display to digest the byte instead of checking its Busy flag. The delays have been set to 50 µs and two are needed per byte, but this remains largely within the time available to receive one character at 9600 baud (about 1 ms). If your display does not perform too well you may want to increase the 50 µs delay a bit.

Transmitter software

To send and manage messages we developed a simple but effective application in C# (C-sharp) as a Microsoft Visual Studio

Message Scheduling

First click the "Schedule a message" button to open the window seen here where you can enter messages and set the time (one minute resolution) and date when it it's due for broadcasting. Click "Save" for every message. The last sent message will remain on the display until it is overwritten by a new message. Therefore, if you want to clear the display after a certain time you can either schedule an empty message or you can schedule the special message "reset" (lower case). Once you are done scheduling messages close this window and click the button "Start scheduled message" to launch the automatic message transmitter.



The message scheduling input window.

This feature employs a temporary file `Display1.csv` in the folder `c:\Elektor`. If this folder does not exist, it will be created. The file is a text type and you can edit it right off. If you want to keep your scheduled messages you should copy the file to a safe place. To resend the messages, at least change the dates of the messages.

2010 Express project. **Figure 4** shows the program in action. This is free software allowing anyone with a Windows PC to modify the program.

To use it first select the COM port that talks to the transmitter module, then start sending. Type your message in the lower window and click the "Send" button. This will clear the lower window to make room for a new message. If your echoing node is operational the message you just sent should appear in the green upper window. If not, it will remain empty. The message should of course appear on the receiving nodes also.
The windows are multi-line and do word wrapping. By inserting spaces the program will try to keep wrapping when sending the messages to the slave displays. The echoing receiver is aware of this word-wrapping feature and inserts spaces too whenever necessary so that the acknowledge window will word-wrap in the same way as the lower window. Although the goal is to create a WYSIWYG (what-you-see-is-what-you-get) experience, it implies that unless you know exactly what you are doing, you'd better not change the size of the windows and the fonts to avoid messing up this delicate equilibrium. If you feel like it, go ahead and create a more suitable window that corresponds better to the display.

You can set a timeout for a message by setting a time (AM/PM format) and then checking the "Message time-out" box. When the timeout expires the slave displays will receive a clear display command. You can send this command manually by clicking the "Clear slave display" button.
Message scheduling is provided as a feature, see the **inset**.

XBee configuration

The XBee modules require one-time configuration. Also, a special utility, Digi's X-CTU, XCTU or XCTU Next Generation depending on the version, is needed. This is a ridiculously large tool for what we

Component List

Receiver

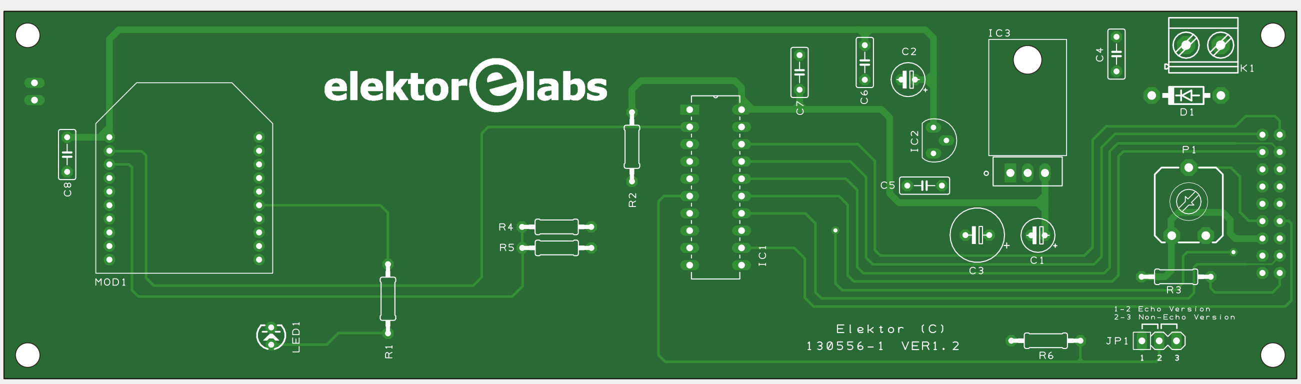
Resistors
(Default: 5%, 250 mW)
R1 = 330Ω
R2,R6 = 10kΩ
R3 = 100Ω
R4 = 1.2kΩ
R5 = 3.3kΩ
P1 = 10kΩ trimpot

Capacitors
C1, C2 = 10µF 50V, radial, 2mm pitch
C3 = 100µF 50V, radial, 3.5mm pitch
C4,C5,C6,C7,C8 = 100nF, 5mm pitch

Semiconductors
LED1 = LED, red, 3mm
D1 = 1N4007
IC1 = ATtiny2313-20PU, programmed, Elektor Store # 130556-41
IC2 = LP2950-ACZ3.3
IC3 = 7805

Miscellaneous
K1 = 2-way PCB terminal block, 0.2" pitch
K2 = 2-pin pinheader, 0.1" pitch
MOD1 = XB24-Z7WIT-004 XBee module
2 pcs 10-way SIL socket for XBee module
LCD1 = LCD, alphanumeric, 40 x 4
2 pcs 9-way SIL socket strip for LCD
PCB 130556-1 v. 1.2

Available in the Elektor Web Store



PCB 90% of true size

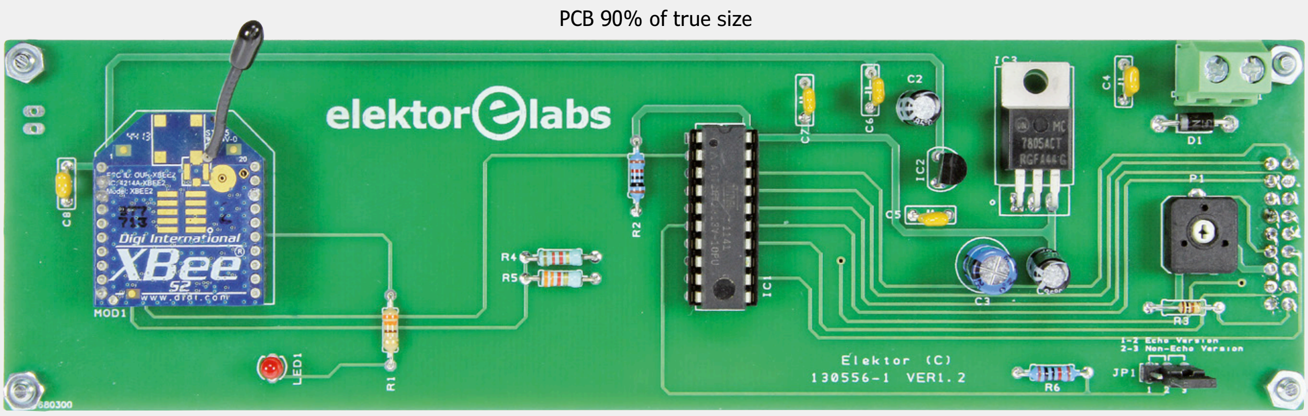


Figure 6. Breakfast is ready! (Mañuel late today)

Web Links

[1] XBee AT commands: http://examples.digi.com/wp-content/uploads/2012/07/XBee_ZB_ZigBee_AT_Commands.pdf

[2] XCTU: www.digi.com/support/productdetail?pid=3352&type=utilities

[3] Project files: www.elektor-magazine.com/130556

[4] Original design on RS DesignShare: www.rs-online.com/designspark/designshare/eng/projects/192

Component List

Transmitter

Resistors

(All 5%, 250 mW)
R1,R2 = 330Ω
R3 = 1.2kΩ
R4 = 3.3kΩ

Capacitors

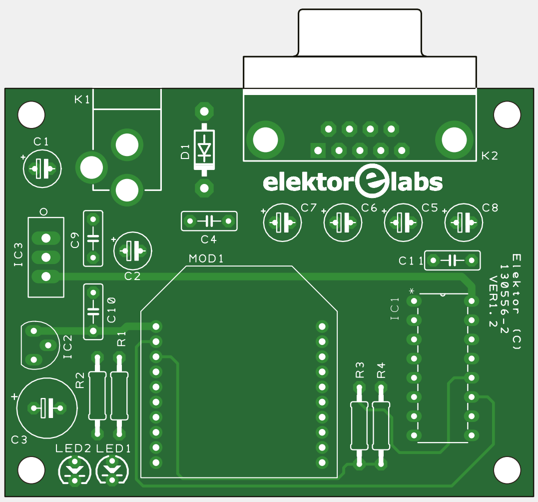
C4,C9,C10,C11 = 100nF, 5mm pitch
C5,C6,C7,C8 = 10µF 50V, radial, 2mm pitch
C1,C2 = 10 µF, 50 V, radial, 2mm pitch
C3 = 100µF 50V, radial, 3.5mm pitch

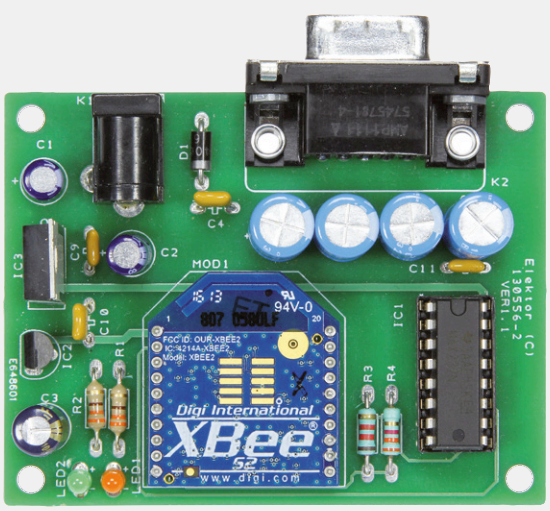
Semiconductors

LED1 = LED, red, 3mm
LED2 = LED, green, 3mm
IC1 = MAX232N
D1 = 1N4007
IC3 = 7805
IC2 = LP2950-ACZ3.3

Miscellaneous

K1 = DC adaptor barrel jack
K2 = 9-way sub-D socket (female), PCB mount
MOD1 = XB24-Z7WIT-004 XBee module
2 pcs 10-way SIL socket strip for XBee module
PCB 130556-2 v. 1.2





need to do with it, but that's the way software gets written these days. We have prepared two configuration (profile) files that have to be loaded in the XBee modules: one for the receiver(s) and one for the transmitter [3]. I suggest that you download XCTU version 5.2.8.6 unless you are using Windows 7-up or OSX. **Figure 5** shows this version.

Launch XCTU and connect the XBee module you wish to configure to a free COM port on your PC. Within XCTU select this COM port on the "PC Settings" tab. Then open the "Modem Configuration" tab and click the "Load" button in the "Profile" group. Navigate to the "130556_Client_Zigbee.pro" or "130556_Server_Zigbee.pro" file that you downloaded from our website [3]. The transmitter must be loaded with the server profile and you can have only one server. Once the profile is loaded XCTU should display "Modem: XBee XB24-ZB" and either "ZIGBEE COORDINATOR AT" (for the transmitter)

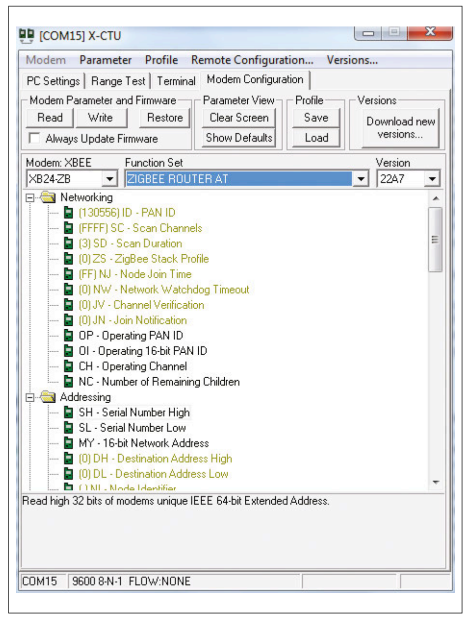


Figure 5. Digi's XCTU utility is needed to load the profile files into the XBee modules.

or "ZIGBEE ROUTER AT" (for the receivers). Now click the "Write" button in the "Modem Parameter and Firmware" group. Repeat these steps for every XBee module you need for your system.

Plug the configured XBee modules on the different receiver and transmitter boards and your system is ready for use. Happy ZigTexting!

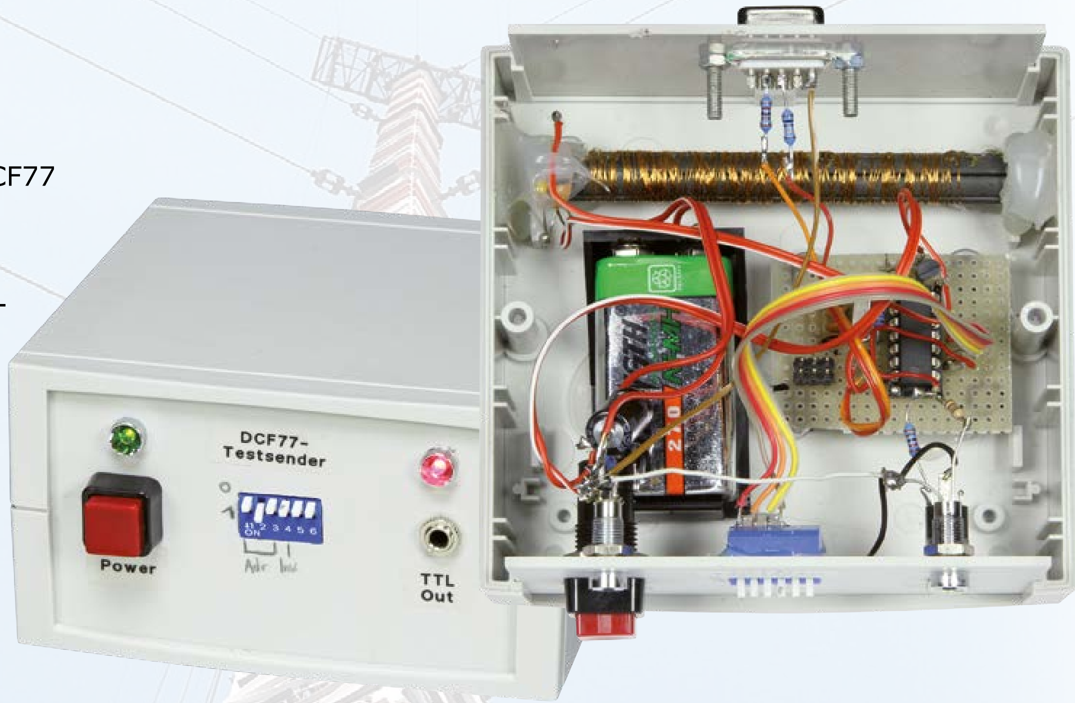
(130556-I)

Air-Your-Own DCF77 Time

Authentic date & time code from a microcontroller

By **Roger Leifert** (Germany)

Testing DCF77 decoders, DCF77 decoding routines or DCF77 clocks is made a lot easier when you have a standards-conformant timecode signal that you can program with your own choice of time and data 'telegrams'.



DCF77 is a VLF time signal transmitter in Mainflingen, Germany, with a range of 1,000 miles. Reception, although not continuous. The author built a circuit using a small microcontroller to produce a DCF77 test signal at TTL level and then (based on [1]) a small test transmitter to produce a radio signal at the correct frequency of 77.5 kHz. In order to check whether a DCF77 device is working correctly, you can preset the DIP switches to simulate particularly critical time events. Another switch lets you detect whether the TTL signal is going out inverted or non-inverted.

The DCF77 time and date are generated continuously over a simplified serial interface. The interface allows you to enter any time and date combination of your choice.

A microcontroller and not a lot more

A glance at the circuit in **Figure 1** reveals that the DCF77 simulator consists of little more than a microcontroller, specifically an ATtiny84 from Atmel with a 20-MHz crystal for the clock generator. The software uses this to produce a standards-compliant DCF77 'telegram' (data packet) once every second. The actual time is programmed using the DIP switches S3 to S5 according to **Table 1**. The output signal is available in TTL/CMOS format on Portpin PA4 (Pin 9) for feeding into a decoder and can be checked with the LED that is driven in parallel.

The output resistor R4 of 1 kΩ enables you to feed the signal directly into circuits operating at voltages below 5 V. The protection diodes integrated into the inputs of all widely used ICs will safely deflect any 'excess current' of up to 1 to 2 mA. Following the suggestion of Burkhard Kainka [4], the serial interface employed doesn't make use of one of the usual interface converters such as the MAX232. To manage without this, we need to invert the signal, since the RS-232 interface operates with negative logic. Most PCs with the original serial interface detect the +5 V/0 V level produced across the current-limiting resistor R3 correctly rather than the standards-specified +12 V/–12 V values. The input levels are taken to TTL level via the protection resistor R2 and the integrated protection diodes in the ATtiny. **Figure 2** shows how the simulator's output appears on the screen of PCs running a Terminal program.

Dividing the system clock by 258 produces, with the help of the built-in timer, a squarewave signal of 77.519 kHz on the OC1A output (pin 7). This is unquestionably close enough to 77.5 kHz (the error is only 0.024% or 240 ppm), meaning that probably all DCF77 clocks will accept the signal as valid. Consequently the method described in [2] for generating 'broken' dividers for the exact DCF77 frequency of 77.5 kHz is not used here. Depressing the amplitude to 15% to mark seconds is achieved using the PWM function built into the ATtiny in the manner set out in [3]. Filtering out the unwanted harmonics of the squarewave signal and coupling to the receiver antenna is handled by the resonant circuit L1/C1. The best tuning of the resonant circuit occurs, in the author's experience, when DCF77 clocks are placed 10 to 50 feet distant from the tester. **Radiating the test signal into free space is an absolute no-no**, so transferring it to the DCF77 antenna of a receiver should be carried out using a coupling loop or by direct connection to the antenna input using screened cable.

The operating voltage can be provided by a 9-V battery or an AC adaptor with 7.5 V regulated output. The voltage regulator IC2 converts this to 5 V for the microcontroller. Current consumption is only a few milliamps, so a battery will provide many hours of test operation.

As time goes by

Once up and running, as soon as you alter the setting of the DIP switches S2 to S5, the software responds to the new date as shown in **Table 1**. DCF77 decoders should continue carry on working without error on their own once they have correctly decoded a DCF77 telegram.

The real 'endurance test' comes at the turn of the year (when both time and date change altogether), at the transition from 28/29 February to 1 March (according to whether it is a Leap Year) and especially at the change from Summer to Winter Time and vice versa. Seven sample times have been programmed in specially for testing these situations, which can be set with the DIP switches S3 to S5. The output begins just over three minutes before the critical moment, so that every DCF77 decoder or decoding algorithm can synchronize at least once. If you then switch off the DCF77 simulator shortly before the critical time, the decoder under test should perform the time leap correctly of its own accord.

Since the end of 2006, bits 1 to 14 of the DCF77 telegram have contained weather information from the MeteoTime organisation together with disaster prevention data. We can generate only random information, as the algorithm for coding this data has not been released publicly [6]. During operation the firmware also polls the serial interface continually. As soon as any date or time data is received in the correct format (DD.MM.YY + CR/LF

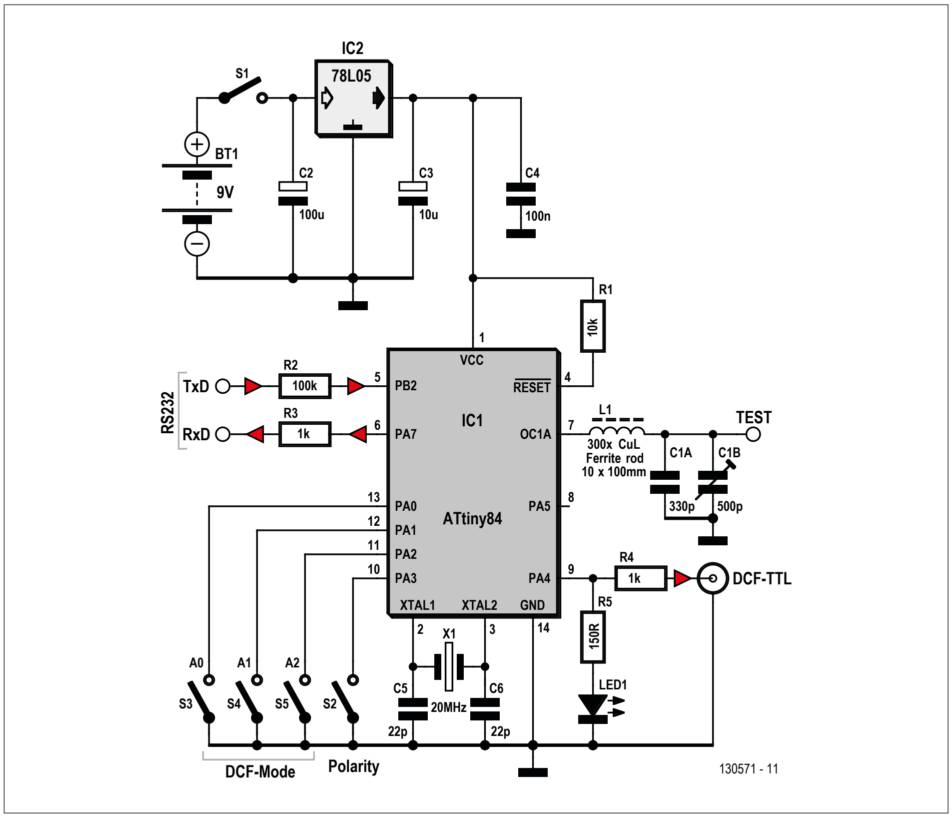


Figure1. The circuit of the DCF simulator consists of little more than an ATtiny microcontroller.

> The DCF77 simulator consists of little more than a microcontroller

and HH:MM:SS + CR/LF), it resets itself and from that moment on it issues DCF telegrams with the new time and date. The firmware is written with the BASCOM compiler [5]. Commented source text, a hex file for burning and the settings of the fuse bytes are freely available and can be downloaded free at [7].
If you wish to alter the program, for example to implement different time and date properties for the DIP switches, the modifications are simple to apply. Since

Table 1. DIP switch settings

DIP switch A2	A1	A0*	Date	Start time	Day of the week	Detail
0	0	0	25.03.12	01:56:50	Sunday	Changeover to Summer Time: the hour 02:xx is omitted. Notification of changeover to Summer Time in DCF-Bit 16=1
0	0	1	28.10.12	02:56:50	Sunday	1. The hour 02:xx is repeated ahead of changeover to Winter Time. Notification of changeover to Winter Time in DCF-Bit 16=1
0	1	0	28.10.12	02:56:50	Sunday	2. The hour 02:xx is repeated following the changeover to Winter Time. Notification of changeover to Winter Time in DCF-Bit 16=0
1	0	0	28.02.14	23:56:50	Friday	No Leap Year: change to 01.03.15 at 00:00
1	1	0	28.02.12	23:56:50	Tuesday	Leap Year: change to 29.02.12 at 00:00
1	1	1	29.02.12	23:56:50	Wednesday	Leap Year: change to 01.03.12 at 00:00
1	0	1	31.12.12	23.56:50	Monday	Turn of the year: change to 01.01.13 at 00:00

*1: DIP switch on; 0: DIP switch off

```
Time: 0:57:01    Date: 28:10.12
Time: 0:57:02    Date: 28:10.12
Time: 0:57:03    Date: 28:10.12
Time: 0:57:04    Date: 28:10.12
Time: 0:57:05    Date: 28:10.12
Time: 0:57:06    Date: 28:10.12
Time: 0:57:07    Date: 28:10.12

Input Date. Format [TT.MM.JJ]
Input Time. Format [HH:MM:SS]

01.02.14 OK. New date accepted.

Time: 02:57:15   Date: 01.02.14
Time: 02:57:16   Date: 01.02.14
Time: 02:57:17   Date: 01.02.14
Time: 02:57:18   Date: 01.02.14
Time: 02:57:19   Date: 01.02.14
Time: 02:57:20   Date: 01.02.14
Time: 02:57:21   Date: 01.02.14

Input Date. Format [TT.MM.JJ]
Input Time. Format [HH:MM:SS]

01:02:55 OK. New Time accepted.

Time: 01:02:55   Date: 01.02.14
Time: 01:02:56   Date: 01.02.14
Time: 01:02:57   Date: 01.02.14
Time: 01:02:58   Date: 01.02.14
Time: 01:02:59   Date: 01.02.14
Time: 01:03:00   Date: 01.02.14
Time: 01:03:01   Date: 01.02.14
Time: 01:03:02   Date: 01.02.14
Time: 01:03:03   Date: 01.02.14
```

the program is somewhat bulkier than the 4 KB allowed with the freeware version of the compiler, the licensed version is necessary. Alternatively you could omit some of the functions and keep the code within the 4 KB limit.

Accurate alignment

The only tricky part during construction is aligning the resonant circuit. In article [1] it is recommended that you wind L1 as a coil with around 300 turns on a ferrite rod of 10 mm diameter and approximately 10 cm long. Getting the precise number of turns and winding them 'tidily' are not important. Instead of making a parallel circuit formed of a single fixed capacitor and a variable capacitor (e.g. 470 pF and 500 pF), you can achieve the same result by adding several fixed capacitors progressively in parallel. Finally, 500 pF tuning capacitors tend not to be found in electronics stores these days (excepting possibly @ Weirdstuff, CA) and not everybody has one in the junkbox.

There are two approaches to alignment, depending on whether you have an oscilloscope or not. If you do, it should pick up the signal at over 4 inches distance from the ferrite rod antenna, using a coupling coil of roughly 10-20 mm diameter made with several turns of copper litz wire. Connect the two wire ends of the coupling coil directly to the input of the oscilloscope, as the signal is very powerful. Then vary the rotary capacitor or the combination of fixed capacitors until you find maximum amplitude on the display. Avoid altering the position and distance of the coupling coil from the ferrite rod antenna during the alignment process.

If you have no oscilloscope, you can generally get by with just a multimeter switched to AC voltage, as long as its input impedance is at least 10 MΩ. You now measure the voltage at the junction of L1 and C1 (Test output) and tune for maximum indicated voltage. Even though few multimeters possess a bandwidth rising to over 77.5 kHz, the high signal level means it is still possible to tune for the relative maximum. If you cannot find a maximum with the given values of L1 and C1, you should either increase the value of one of the two capacitors in 10% steps or reduce the number of turns on the ferrite rod antenna.

(130571)

Literature and Web Links

[1] Martin Ossmann, AVR Software Defined Radio (3), Elektor May 2012, www.elektor-magazine.com/100182

[2] Martin Ossmann, SDR Software Defined Radio (1), Elektor March 2012, www.elektor-magazine.com/100180

[3] Martin Ossmann, SDR Software Defined Radio (2), Elektor 04/2012, www.elektor-magazine.com/100181

[4] Burkhard Kainka, BASCOM AVR Course, Part 1, Elektor September 2008, www.elektor-magazine.com/080330

[5] BASCOM-AVR Compiler: www.mcselec.com/

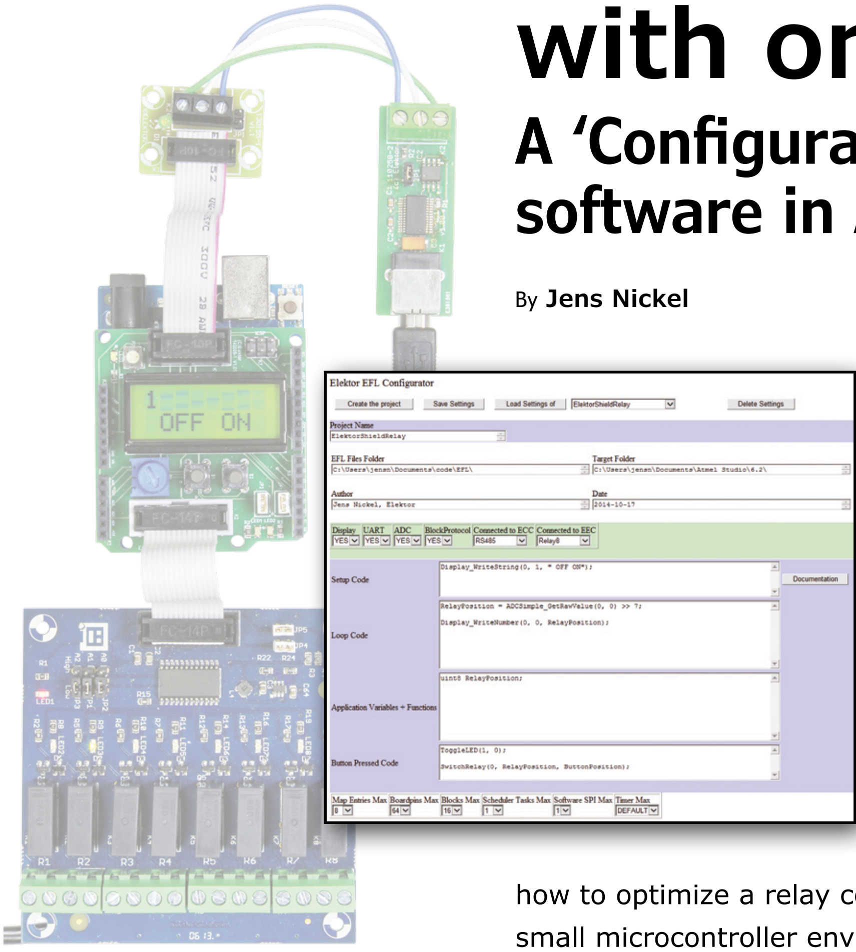
[6] Information on DCF time code: http://en.wikipedia.org/wiki/DCF77#Time_code_details

[7] www.elektor-magazine.com/130571

Make C projects with one Click

A 'Configurator' for modular software in Atmel Studio

By **Jens Nickel**



Modular software libraries such as our EFL have many advantages but using them on a project in a development environment introduces additional steps. This hurdle can sometimes be off-putting for the beginner. Here we will describe a script-based PC program which generates a new EFL project at the click of a mouse. For the more advanced user we include tips to help reduce the firmware's memory requirements. As an example we show how to optimize a relay control and display application to run in a small microcontroller environment.

In the previous edition we looked at some of the advantages that modular software libraries such as EFL provide [1]. Each software module makes use of the properties of the controller and board to which it is assigned but remains independent of all the other hardware in the project. This allows you combine software modules without the need for any changes in the code, in the same way that fully characterized and tested hardware modules can be combined to quickly build a more complex prototype.

The cost of emulating this hardware flexibility in software is the increased number of individual files which need to be integrated into the (EFL) project. There is a (.h/.c) file pair associated with the controller, the controller board and each expansion board. Added to that are the many block files which provide low-level functions for peripherals such as displays (these abstract, for example the wiring between the controller and display). In addition we need the higher level libraries which help simplify programming by the user allowing every peripheral block of the same type on all the boards to communicate using the same function. Lastly we also need the Common-Files which form the basis of EFL.

One step at a time

Unfortunately to set up a modern software development environment like Atmel Studio it is not enough to just drag all the necessary files into a Windows folder. All of the code files must be included in the project with the help of Atmel Studio commands. First of all starting in Atmel Studio with a new project we also need to place the three EFL project folders 'Common', 'Hardware' and 'Libraries'. We also need to make the path of these folders known to the Compiler. In the firmware's main file which contains the application code it is necessary to 'Include' all the Board, Extension and Library Header files. Lastly don't forget the Init and Setup functions for the hardware and library modules.

You can of course generate a new project using a suitable existing EFL project (or use one of our demos) by just copying the entire Atmel Studio project folders and then making changes as necessary. Unfortunately the project's original name is maintained. Manually changing the project name in 'Solution Explorer' gives only a partial solution to the problem, remnants of the original project name still persist.
It was necessary to go, step by step through the process outlined above from the beginning to produce the 'clean' demo project for the EFL article mentioned above.

A project with three files

In the search for a solution I first created a new C project in Atmel Studio and then looked at all the folders and files that were generated. All the files are located in a main folder with the same name as the project (the Atmel-Studio user chooses the name before the project is created, the default name is 'GccApplication1'). In the main folder you will find a file with the project name with the extension .atsln and another folder with the project name (**Figure 1**). This contains two files with the project name, one with a .cproj extension and one with a .c extension. In addition there is an empty 'Debug' folder.

The files with .atsln and .cproj extensions can be opened using a text editor and are formatted using pure XML syntax. This is already a big step on the way to automatically generating a project (without the need to use Atmel Studio itself). The creation of folders and text files with specific names and contents can be performed by all of the normal PC programming languages. A closer look at the .atsln files indicates that it is only the project name that's relevant, the other information is not directly project related. To create a tailor made .atsln file we can start off with one already generated by Atmel Studio as a template. Now we can read and edit the text and replace the project name in (XML-) text with the desired project name and then save the file with the new contents and new name (i.e. the project name with the .atsln extension). Incidentally the .atsln type file is used here because Atmel Studio is based on Microsoft; this IDE is also used in VB.NET und C# amongst other PC programming tools. Visual Studio differentiates between a 'Solution' and a 'Project'; each solution can consist of a number of projects. For our purposes we have just one solution which contains just one project and both have the same name.

A look at the .cproj file contents shows that it contains all the relevant information for the current Atmel-Studio project. For comparison I looked at a cproj file that was created in a 'manually' generated EFL project (**Figure 2** shows a section displayed in the free Notepad++ editor). Everything we need is written down in the form of XML elements:

- The names of the EFL project sub-folders ('Hardware', 'Libraries', 'Common').
- The corresponding path information for the Compiler.
- All the source code files belonging to the project.
- The symbols defined for the preprocessor. So far we have already used the Definition F_CPU=16000000UL (The clock frequency used by the ATmega328 in the Arduino Uno) in the EFL. Using these symbols in the controller code we can for example define the register contents to calculate the baud rate for the UART.

Finally, a look at the .c file. This is the main file of the source code containing the 'main' function and was created by Atmel studio. This file is of course a text file so its contents can be easily edited. The right hand side in **Figure 3** shows how the folder structure of an EFL project should look in Atmel Studio. To create a project like this (or any another modular C project) in Atmel Studio we use the following approach. We deposit three template files in a specific folder for use as the output destination for the ...atsln, ...cproj and ...c files. First off our PC software creates the necessary folder for the Atmel Studio project. Then each of the template files is sequentially scanned, processed and saved to the correct location in the project folder structure under their new name. To simplify the process of automatically editing the template data we use easy to detect repeated character sequences such as 'AAAAA', 'BBBBB' etc. in the template. These are then overwritten by the required content which could for example be the XML elements binding the source code files in the project.

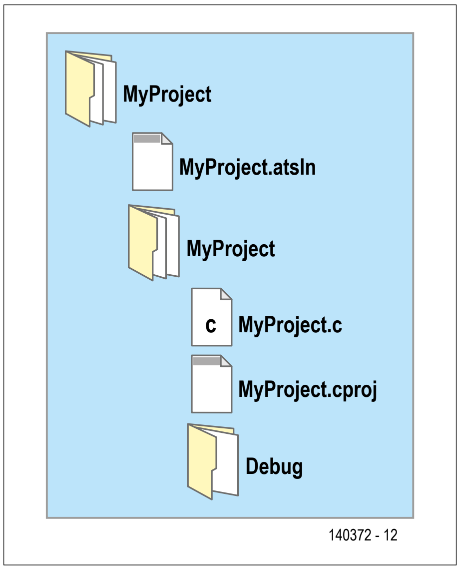


Figure 1. The file structure generated by Atmel Studio when a new C project is created.

Also we attach the sub-folder 'Common', 'Hardware' and 'Libraries' (next to the 'Debug' sub-folder) for the EFL project and copy all the necessary EFL files here. The PC-Software knows where to find the relevant EFL files ('Controllers', 'Boards', 'Blocks' etc.).

The Script System

Now we need to create some PC-Software that will be suitable for the job. In principle any program language can be used which has commands to create folders, to read and save text files and which

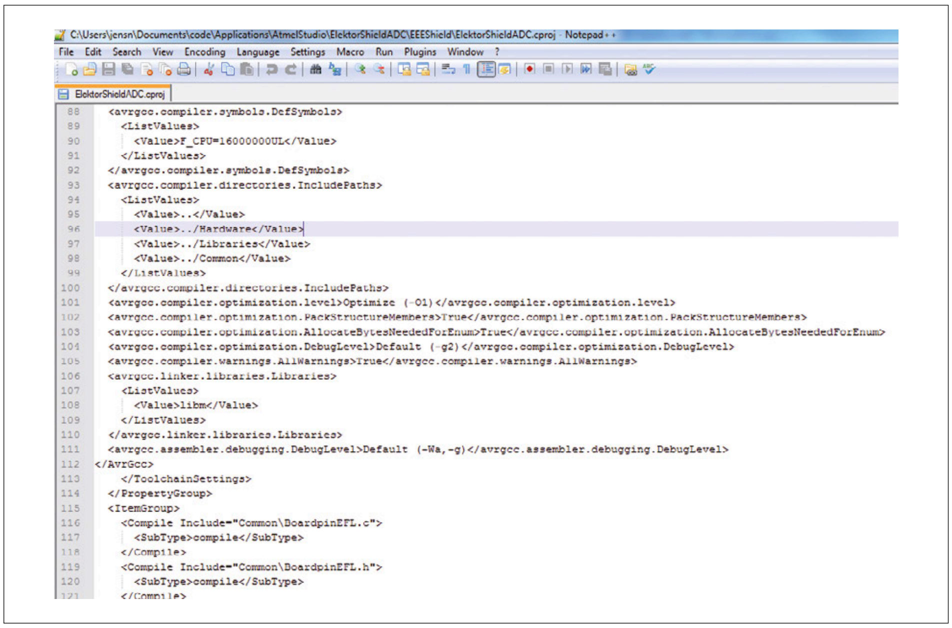


Figure 2. The .cproj file contains all the information for an Atmel Studio project in XML form.

can manipulate text and strings. I have chosen here a script-based framework which I developed for use in publishing. Amongst other things we use it for generating lists and files from data for our editorial planning system.

For a local application like the following software example the scripts are stored as text files. They are processed by an Exe application (i.e. the Interpreter or 'Script processor'). The first script to be read generates an HTML based user interface which is represented as a web control element by the Exe application. When a button is pressed in the HTML interface, the event from some Java script is passed 'outwards' to the Exe application which can then call another script. This sounds unnecessarily complicated but has a significant advantage; when you find you need to introduce other elements in the user interface and/or other functions then it's only necessary to change text in the script file, the Exe file remains unchanged. In principle the script is also platform-independent. The script processor (implemented in C# under for .NET) is only suitable for the Windows environment.

The Exe file, the dll files and the scripts can all be found in a common folder which you can download from the project site relating to this article [Proj]. The scripts, together with the configuration and additional files belonging to the script application are located in the sub-folder 'APP'. We can see our three template files in this 'APP' folder (in the APP\FILES\ sub-folder).
The more advanced PC programmer will of course take a look at the script files (in sub-folder APP\SCRIPTS\) and the data files (APP\DATA\). User defined settings will, for example, be stored in the data files. A brief overview of the script language is given at the end of this article.

Generate a project…

A click on the Exe file will start up the 'EFL Configurator' App. First of all, the HTML-based user interface is built up and displayed (**Figure 4**).
The saying goes that all journeys start with a single step: Due to time constraints, in its present state the Configurator is limited to a very specific hardware setup. There are of course many extensions and enhancements imaginable where you would be able to first choose the processor and controller board and then at another level select an appropriate expansion board. In its present stage of development however the EFL Configurator is only suitable for the Elektro Extension Shield [ExtShield] featured in the summer edition of Elektor, connected to an Arduino Uno. Any of the following modules can also be connected to the shield:

- ECC-RS485-Module and/or
- EEC-Module with eight relays or
- EEC-Module with external 16 bit ADC

All three modules and the corresponding demo project were described in the previous edition of the magazine [CModule].

You can enter the project name at the top of the user interface screen. In the fields below you can enter the path pointing to where the latest version of the EFL code inventory has been saved on your computer (this can also be found in the download for this article [Proj]). This points to the EFL folder so the path ends with \EFL\. You also need to specify the path where the projects generated in Atmel-Studio will be stored. Next is the text box for the project author's name and the date, the format of which you can choose yourself. You will find this information later on as comments in the source code.

Below you can find selection boxes where you can enter the names of libraries used in this project. The LEDButtonEFL- library is used in every project. The DisplayEFL-, UARTInterfaceEFL- and ADCSimpleEFL- library are also required when you want to use a display or the UART for communication or to measure an analog input value. The default here is three times 'YES' so that all the functions of the Elektor Extension Shields and the Arduino (display, pot and UART) can be used with the least hassle. The BlockProtocol [4] allows you to remotely control the peripheral devices on the board and the expansion modules

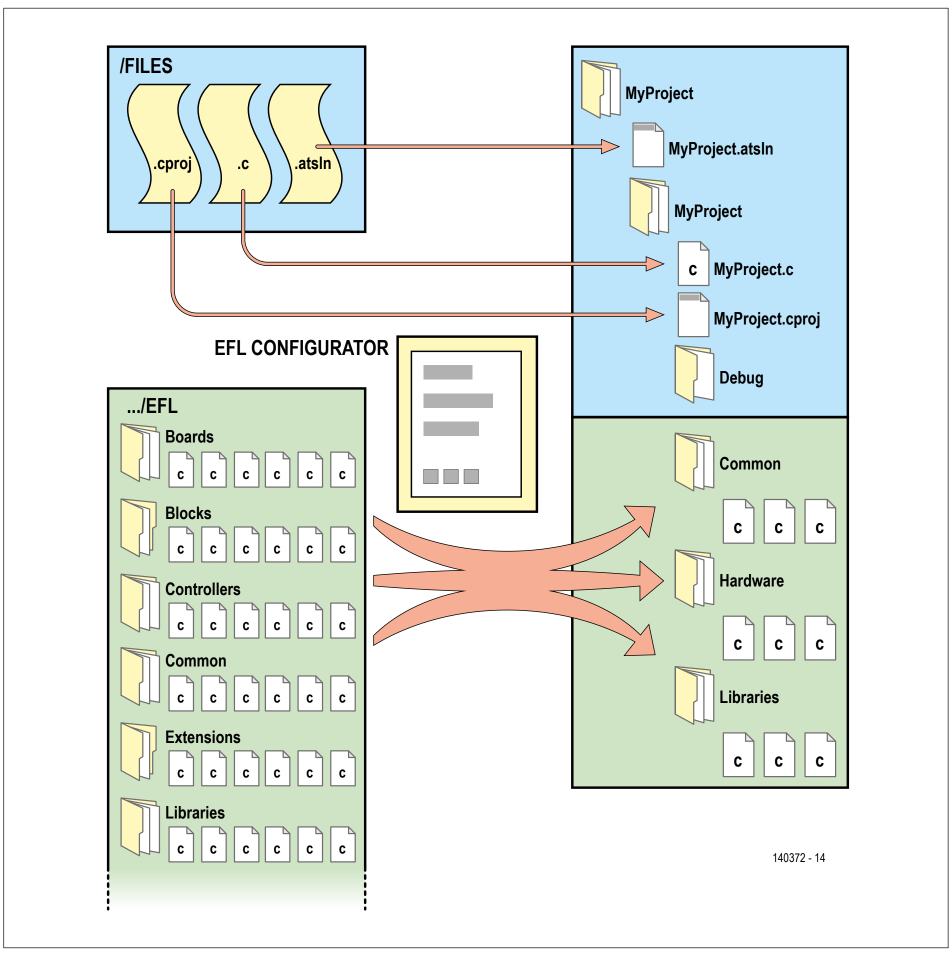


Figure 3. With just three template files and a folder for all the generated EFL files for our configuration we build a complete EFL project for Atmel Studio.

via a terminal emulator program. You can specify which expansion modules are used in the neighboring dropdown box.

…And manage it

In the Text boxes below you can enter the application-specific source code for the EFL projects. In the uppermost text box you need to enter the functions to be called when the application starts. This is followed by code that is executed repeatedly. Application specific (global) variables and functions come in the next text box. In the last text box is the code that runs when one of the buttons are pressed (from the first button block). Which button is pressed is given by the ButtonPosition variable. Pressing the 'Documentation' button opens up the Doxygen documents in Internet Explorer. Here you can view the syntax of the library functions.

It is also possible to leave the text boxes empty and generate an EFL project that just contains the essential initialization and setup commands for the libraries. The actual application code can then be entered directly using the Atmel Studio Editor.

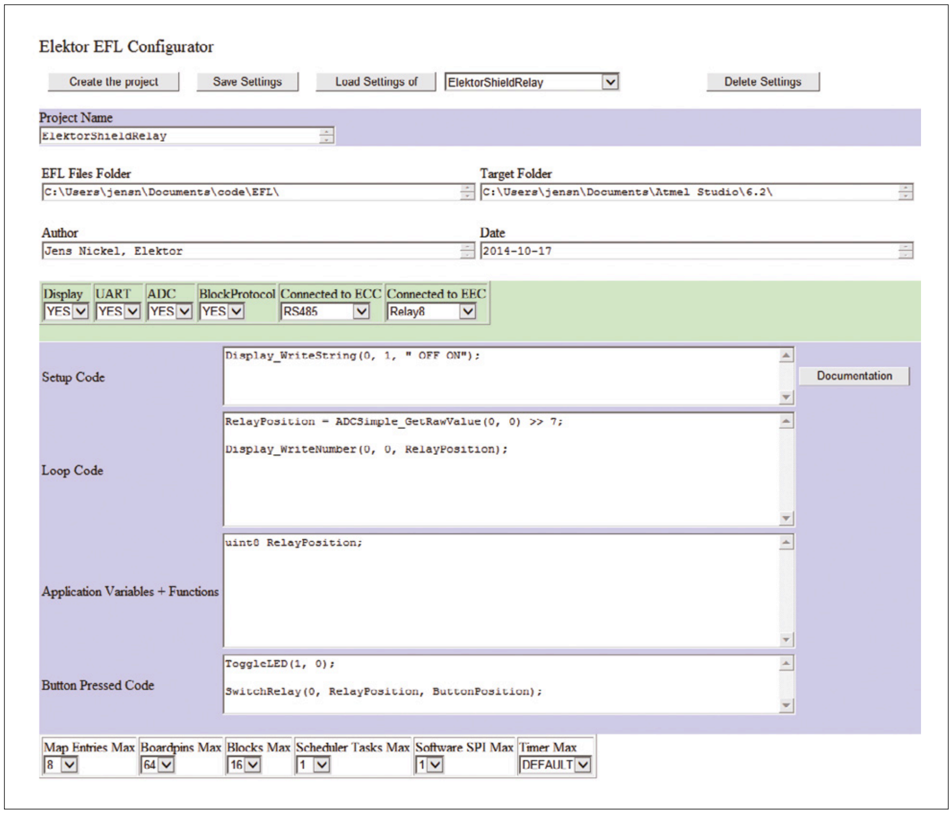


Figure 4. OK so the EFL Configurator won't win any beauty contests but it is the result of a lot of work. With just a few clicks you can generate different firmware versions to investigate its influence on the code and RAM needs.

Scripts

The scripts used are written in the scripting language 'sheets' developed by the author. They are stored as text files in a local App such as the EFL Configurator. A script can have multiple input values and one output value. The scripts were originally used to dynamically generate XML or (X) HTML documents so the output is called *XMLResult*. The script code itself is written in the form of XML and all of the script commands are XML elements. The commands allow you to mix XML- or (X)HTML elements in script code which by script execution are simply written to XMLResult. Scripts that generate HTML pages (e.g. user interfaces) or XML documents, make it easy to keep a record (see for example the script 'Base.txt' of the EFL Configurator). The command TEXT simply writes the text placed between the tags in XMLResult:

```
<DIV>
<TEXT>Hello World!</TEXT>
</DIV>
```

```
XMLResult then contains:
<DIV>
Hello World!
</DIV>
```

The command

```
<TEXT Press='DoSomething' >Do something!</TEXT>
```

Instead of just text this will create an HTML button labeled 'Do something!'. Later on when this button is pressed in an HTML form, the *DoSomething* script is read from the text file *DoSomething.txt* and processed. In this script you have access to the contents of HTML control elements of the user interface that was called from the script.
This can for example be text boxes. Text boxes can also be easily generated by a script:

```
<ENTER ID='EnterText' >Replace the text here</ENTER>
```

In the *DoSomething* script we can gain access to the contents of this text box using the expression `@EnterText`. This is then stored in the *EnterText* variable when the script is called and given as an input value.
We can also use simple variables like this in the script itself and give them a value. The commands:

```
<SET RName='TargetFolder' >C:\MyFolder\</SET>
<TEXT>@TargetFolder</TEXT>
```

Will write the text 'C:\MyFolder\' in the XMLResult.
The character '@' gives us access to the contents of a variable. Instead of '@' plus a variable name we can also use the result of a calculation in our script, the arithmetic expression must be bracketed with '@{' and '}'.

The commands

```
<SET RName='TargetFolder' >@{Add(C:\
MyFolder\,MySubfolder)}</SET>
<TEXT>@TargetFolder</TEXT>
```

Outputs the text 'C:\MyFolder\MySubfolder'. We can achieve the same result more directly with

```
<TEXT>@{Add(C:\MyFolder\,MySubfolder)}</TEXT>
```

Any special symbols such as 'comma', 'open brackets', 'close brackets' etc. must be replaced by predefined constants which always begin with '@@': @@comma, @@Open, @@Close. Status variables also in the Script interpreter begin with '@@': for example '@@DateToday' contains the current date formatted as YYYY-MM-DD.
The most important (complex) data type of the scripting language are the spreadsheet Sheets which are composed of from 1 to 64 columns and which can contain data of different types. Each row of the Sheets contains a data set with the data from all the columns. A sheet can have up to 1023 rows. Each column has a name to allow referencing a specific table entry within the data sets.
All the sheets used in a script (up to 32) must have their names declared at the beginning; multiple sheet entries are separated with semicolons:

```
<SHEETS>Files;Symbols;</SHEETS>
```

With the SET command you can not only fill with simple variables as shown above but also fill the whole sheet.
The command

```
<SET Array='True' RName='Folder;File;' RS='Files' >
Common;BoardpinEFL.c;
Common;CommonEFL.h;
Libraries;LEDButtonEFL.h;
</SET>
```

Will put two columns with the names 'Folder' and 'File' in the 'Files' sheet and fill the table with the values between the tags. Here we can also reference variables (@...), Status variables (@@...) and computational expressions (@{...}).

With the expression @*Sheetname*.*Columnname* you have access to the contents of a specific column of the current data set (row) of a specified sheet. The expression @*Columnname* will be sufficient if the column name is only used in one sheet. After filling the Sheets the first data record is always current. The command FOREACHROW allows you to go through the sheet row by row, for example look at the 'Files' sheet defined above. Inside the loop you have access to the data in individual columns.
The commands:

```
<FOREACHROW Of='Files' >
<TEXT>@Files.File</TEXT>
<TEXT>;</TEXT>
</FOREACHROW>
```

Produce the following text in XMLResult:

```
BoardpinEFL.c;CommonEFL.h;LEDButtonEFL.h
```

READ and WRITE are instructions for reading and storing data in a database. In a local application such as the EFL Configurator the database consists of text files, just like the scripts you find in the APP folder. The sub folder is named after the database (in our example just 'DATA'). Each text file contains a table, organized in the same way as a Sheet with rows and columns.
Using FILEREAD and FILEWRITE allows text (and binary) files to be read and generated. FILERC (RC = Rename/Copy) allows you to move and copy files and also to rename them. With MAKEFOLDER you can create a folder and all the parent folders will also be automatically created. MESSAGE writes a message to the screen. There are also many other commands that we haven't yet covered and we are currently working on more comprehensive user documentation.

The advantage is that once the application has been programmed into the Configurator and 'Save Settings' has been selected it saves all the settings including the codes in a text-based database (see APP\DATA\tblProjects.txt). The data set is then stored under the name assigned to the project. Using the drop down box at the top you can select any of the projects that you previously saved and then with 'Load Settings of' you can load the project together with all the settings and code into the EFL Configurator.
This provides a degree of EFL project management – and (importantly for ongoing project maintenance) is independent of the chosen development environment. Using the settings and application code written in this case for the 'Atmel Studio' target could just as easily be used in the future to generate an Eclipse project. This means that for system maintenance, programmers can look beyond not only the limitations of contemporary controllers and boards but also the development environments used to build the applications.

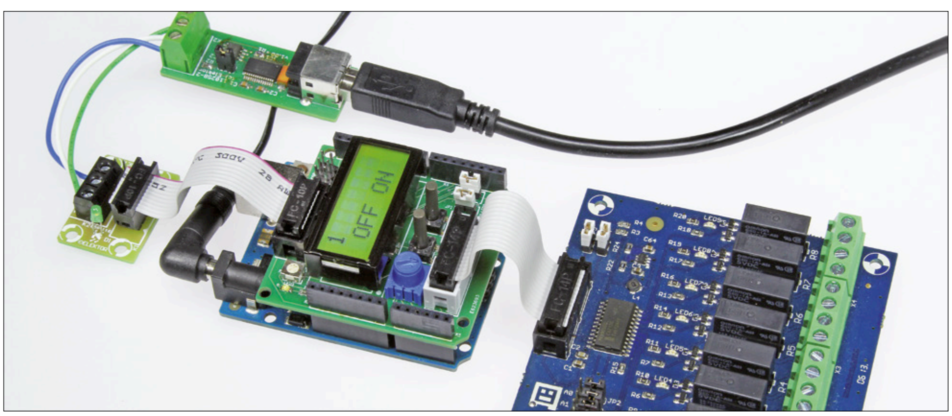


Figure 5. The first example generates code for the relay controller, described in the last edition [1].

The First Test

Every EFL article benefits from an example: For the basis of our project management we will use the three applications examples that were prepared in the last issue [1].

The first example uses the 'ElektorShieldRelay' application (Figure 4). **Figure 5** shows the project hardware that we described in the last issue. Eight relays can be operated locally via the pushbuttons, the pots and display on the extension shields. In addition they can be controlled from a PC using a terminal emulator program; for simplicity we used the Block protocol.
When we now click on the 'Create Project' button the complete EFL project will be generated at the chosen location.

Clicking on the newly generated .atsln file will open the project in Atmel Studio (**Figure 6**). In Solution Explorer you can see all the necessary files are shown in the three EFL folders. The main file of the source code contains all the #include-, Init- and all the setup instructions for libraries plus the application specific code.

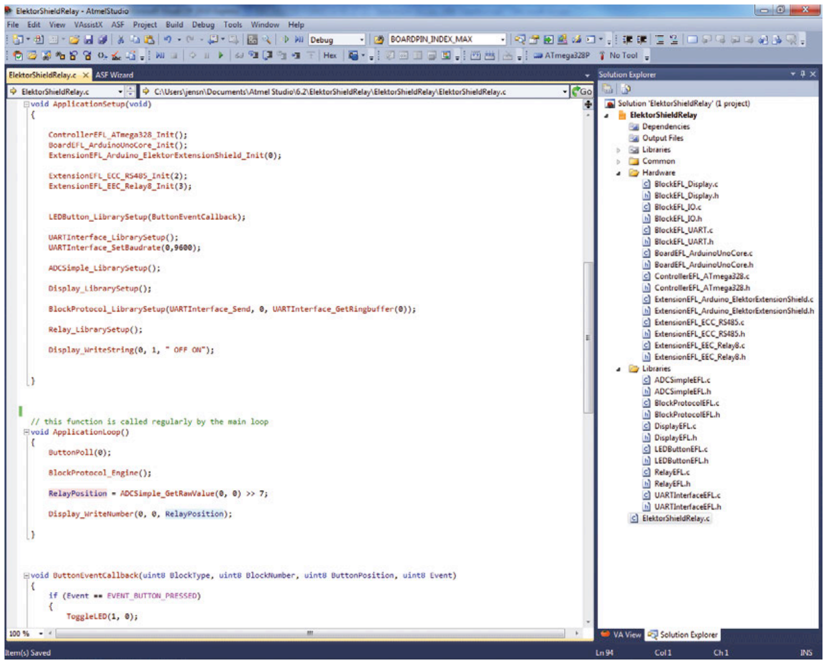


Figure 6. Yeeha, it works: the automatically generated project in Atmel Studio.

The project should now compile without difficulty and can then be loaded into the controller.
During project development it is recommended to have both the EFL Configurator and Atmel Studio opened in parallel. When changes to the code are made in any of the EFL configurator's text boxes you can just press 'create project' without changing the project name. The main file of the code in the existing Atmel studio project is then replaced by a new File and is changed (after confirmation in a dialog box) in Atmel Studio. Now we can compile and test the new code.

Minimize memory usage

With the help of the project manager we are now able to generate different versions of an application. We need this too in our EFL project to help optimize the program's memory requirements. In Configurator along the bottom there are some boxes that you can make selections which influence the project's RAM requirements. It starts with the size of the EFL tables [5], namely the Controller-Features map such as ADC, UART etc. The board pin table to map the wiring and the Block-Table which describes all the peripheral blocks and the expansion connectors.

Once the firmware described above has been burned to the controller then we can not only control the relays but also look at the contents of the EFL tables currently in use using a terminal emulator program (command *x* <CR>) [4]. We can see that we need four entries in the map table, around 60 board pin entries and 12 entries in the Block table (**Figure 7**). There is memory space for eight Map entries (4 bytes each), 64 board pin entries (2 bytes) and 16 entries (5 bytes each) provided in the block table. We can optimize the memory allocation by reducing the number of Map table entries to four and the size of the block table to 12. Apart from these requirements the Software-SPI interface controller code implements a 64 byte ring buffer. The code for the three timers is also relatively memory-hungry. The same is true for the scheduler that can call timer-con-

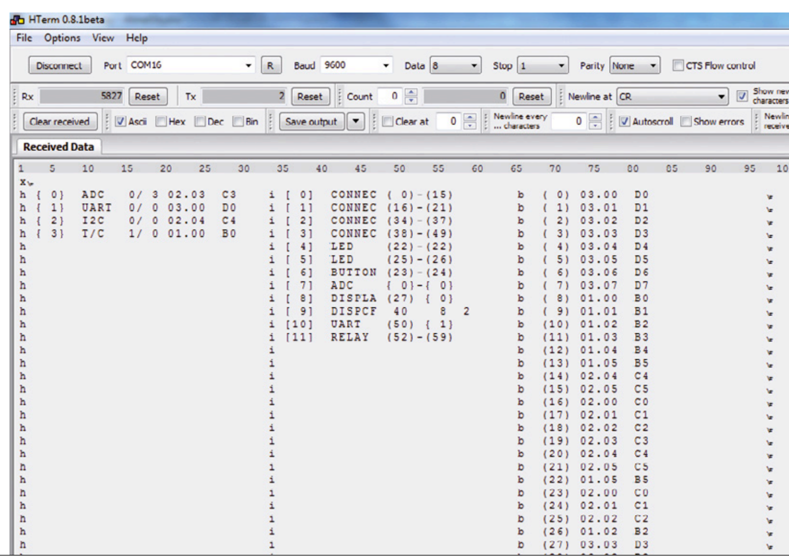


Figure 7. In Terminal program we see how big the EFL tables must be for our application. Now we can optimize the RAM requirements so that the firmware will run on smaller controllers.

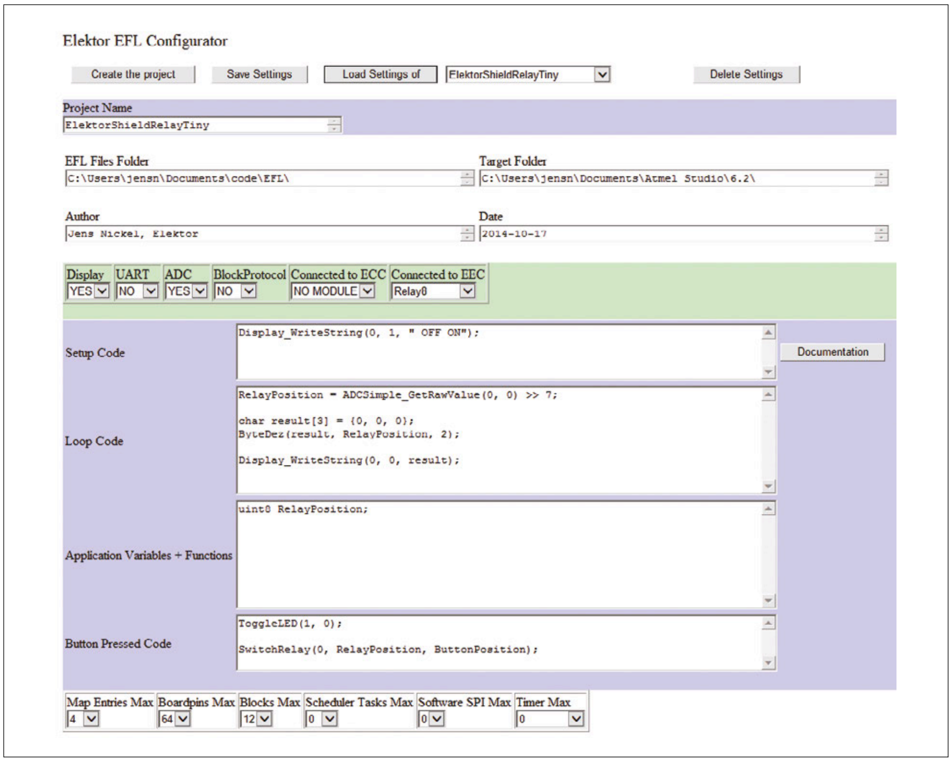


Figure 8. We set the memory requirements of both the software SPI interface and the timer to zero (below right). Neither of these features are necessary.

trolled functions (which have not yet been described). We therefore optimize the code as shown below in **Figure 8** because we do not need the timer or the SPI.

When we now produce a modified project the corresponding symbols are stored to the .cproj file (the symbols can also be read using Atmel Studio, when you right click on the project name and select Properties, Toolchain and Symbols). When the project is compiled again after optimizing

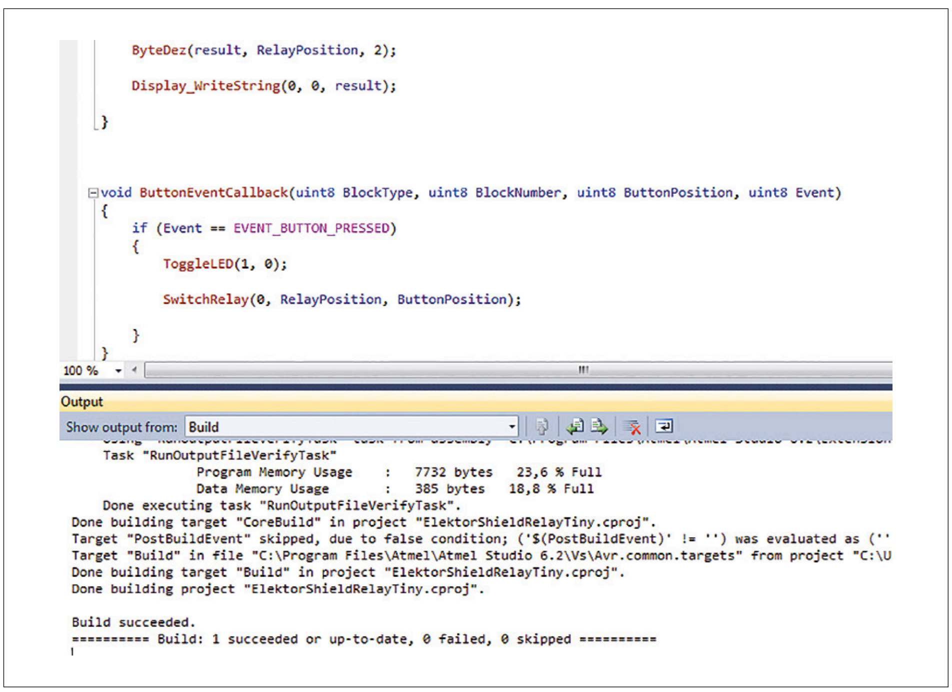


Figure 9. With the remote control function in the firmware removed, the relay control program fits into an ATmega88.

the memory needs we can see the difference. The results (in the Out window) now show that compared to the original we need just 923 bytes of RAM instead of 1091 bytes. The program still retains all the functionality of the original version. The Flash memory requirement is 16,260 Bytes which comfortably fits the available memory space of an ATmega168.

We can make more cuts if we dispense with the remote control function for the relays. We can then just unplug the RS485 module and select 'NO' in the dropdown box by BlockProtocol and then UART.

After compiling the new project we arrived at a finished code size of 9,072 bytes with a RAM requirement of 389 bytes; This implements a nice relay control using the pot and switch with a status display on the screen. Unfortunately this puts the program's Flash needs beyond the magic 8 K available with an ATmega88 (or ATtiny85).

It didn't take long to track down a memory hog in the application code. The `Display_WriteNumber(...)` function accesses the function `sprintf(…)` in the DisplayEFL-Library. We don't really need such a powerful function to display the characters 0 to 7. We substituted the lines shown in the text box Display_WriteNumber(...) with the lines shown in **Figure 8** in the text box 'Loop Code' and generated the project again with the EFL Configurator. You can see the results in **Figure 9**. The project 'ElektorShieldRelayTiny' can be found in the supplied Project settings.

Work on EFL Configurator is ongoing, support for additional hardware and other EFL applications is already underway.
Perhaps we have given you an appetite to develop your own Configurator for Atmel Studio projects or even for use with other development environments. We are always happy to hear of your own experiences, you can reach us via elektor-labs.com or just mail the editor!

(140372)

Web Links

[1] www.elektor-magazine.com/140328

[2] www.elektor-magazine.com/140372

[3] www.elektor-magazine.com/140009

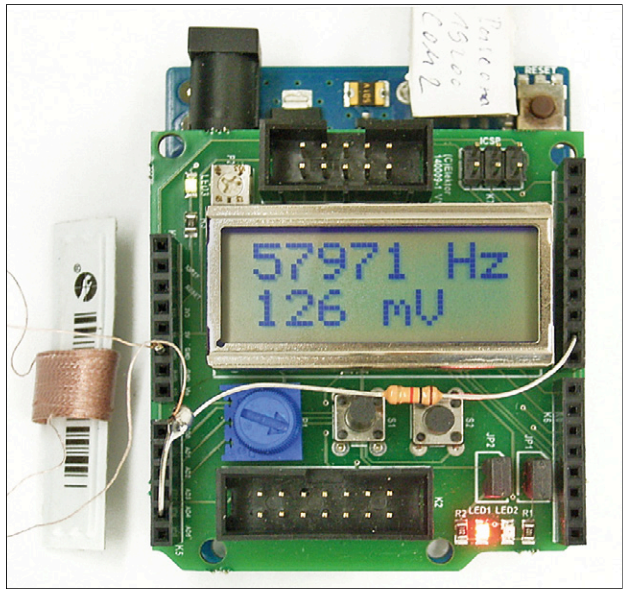
[4] www.elektor-magazine.com/130154

[5] www.elektor-magazine.com/120668

Security Labels are the Key

Door-entry system using Bascom

By Burkhard Kainka



As we saw in our recent microcontroller course, Bascom running on an Arduino Uno together with the Elektor Extension Shield builds a powerful and versatile environment that can be used for a wide variety of applications. Here we build a low-cost contactless door-entry system and use some clever software to evaluate the resonant behavior of a standard security label.

It's increasingly common for shop owners to protect high value goods with security labels. When you buy an item that carries a label the cashier will rub it over a special area on the counter to deactivate it before packing it in a bag. Should the cashier forget to deactivate it, an alarm sounds when you pass through the exit. Maybe these labels can be reused for other purposes… To find out how they work The Net is, as usual a good place to start; try searching for 'acousto-magnetic strips'. This type of label is made up of two or more metal strips (**Figure 1**). A soft iron magnetic strip (sometimes two) of a defined length with a mechanical resonance of 58 kHz. One strip is made up of a special material called amorphous metal (also known as metallic glass or glassy metal). In order for the resonant strip to couple to an external alternating magnetic field the resonator needs to have a magnetic bias. This is taken care of by a second strip made of semi-hard magnetic material (**Figure 2**) which becomes magnetized. With these strip in close proximity the resonator vibrates in response to a 58 kHz alternating magnetic field and emits an alternating field that can be detected. This is the signal that triggers the alarm at the exit. Once the label has been degaussed at the checkout counter its magnetic field is neutralized and the label no longer responds to the 58 kHz field.

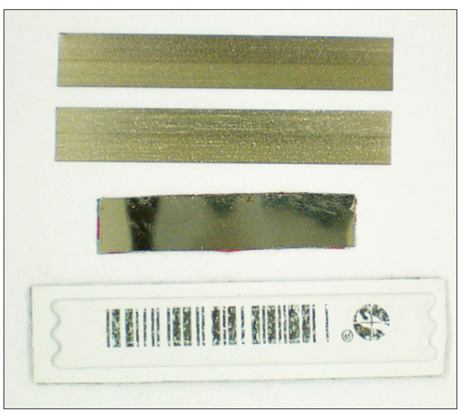


Figure 1. The AM label components, shown here are two resonators and a (shorter) magnetic strip.

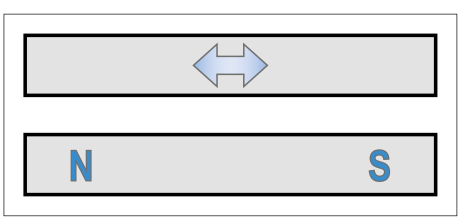


Figure 2. The AM label layout.

Reactivate

At home your purchase still carries the deactivated label. You can make it active again by stroking it with a strong magnet. This re-magnetizes it and converts it back into a resonator. A quick test rig can be made by connecting the output of a signal generator to a coil of about 100 turns and using a scope to explore the label's properties (**Figure 3**). It will typically show a resonance at 58 kHz. A deactivated label shows practically no peaking response to the signal. The mechanical resonance also produces a sound that can be detected with an ultrasonic detector.

Armed with this knowledge you can begin to think of other possible applications of the technology. How about, for example, an electronic door-entry system that uses security labels as a key to gain access? The simplest way to build such a system would be to use a microcontroller like the Arduino Uno. Thinking over the hardware requirements you might assume that a DDS generator would be needed to generate the signal and then a complex detector/rectifier so that the controller can measure the signal. In practice we can simplify things. A controllable oscillator with sufficient accuracy can be made using the timer output OCC1A of the ATmega328 controller. A rectifier to measure the signal will also not be needed providing an AC signal no larger than about 0.3 V_{peak} is applied to the analog input, sampled often enough and the measured values averaged. All negative signals will have a value of zero so this gives us a signal detector with almost ideal properties. The complete circuit (**Figure 4**) is easily implemented with an Arduino Uno and an Elektor Extension-Shield [1]. This shield has the correct type of LCD to allow the display of measured values. The 'unlocked' signal of the electronic lock is output on pin

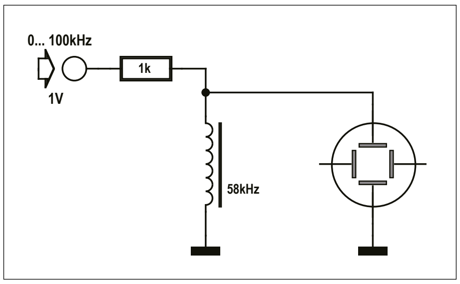


Figure 3. Measuring the resonant frequency using a signal generator and scope.

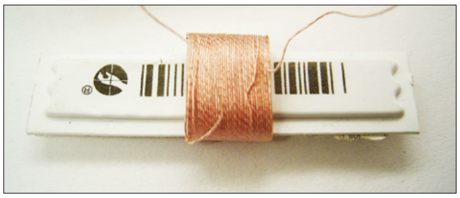


Figure 5. The medium-wave coil.

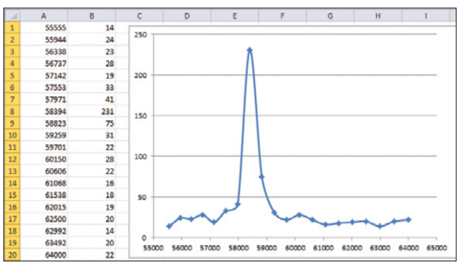


Figure 6. The resonant response.

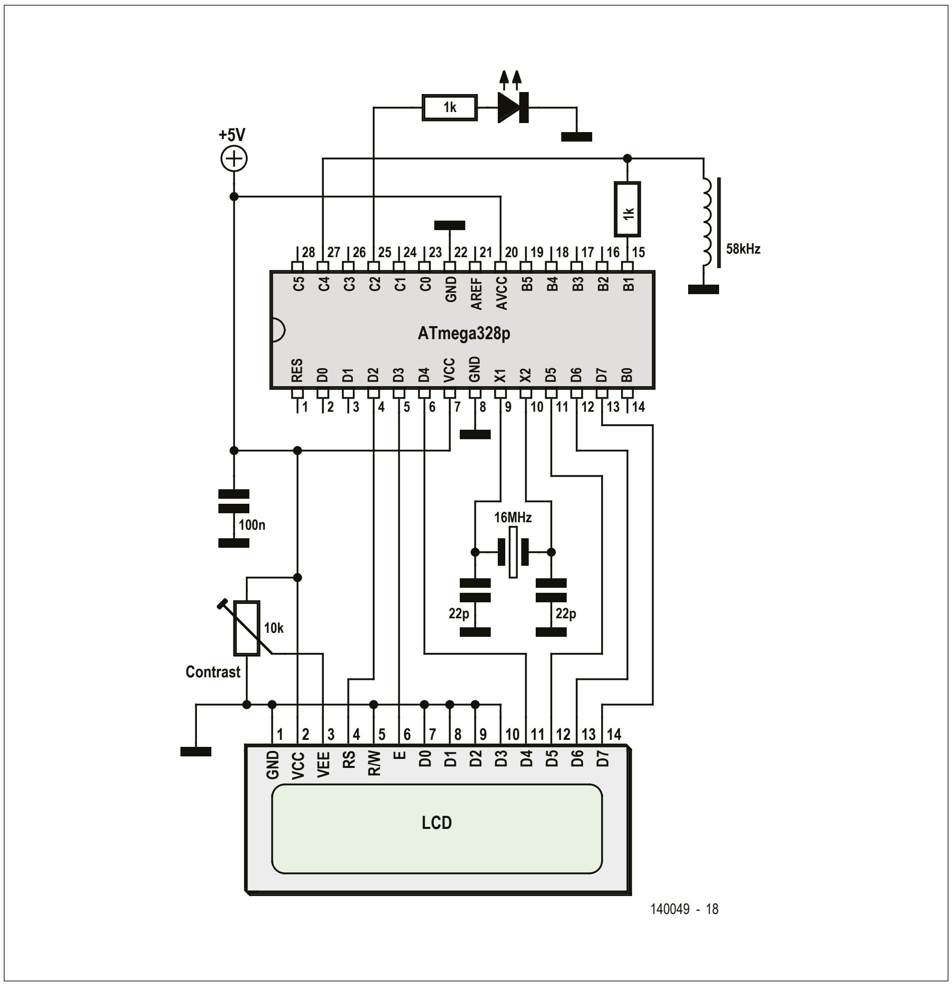


Figure 4. Values displayed on the ATmega328.

PC2, indicated by the state of LED1 on the shield. In addition to this you just need one resistor and a suitable coil. The coil we use here (**Figure 5**) is a medium waveband antenna coil from an old AM radio fitted with a 10 mm diameter ferrite rod. A little bit of pressure squashed the coil slightly so that the label slides neatly inside. The same coil has also made an appearance in another Elektor article; it was used in the preselector of the Elektor SDR [2]. You can wind your own coil here using 80 to 100 turns of 0.2 mm (approx 32 AWG) enamel coated wire or salvage a winding from a junked radio. It's important to ensure that the coil has a low value of capacitance and has approximately the right value of inductance. Make sure that it doesn't exhibit self-resonance in the frequency range of interest.

Software

The software (**Listing 1**) is written in Bascom and uses Timer 1. The program generates a rising series of 20 frequencies centered on 58 kHz. At each frequency 64 measurements are made using ADC4 and the values averaged. The resulting values are then stored in an array. During measurement the values of frequency and signal amplitude are sent to a PC using the serial interface. Here a graph of the resonance characteristics can be plotted (**Figure 6**). It is also possible to display the values on an LCD by removing the comment marks inserted in front of the LCD commands in the code listing.

After the measurements the program searches for the maximum amplitude value stored in the array and its corresponding frequency. The resonant frequency is then transmitted again so that the waveform can be easily displayed on an oscilloscope. The LCD shows the measurement values. The software now decides if the key is recognized or not. When the resonant frequency lies between 57 kHz and 59 kHz and the signal level is above a threshold the lock will open and the red LED comes on. The measurement process will be repeated with the same result so long as the label remains in the coil.

This contactless entry system is convenient to use but will only be secure if its operating principle remains secret. If the level of security needs to be much higher, the worry is that there are so many potential keys in circulation that could be used to open the lock. You can however remedy this and make it more secure; trim the strip so that

it's a bit shorter and this will change (increase) its resonant frequency. Using this trick you can make several 'keys' with different length strips and then build a multi-function lock that responds in different ways, depending which key is presented.

(140477)

Web Links

[1] www.elektor-magazine.com/140009

[2] www.elektor-magazine.com/090615

Listing 1. Resonant frequency measurement.

```
'-----------------------------------------------
'UNO_AMetikett.BAS  58 kHz
'-----------------------------------------------
$regfile = "m328pdef.dat"               'ATmega328p
$crystal = 16000000                     '16 MHz
$baud = 9600
$hwstack = 32
$swstack = 32
$framesize = 64

Dim D As Long
Dim F As Long
Dim N As Byte
Dim U As Word
Dim Um As Word
Dim Ux(50) As Word
Dim I As Word

Led1 Alias Portc.2
Led2 Alias Portb.2
S1 Alias Pinc.0
S2 Alias Pinc.1
Portc.0 = 1                             'Pullup
Portc.1 = 1
Config Led1 = Output
Config Led2 = Output

Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5
   , Db6 = Portd.6 , Db7 = Portd.7 , E = Portd.3 ,
   Rs = Portd.2
Config Lcd = 16 * 2
Cls
Cursor Off

Config Timer1 = Pwm , Prescale = 1 , Pwm = 10 ,
   Compare A Pwm = Clear Up

Tccr1a = &B10000010                     'Phase-correct
   PWM, Top=ICR1
Tccr1b = &B00010001                     'Prescaler=1

Config Adc = Single , Prescaler = Auto , Reference =
   Internal_1.1
Start Adc

Do
  For I = 1 To 20
    D = 145 - I
    F = 16000000 / D
    F = F / 2
    Print F ;
    Print "  ";
    Print Chr(9);
   ' Locate 1 , 1
   ' Lcd F
   ' Lcd " Hz       "
    Icr1 = D
    Ocr1a = D / 2
    Waitms 10
    Um = 0
    For N = 1 To 64
      U = Getadc(4)
      Um = Um + U
    Next N
    Um = Um / 32
    Ux(i) = Um
    Print Um
   ' Locate 2 , 1
   ' Lcd Um
   ' Lcd " mV    "
   ' Waitms 500
  Next I
  Max(ux(1) , Um , I)
  D = 145 - I
  F = 16000000 / D
  F = F / 2
  Locate 1 , 1
  Lcd F
  Lcd " Hz      "
  Icr1 = D
  Ocr1a = D / 2
  Locate 2 , 1
  Lcd Um
  Lcd " mV    "
  If F > 57000 And F < 59000 And Um > 50 Then
    Waitms 50
    For N = 1 To 255
      U = Getadc(4)
      Um = Um + U
    Next N
    Um = Um / 127
    If Um > 50 Then Led1 = 1
    Lcd F
    Lcd " Hz       "
    Icr1 = D
    Ocr1a = D / 2
    Locate 2 , 1
    Lcd Um
    Lcd " mV    "
  Else
     Led1 = 0
  End If
  Waitms 1000
Loop
```

Gyrator-tuned Ferrite Antenna

Potentiometer replaces tuning capacitor

By **Martin Ossmann** (Germany)

Gyrator circuits can be used to simulate capacitors and coils. What's more, they can also be used as impedance converters, as described in this article. After that we will see how a gyrator can be used to tune a ferrite rod antenna.

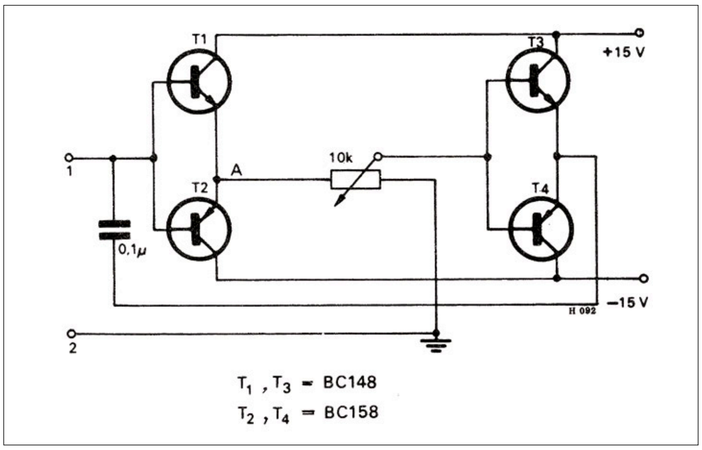


Figure 1. A 'variable capacitor' covering up to 100 µF, published in Elektor February 1975).

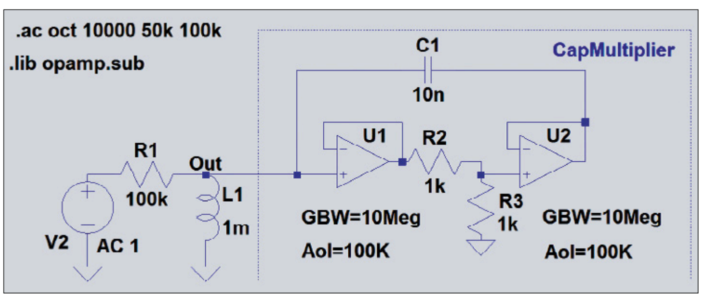


Figure 2. Simulation of the resonant circuit with a variable capacitor.

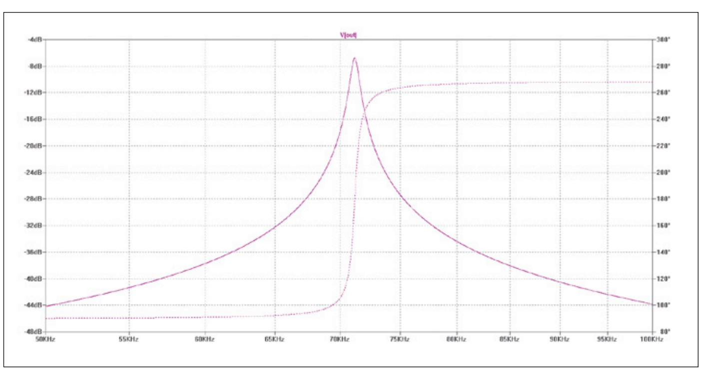


Figure 3. Frequency response of the RLC filter of Figure 2.

The author well remembers discovering a circuit for a variable capacitor as a teenager in an edition of Elektor back in 1975 (**Figure 1**).

Almost immediately it struck him that it would be possible to replace the variable capacitor in the resonant or tuned circuit of a single-stage AM receiver using instead a potentiometer or 'pot' for tuning. He tested his theory but it didn't work. Nevertheless, the dream remained of substituting a pot for a variable cap (even after varicap diodes became available had, albeit none with values as low as a couple of nanofarads—nF). With the limited resources of a schoolboy experimenter, I was unable in those days to discover why my substitution scheme had failed. Today, with better measuring technology and a more solid grasp of analog circuit technique, I can not only understand my lack of success then but also (with modern opamps) even achieve my dream from those days of tuning a resonant circuit with a pot, i.e. an adjustable resistor. The deliberations and experience involved are revealed in this article.

Capacitor substitution

To understand the circuit better we have simulated it in LTspice. **Figure 2** shows the schematic. With one resistor and one coil we can replicate a simple RLC filter with a single resonant circuit. In **Figure 3** we see the frequency response, simulating the behavior of a small signal by performing an AC Sweep.

In point of fact everything appears promising so far. The resonance curve is looking good. The circuit simulates, thanks to the voltage divider R2/R3, a capacitor of 0.5 × 10 nF. With L = 1 mH this results mathematically in a resonant frequency of around 71 kHz, which matches the simulation well.

However, if you construct the circuit your reward will be bitter disappointment. Instead of functioning as a selective filter, it just 'hoots' in lively fashion. This too is something you can reproduce in Spice with the help of transient simulation.

In **Figure 4** we see the exponential oscillation as the simulation result. At first glance this looks confusing, since the two op-amps in series achieve a total gain of just 0.5. But beware, appearances can be deceptive!

The capacitor C1 and coil L1 form a series resonant circuit, which raises the output voltage of the opamp U2 by the Q factor (also known as the quality factor or figure of merit), so that the oscillating condition is satisfied in the feedback. Owing to the principles involved, Spice cannot recognize this in small signal simulation. We must keep a close eye on this without fail: small signal simulation may well tell only the story when we are building circuits that are capable of oscillating.

Gyrator plus op-amps

For simulating capacitors and coils we can also employ gyrators. In this situation the gyrators are being used as impedance converters and that's exactly what we too are about to do now. Normally gyrators are used in four-pole circuits. Gyrators go back a long way in this magazine, right back to the very first German Elektor [2] (the magazine was not yet being published in English—*Ed*.) In **Figure 5** shows the conventional circuit of a gyrator made using opamps. The simulated impedance z comes about as follows:

$$z = \frac{z_1 z_3 z_5}{z_2 z_4}$$

For example, if we use a capacitor for z_2 and resistors for the remaining impedances, we can simulate an inductance. We then have:

$$z = j\omega \frac{R_1 R_3 R_5 C_g}{R_4} = j\omega L_g$$

From this we get:

$$L_g = K C_g \quad \text{with} \quad K = \frac{R_1 R_3 R_5}{R_4}$$

The configuration in **Figure 6** is also a gyrator. In this circuit the following holds good:

$$z = \frac{z_1 z_3 z_5}{z_0 z_4}$$

By adding capacitor C and a resistor we can enhance the circuit in Figure 5 to make a simple RLC bandpass filter (as shown in **Figure 7**). On the left we have the version using a gyrator and on the right, the equivalent circuit. We can now adjust the gyrator constant K using the resistors. In this way we can simulate capacitors or coils that are adjustable with a resistor.

If, for example, you alter the gyrator constants in Figure 7, then the value of the simulated inductance L_g changes and consequentially the resonant frequencies of the tuned circuit formed by C and L_g.

Simulation of gyrator circuits

Gyrator-based filter circuits are of course well-known nowadays. Three versions are shown in **Figure 8** as simulations in LTspice. The left-hand variant corresponds to the filter in [3]. The gyrator simulates an inductance and in the process an LC tuned circuit is created, with the input voltage U_{in} fed in via resistor R9. The output voltage U_{out} is extracted through the parallel resonant (high-impedance) circuit. Normally another impedance converter is used after this. The maximum gain at resonance is 1 (0 dB) and the Q factor is determined decisively, for example by resistor R9.

At the center we see the so-called DABP (Dual Amplifier Band Pass) filter from [4]. In this embodiment it is by no means so

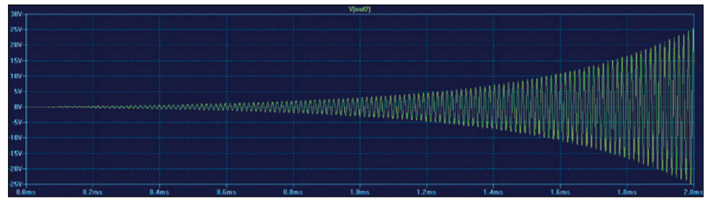


Figure 4. The transient simulation displays instability.

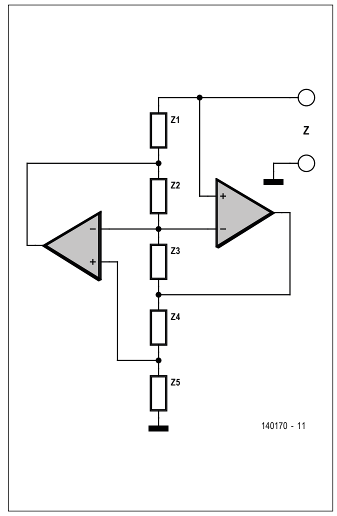


Figure 5. Gyrator with opamps.

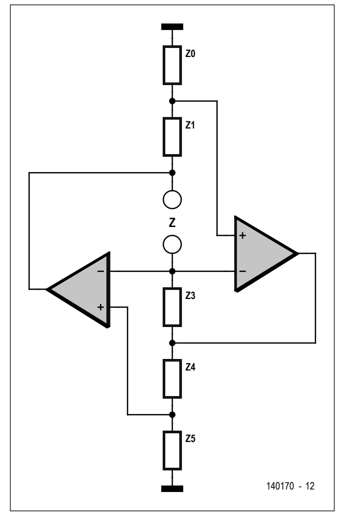


Figure 6. Alternative gyrator circuit.

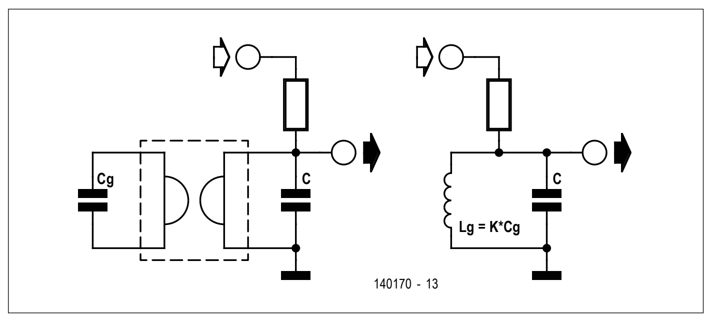


Figure 7. Construction of an RLC filter with gyrator.

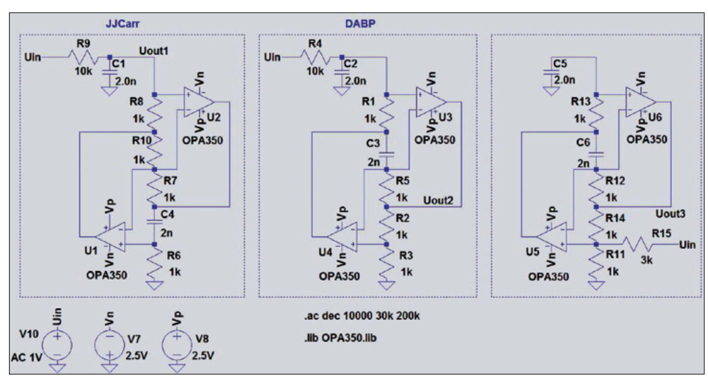


Figure 8. Three gyrator variants of the RLC filter.

easy to recognize this filter as using a gyrator. The difference from the circuit on the left is the (low-impedance) uncoupling at the output of the op-amp. Gain can be achieved in this manner too.
In the right-hand circuit the input voltage U_{in} is no longer fed to the resonant circuit via a resistor but instead coupled into the gyrator itself. This method is applied to the coupling of gyrator filters in [5] for example. In this way the *Q* factor is no longer attenuated by the input coupling resistor, enabling us to achieve significantly higher gain and *Q*. In one sample around 300 Hz bandwidth (3 dB) is achieved at a center frequency of 90 kHz, corresponding to a *Q* of at least 300.

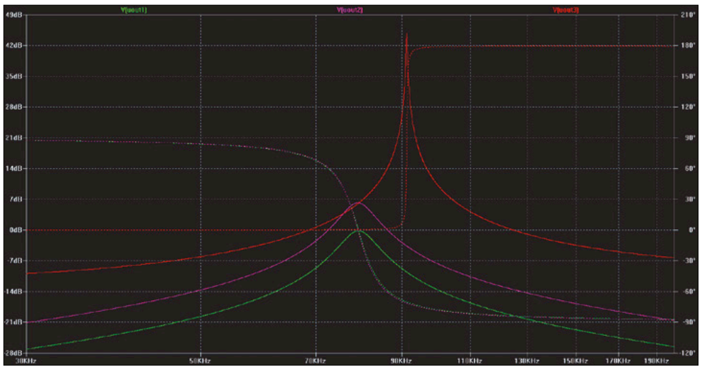


Figure 9. Frequency responses of the circuits in Figure 8 (green = U_{out1}, purple = U_{out2}, red = U_{out3}).

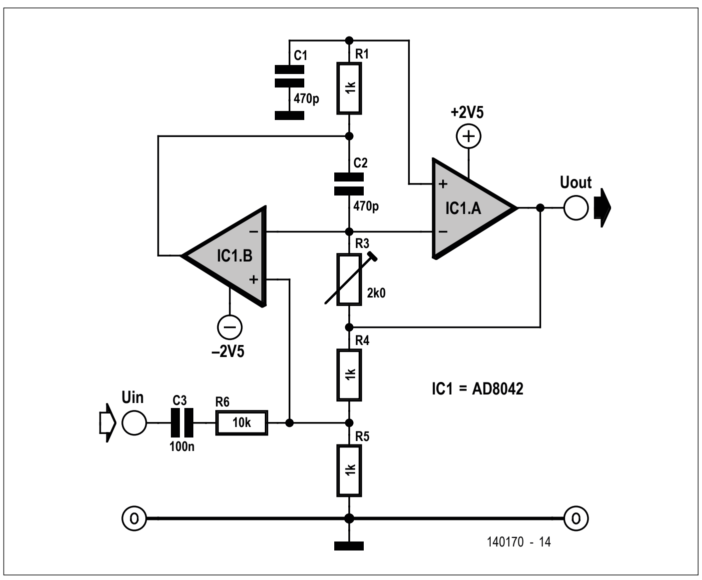


Figure 10. Practical circuit for a 455 kHz RLC filter.

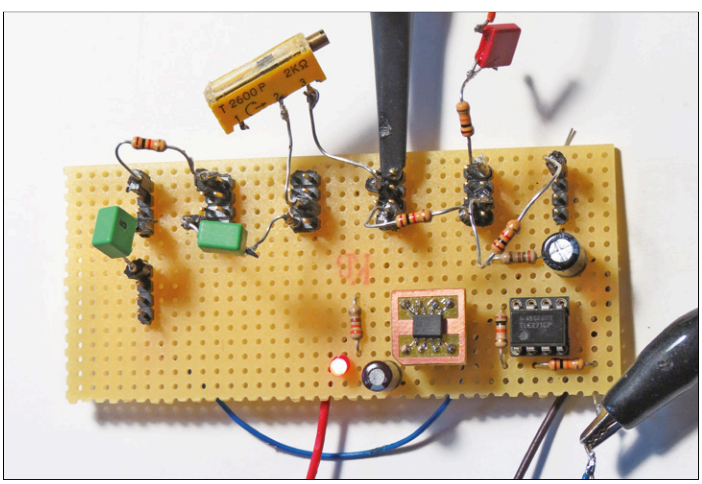


Figure 11. Breadboard for gyrator circuits.

Figure 9 shows the frequency response curves for the three filter circuits. The reason why the right-hand circuit possesses a higher resonant frequency lies in R15, which in small signal terms appears paralleled with R11 for the gyrator. This effect can be offset by altering R11.

Practical tests

Now it's time to put the circuit in Figure 8 (far right) to practical test at f = 455 kHz. The relevant schematic is shown in **Figure 10**. We are using a twin op-amp AD8042 with a gain-bandwidth product (GBW) of 160 MHz.

The author created a special breadboard (**Figure 11**) for these gyrator experiments. Individual impedances can be 'plugged in', making it possible to work your way through differing configurations.

In **Figure 12** we can see how the frequency response of the filter measures up. At a center frequency of *f* = 455 kHz the circuit achieves a 3dB bandwidth of 3 kHz. The figure of merit is around *Q* = 150 and must not be locked away behind LC filters. If the circuit is inclined to 'take off' on account of parasitic oscillation, some additional attenuation will assist (place a resistor in parallel with C1). This demonstrates definitely how gyrator filters of high *Q* can be constructed for the medium wave band using modern op-amps. Our original target of a resistor-tuned ferrite antenna still remains to be dealt with.

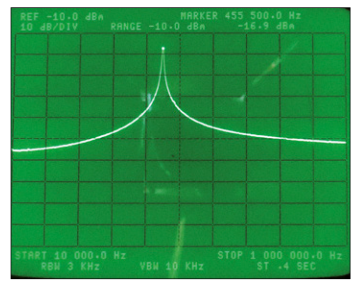


Figure 12. Frequency response of the 455 kHz filter.

Resistor-tuned ferrite antenna

In his quest for a tuning capacitor replacement the author had the following thoughts in mind: if you can create a gyrator from adjustable coils and resistors, it must also be possible to make a tunable ferrite antenna. A first attempt is shown in **Figure 13**.

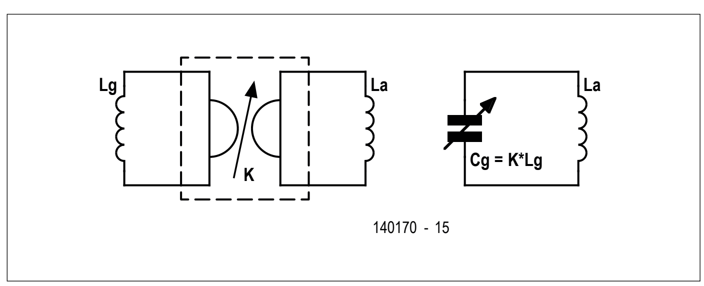


Figure 13. Simulation of adjustable capacitance.

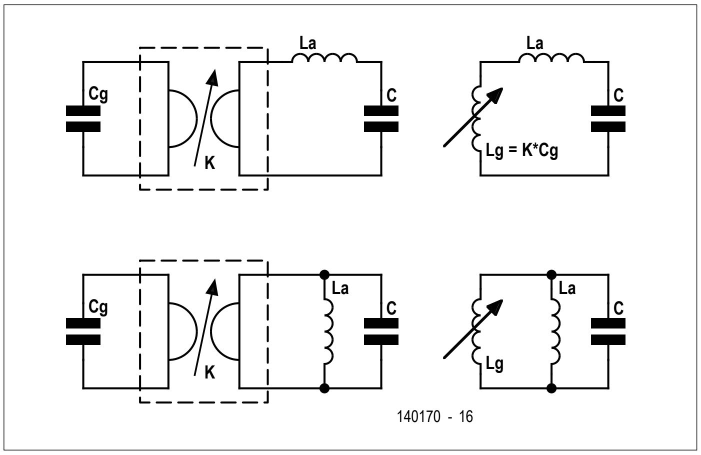


Figure 14. Using adjustable inductances.

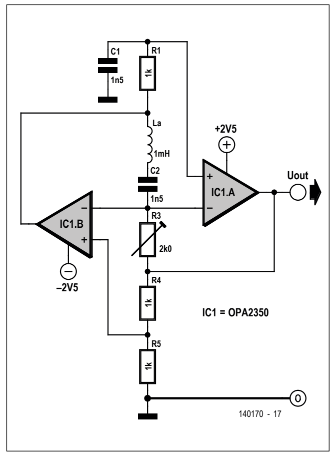


Figure 15. Tunable DCF77 gyrator ferrite antenna for VLF reception at 77 kHz.

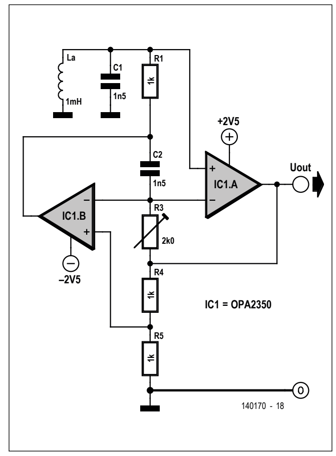


Figure 17. Alternative concept for 140 kHz to 200 kHz.

Using a fixed inductance L_g and an adjustable gyrator we can reproduced the resonant circuit capacitance C_g, which with the ferrite antenna L_a forms a resonant circuit. However, stability problems made this idea impractical.

An alternative approach was to make the gyrator simulate a variable inductance L_g again. This adjustable inductance can then be wired either in parallel or series with the ferrite rod, in order to be able to vary the resonant frequency of the tuned circuit with the gyrator constant *K*. **Figure 14** shows the circuits, the gyrator circuit on the left and the equivalent circuit to the right.

Our first practical circuit was dimensioned for receiving DCF77, the German counterpart of the WWV and MSF time signal transmitters. **Figure 15** shows the schematic. For the gyrator we used the version shown in Figure 6, in which z_0 is replaced by capacitor C1. The inductance created then forms a resonant circuit with L_a and C2. The oscilloscope shot in **Figure 16** is the DCF77 signal with its characteristic one-second pulse cycle clearly recognizable. You can adjust the resonant frequency simply with R3 and a bandwidth of 350 Hz is produced. This proves its practicability.

In **Figure 17** we have another variant. This time the inductance created is wired in parallel with L_a and C1. The resonant frequency is adjustable between 140 kHz and 200 kHz. **Figure 18** shows the signal received on 162 kHz from the French time signal station TDF. Good reception was also had of the BBC Droitwich on 198 kHz. A bandwidth of around 5 kHz was achieved. These two transmitters could also let you carry out experiments from the SDR tutorial [6]. The figure of merit achievable in the examples given is decisively dependant on the parameters of the op-amp. In this respect making a simulation before starting to solder components will help and it should also address any issues regarding stability.

With this done, the author fulfilled his goal of substituting a pot for a variable capacitor in connection with ferrite antennas.

(140170)

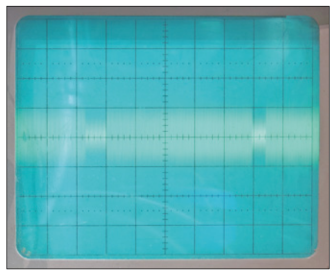


Figure 16.
DCF77 signal with its one-second pulse cycle.

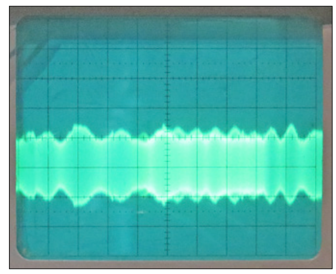


Figure 18.
Signal from time broadcast station TDF received on 162 kHz.

Literature

[1] Elektor November 1970 (German edition only): Variable Capacitance up to 100 µF, page 40.

[2] Elektor May 1970 (German edition only): The Gyrator, a wireless inductance, page 27.

[3] Carr, J.J., Secrets of RF Circuit Design, Tab Books, page 281.

[4] Linear Circuit Design Handbook, Analog Devices, Elsevier, page 640.

[5] Lynch, T.H., The right gyrator trims the fat off active filters, Electronics, July, 1977, page 115.

[6] AVR Software Defined Radio, Elektor E-Book.

From Knock Sensor to Stethoscope

By **J.T. van Es** (Netherlands)

In 2007 a friend of mine, who loves to build and tune racecar engines, needed a better way to listen to the sound in the engines, preferably from the control area of the test rig. "Why don't you listen to the output of the knock sensors?", I asked him, "that's what your ECU does." The look on his face told me that he started to doubt my sanity. But half an hour later he asked: "Is that possible?". "Not sure, but I can find out for you."

That day I took a handful of knock sensors back home with me. I then started to take one apart to find out what exactly was inside the black protective cover. This turned out to be quite tricky, since the molded cover was made out of a tough, hard plastic. I wasn't exactly over the moon with what I discovered (**Figure 1**): a ring made out of a piezo-electric material, firmly clamped between a couple of metal discs. An M8 bolt is used to fix the device to the engine block via a hole in the center. The capacitance of the ring turned out to be 1000 pF (in 2007 I couldn't find a datasheet for it, but now it's on the Internet [1]).

Figure 1. The knock sensor consists of a disc made out of a piezo-electric material, which has been sandwiched between a couple of metal discs.

A better way to listen to the sound in the engines, preferably from the control area of the test rig.

To get an idea of the type of output generated by the device, I stuck one onto a loudspeaker cone. The results looked pretty good, with the device acting as a microphone across a wide frequency range. It was something I could work with, although I still had to find out what its sensitivity was, and what the output voltage would be.

In **Figure 2** you can see the circuit diagram of the amplifier I designed for this sensor. Since the device is capacitive, I chose to use a voltage amplifier consisting of two stages made with standard FET opamps. Each stage has a gain of about 10, and the –3dB bandwidth is from 70 Hz to 36 kHz.

C1 and V1 in the circuit represent the (capacitive) sensor. R16 has been added for the benefit of the simulator, since it couldn't determine the potential difference between C1 and C7 without it. The function of C7 is to isolate the leakage resistance of the sensor from the opamp. C1 and C2 determine the gain of the first stage; R14 and R15 determine the gain of the second stage. C2/R1 determine the lower cutoff frequency of the bandwidth. R1 should really have had a value of 33 MΩ, but I didn't have one available. Instead, I added a potential divider (R17/R18) to the output of the opamp, which reduces the signal to R1 to 1/3 of its value. R7 and C1 determine the higher cutoff frequency of the bandwidth.

The measurements made on the amplifier corresponded to those of the simulation. Once the sensor was connected, it turned

Knock sensor

A knock sensor is used in many cars to prevent knocking of the engine (detonation as a result of uncontrolled and premature combustion of the fuel). The engine vibrations are first recorded during the normal combustion process. When the engine starts detonating, stronger vibrations will occur in a certain frequency range. The signal from the knock sensor is filtered and processed by the engine management system (ECU), which then uses this information to determine if the engine is knocking or not. When knocking is detected, the ignition timing is delayed in steps of about 3° prior to TDC. Once the detonation has stopped, the ignition will return to its normal setting in steps of about 0.5°.

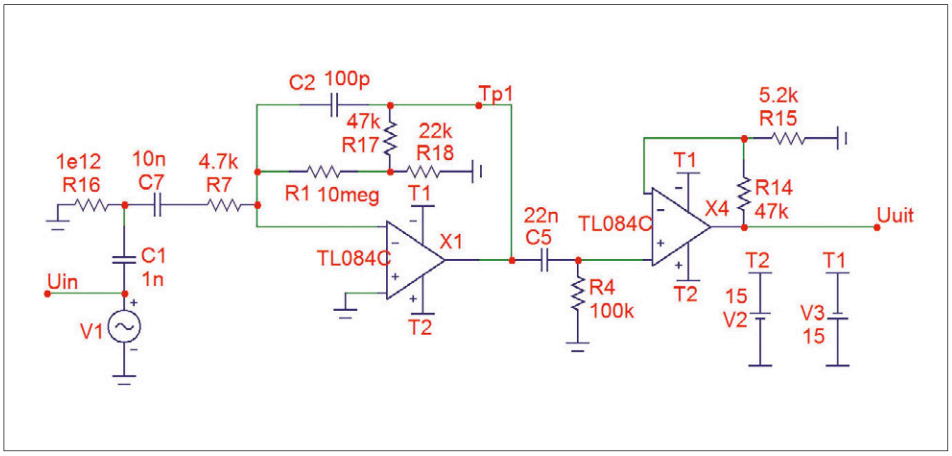


Figure 2. The amplifier circuit consists of two stages, which together provide a gain of 100.

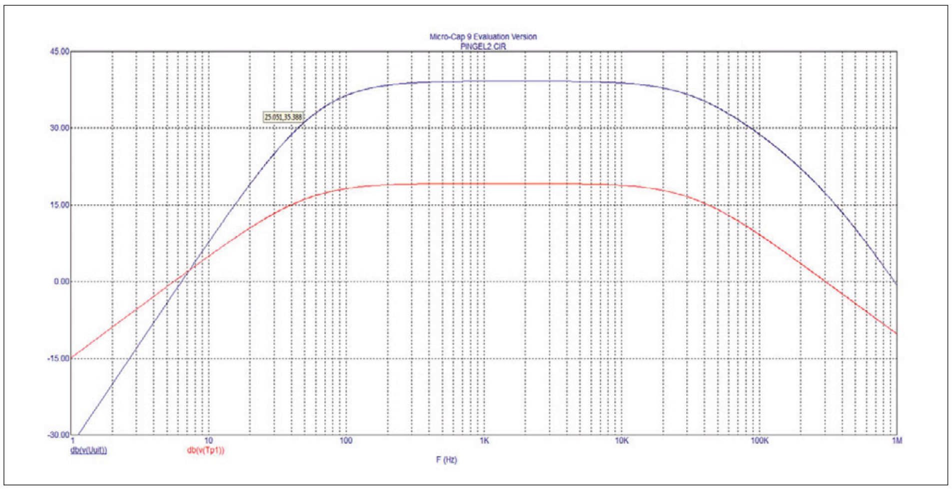


Figure 3. The simulation of the circuit results in this frequency response (red = output of first opamp, blue = output of second opamp).

out that I had created something highly sensitive. In fact, it was much too sensitive for my friend. He had no need at all of the amplifier! The enclosure (see **Figure 3**) disappeared into the junk box. Now, in 2014, I discovered it again and carried out a few more measurements. The noise at the output is 240 µV (which corresponds to 2.4 µV at the input).

Securely wrapped in aluminum foil the sensor is now clamped in my table vice. My scope is set to 10 mV/div. Just touching the table shows an intense response. Dropping a match from an inch above the table results in an output of 40 mV. Even speech became visible. That is an incredible sensitivity for a sensor that could only be damaged using extreme force, such as swinging an axe at it. Just about every fairly modern car will have at least one of these, which means you should easily find one at the salvage yard. It is an interesting component to experiment with!

Now to think of some applications for this level of sensitivity. A pickup element? A stethoscope? For now I'll just put it back into my junk box.

(140324)

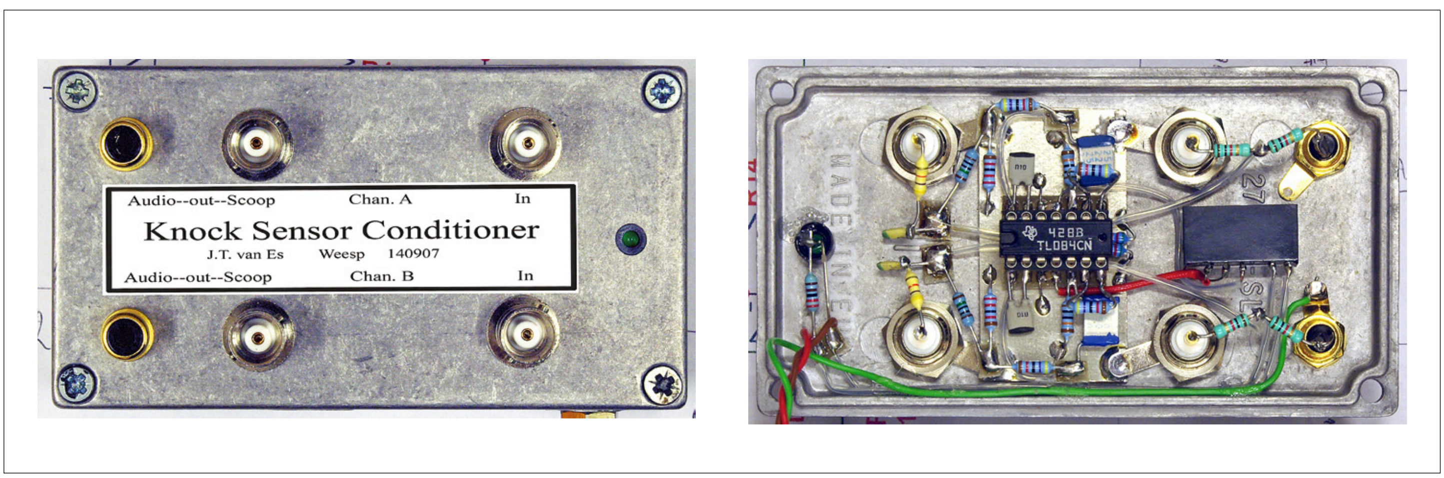


Figure 4. The circuit built by the author was put inside an aluminum enclosure to provide optimum shielding.

Web Link

[1] www.bosch-motorsport.de/media/catalog_resources/Knock_Sensor_KS-R_Datasheet_51_en_2779074187pdf.pdf

Siphonic Rain Gauge

No moving parts, and compatible with USB Weather Logger

By **Paul Cordonnier** (Belgium)

Meteorologist electronics technicians don't trust mechanical apparatus, which are subject to wear and hence inaccuracy. However, in the field of precipitation detectors, there's no great rush of innovators. So that's why we didn't hesitate for long when we received this suggestion from a country where they know all about precipitation.

The circuit in three p(h)rases

- The reservoir of the no-moving-parts rain gauge is fitted with a siphon, which empties it automatically every time the water collected reaches the required level.
- In order to establish accurate, reliable pluviometry, all you need do is measure once only the volume of water in the siphon, then count the number of times the siphon operates and note the time they happen.
- This rain gauge with no moving parts can be associated with the Elektor USB Long-Term Weather Logger, the firmware for which has been adapted.

Weather stations are among the favorite subjects for the average electronics technician— which is kind of strange when you think about it, since rain or fine weather have virtually zero effect on how they practice their craft. Might all this interest be explained by an obsession with everything that can be measured accurately? One thing is sure—electronics technicians don't trust mechanics. Moving parts bring them out in a rash and make them redouble their ingenuity to try to avoid needing them. We've seen other examples only recently in Elektor with various air flow measurement circuits to finally do away with weather-vanes and cup anemometers.

Paul Cordonnier has managed to simplify the construction of his rain gauge (also called "pluviometer") to the point of totally eliminating moving parts, without sacrificing either the reliability or the accuracy we are entitled to expect from such an accessory.

Contrary to what you might think, there are a good many problems to be solved to obtain an accurate rain gauge.

We won't go into details here [2], but let's just remember that a rain gauge is an instrument used to measure the height of precipitation during a given period. By height of precipitation, we mean the thickness of rainwater that would have covered an unobstructed horizontal surface if the water that fell had neither soaked in, run off, nor evaporated. It is expressed in millimeters or in liters per square meter (1 mm = 1 l/m²).

To replace the usual system* of tipping buckets used in current rain gauges, the principle adopted here is that of the siphon, *but with no mechanics*. At a siphon is a curved tube (*siphôn* in Greek) that transfers a liquid from a given level to a lower level, passing via a level that is higher than the other two.

Our siphonic detector is formed from a small-diameter (e.g. 5 mm) flexible plastic tube, dipped into a reservoir or test tube (**Figure 1**), topped with the collecting funnel. The test tube is simply a rigid tube a few centimeters in diameter, closed at the bottom by a stopper, through which the bent flexible siphon tube passes. The rain collected by the funnel raises the level of liquid in the reservoir. When this level reaches the bend in the siphon, it is primed and empties the contents of the test tube: the water flows out to the tube outlet. This quantity of water is always the same, to within a few drops. So all we have to do is count the number of times the siphon is primed in order to find out the mass of water collected. To sum up, as in a tipping-bucket rain gauge, it is the passage of a known quantity of liquid that enables us to calculate the level of precipitation, by simple arithmetic. But here, it's not the

* *there are also weighing rain gauges, in which the mass of the precipitation collected is weighed.*

movement of the bucket that is detected, but of the water itself!
As it passes through the siphon discharge tube, the water creates a short-circuit between two fine stainless-steel electrodes that pass through the walls of the flexible tube underneath the test tube. This is where the detector stops and the electronics begin. The two electrodes are connected to K2 in the circuit in **Figure 2**; when the non-inverting input of comparator IC1 is taken low by the liquid passing between the electrodes, its output changes state. And that's where the description of our rain gauge stops! It is connected to the input of a computer (or some other programmed circuit) which will be used to exploit this raw siphon signal.

Drains

If you've looked closely at Figure 1, you'll have noticed a strange detail in the branches of the siphon, upstream and downstream of the electrodes, in the form of a fine line. This represents a kind of wick, made from a simple fine cotton thread, threaded into the tube in order to drain the flow of residual water droplets each time the siphon empties; without this, droplets remaining between the electrodes could cause a short-circuit. If this continued till the next flush, this wouldn't be indicated by the flipping of the comparator and the count would be falsified.
For the same reason, experience has shown it is useful to have a piece of thread in the upward branch of the siphon. This eliminates the droplets of water remaining in the siphon after it empties, which are liable, when next it rains, to upset the rise of the water in the siphon.
Do note that under no circumstances must these drains run either at the level of the electrodes, or around the bend (risk of self-priming of the siphon (by capillary action) during a slow rise in level).
When this rain gauge was presented on the Elektor Labs website [1] prior to being published here, user David A., a supporter of the capacitive approach, well-known to Elektor readers, suggested that two electrodes could be glued either side of the external walls of the tube to form a capacitor; would the passage of the water cause a disturbance to the capacitance such that it could be reliably detected? This would certainly be worth experimenting on, which remains to be done. In this event, the mechanics of the electrodes would be simplified, that's for sure, but not the drive circuit, as it would probably be necessary to go over to AC.
In any event, the siphon principle is as old as the hills, and there are even siphonic rain gauges that use a moving part, but it seems that to date no-one has thought of using the direct measurement without mechanics (other than fluid mechanics) presented here.

We won't go into detail of construction. The hardware needed can easily be found in the plumbing section of a big DIY store:

- funnel with filter (watch out for dead leaves);
- approx. 20 cm (8 inches) of plastic tubing around 2–3 cm diameter (1 inch), with stopper;
- approx. 1 m (3 feet) of 5 mm (3/16 inch) plastic tubing;
- electrodes, cotton.

The bend in the flexible tube forming the siphon mustn't distort it too much, at the risk of obstructing the passage of the water. All you need do is dip it into hot water to soften it at the moment of bending it (gradually).

Rain Gauge and USB Long-Term Weather Logger

We thought it would be a good idea to combine this unusual detector with Elektor's well-known **USB Long-Term**

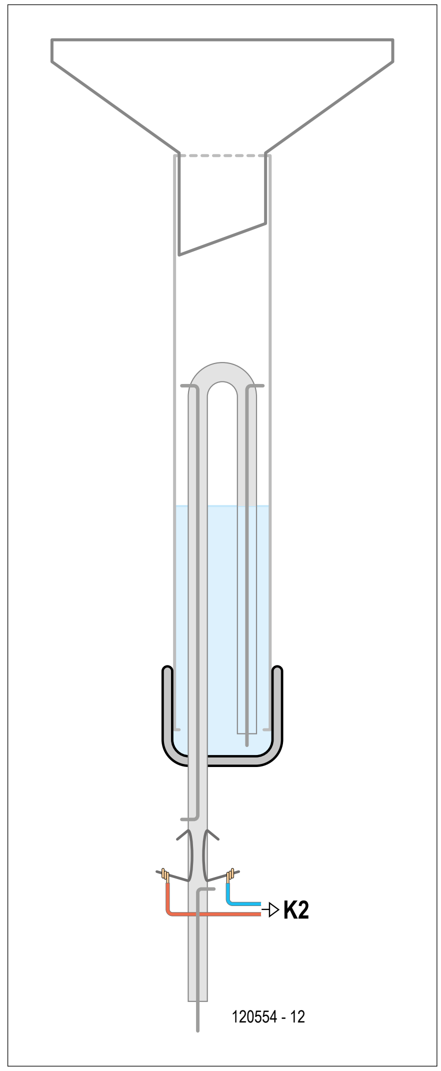


Figure 1. This rain gauge uses the siphon principle, making it possible to dispense with any moving parts.

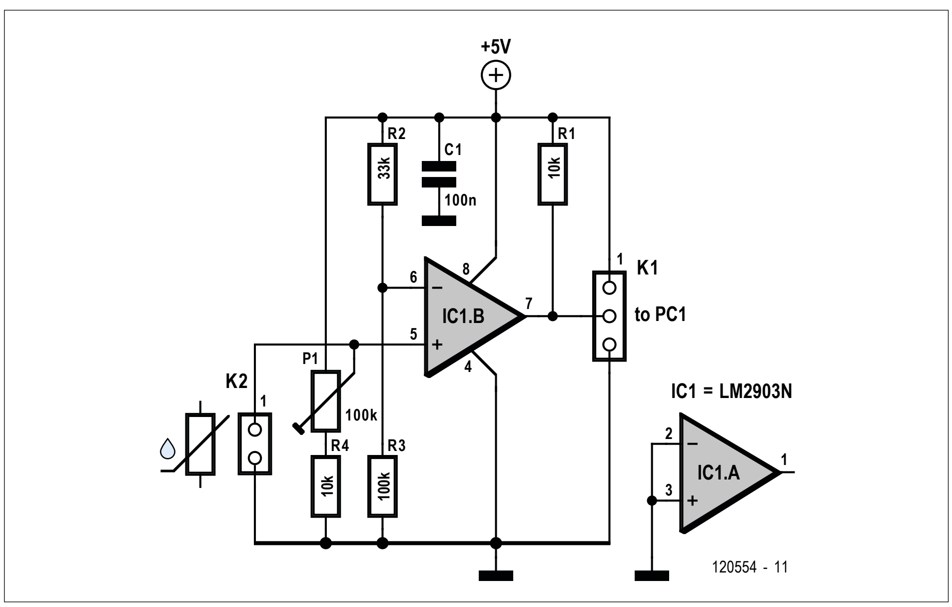


Figure 2. The detector circuit diagram is reduced to a single comparator, whose switching threshold is adjustable.

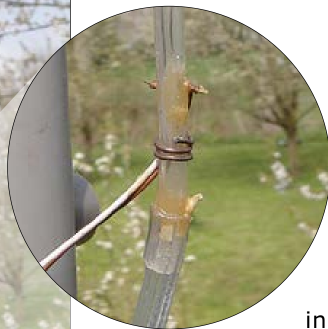



Figure 3. These are the two electrodes at the siphon outlet that detect the emptying.

Weather Recorder [3]. Using a single battery, this self-contained data recorder is capable of accumulating some six to eight weeks of data furnished by I²C detectors for atmospheric pressure, temperature, and humidity, and displaying this on demand on a two-line LCD (**Figure 4**). The results, collected via a USB port, can be plotted graphically on a PC using a program like, for example, GNUplot. There is now a version of the firmware for this recorder [4] suitable for the rain gauge described here. This version does not support the other detectors (temperature, humidity, and pressure). The output of the comparator in Figure 2 is applied to pin PC1 of the ATmega168 in the recorder. The use of a DCF77 (time over-air) module for time and date stamping the result is an interesting option offered by the recorder.

Getting rid of the bubbles

Each time the siphon operates, the recorder's firmware stores the time and date in the EEPROM. The state of the detector is examined once per second. If water is flowing, a counter is incremented; if it's beer or champagne… oops, sorry! As bubbles will form at the end of the emptying, a delay has been incorporated into the software to eliminate unwanted counts as these bubbles pass the electrodes. If after 10 s it is still flowing the time and date are recorded in the EEPROM. This avoids erratic counting of stray drops or thin streams of water. The duration of 10 s corresponds to the capacity of the author's reservoir and the siphon time. Depending on the size of your own reservoir and the length of the pipes, you will need to lengthen or shorten this 'gurgling' time.

For the rest, the operation of the USB weather recorder is unchanged. You must start by telling it the time and date, either manually, or using a DCF module. Then the display indicates the time and number of siphons (N= xxx) on the first line and the date on the second line. As soon as the detector detects siphoning, the display shows the message "Flushing" on the 2nd line, followed by the duration of the current operation. After 10 s, the time and date of the current operation are recorded in the EEPROM, but the counter continues to operate while water is still flowing. In the event of a short on the electrodes, no provision has been made to stop the counter. So nothing will happen while the short remains.

Press S1 (left-hand button) to request a new recording session, then S2 to confirm… or S3 to return to the current session… or S1 to access the menu for the USB link with the PC. On the computer, all you have to do is program the terminal as a Hyperterminal, using the following parameters: 9600, N, 8, 1, no flow control. Type 'H' + ENTER in the terminal program to obtain the following information:

- **h** help
- **a** reads the number of siphons, current date and time
- **c** clear: initializes new session
- **p** print: transmits all the recorded events to the PC
- **x** exit: interrupts the serial link, recording continues

Press S1 three times in succession to see the recorded events displayed one after another at 2 s intervals on the second line of the display.

(120554)

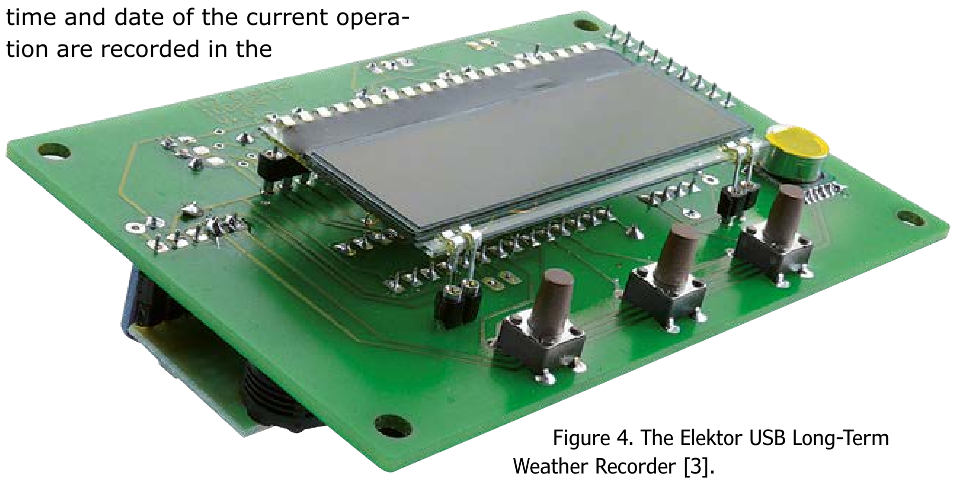


Figure 4. The Elektor USB Long-Term Weather Recorder [3].

Web Links

[1] www.elektor-projects.com/node/2574

[2] https://en.wikipedia.org/wiki/Rain_gauge

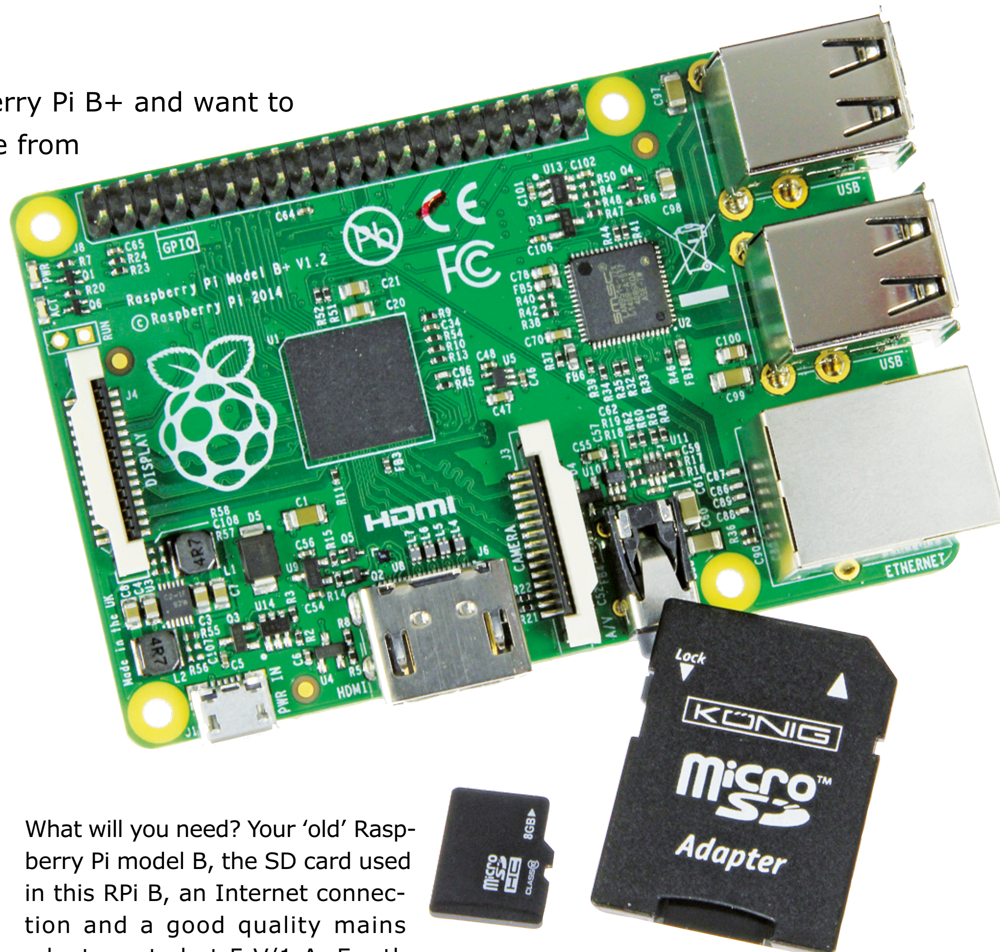
[3] USB Long-term Weather Logger, Elektor September 2011: www.elektor.com/100888

[4] New version of the firmware: www.elektor-magazine.com/120554

SD Card Upgrade for the RPi B+

By **Bert van Dam** (The Netherlands)

If you've recently bought a Raspberry Pi B+ and want to use your SD card with the software from the previous version you run into two problems. The first is that the model B+ has a micro SD card slot. The second is that the original software isn't (completely) compatible with the B+, meaning that it won't boot up. Fortunately, you can get round these problems fairly easily!



The B+ has a number of improvements compared to the B. These are the result of a change in some of the hardware components, which means that the software has to be adapted. You can of course download an up to date Linux version from the Raspberry Pi website [1] and store it on a micro SD card. This will get the B+ working without any problems. But there are many RPi users who already have an SD card with the operating system and many other programs with which they've become familiar (for example, the SD card for the Elektor book 'Raspberry Pi — Explore the RPI in 45 electronic projects' [2]). Fortunately, it's not difficult to upgrade the SD card, so it can be used on a micro SD version in the B+ model.

This upgrade consists of two stages. The first stage is to upgrade the operating system. This stage is identical for all SD cards that have the Debian Wheezy distribution. The second stage is the adaptation of files and programs that no longer work properly. This stage obviously depends on the installed programs and will be different for all SD cards. In this article we'll be using the SD card for the Elektor book [3] to demonstrate how to upgrade it.

What will you need? Your 'old' Raspberry Pi model B, the SD card used in this RPi B, an Internet connection and a good quality mains adapter rated at 5 V/1 A. For the new B+ you'll need a 4 GB micro SD card (class 4 or better), along with an SD card adapter, see main photo.

Preparations

Create an image of your SD card on your Windows PC with the help of an SD card reader and the program DiskImager, which is part of the free download for this article [4]. Next, plug the SD card adapter with the micro SD card into the reader and use DiskImager to copy the saved image to the micro SD card.
Plug the adapter with the programmed micro SD card into your 'old' RPi B and connect it to the Internet. You should use a cable for this, rather than a wireless connection, since it is vital that you don't lose the Internet connection during the upgrade. You can use a keyboard and screen with the RPi, but it is easier to use it in a 'headless' mode.
This means that you'll be using a Windows PC to control the Raspberry Pi via a program called Putty (also part of [4]). This way you won't have to manually type in the commands later on, as you'll be able to copy and paste them from the download for this article.

Start Putty and enter the IP address of your Raspberry Pi. (If you don't know the IP address you'll have to log on to your router. This has a list of all connected devices, including your Raspberry Pi). Verify that port 22 has been selected and that SSH has been ticked. Now click on 'Open'. You may get a message from Putty stating that this IP address has not been used before, and if it is correct. Click on 'OK'. After a while you'll be connected to the Raspberry Pi and you'll be asked

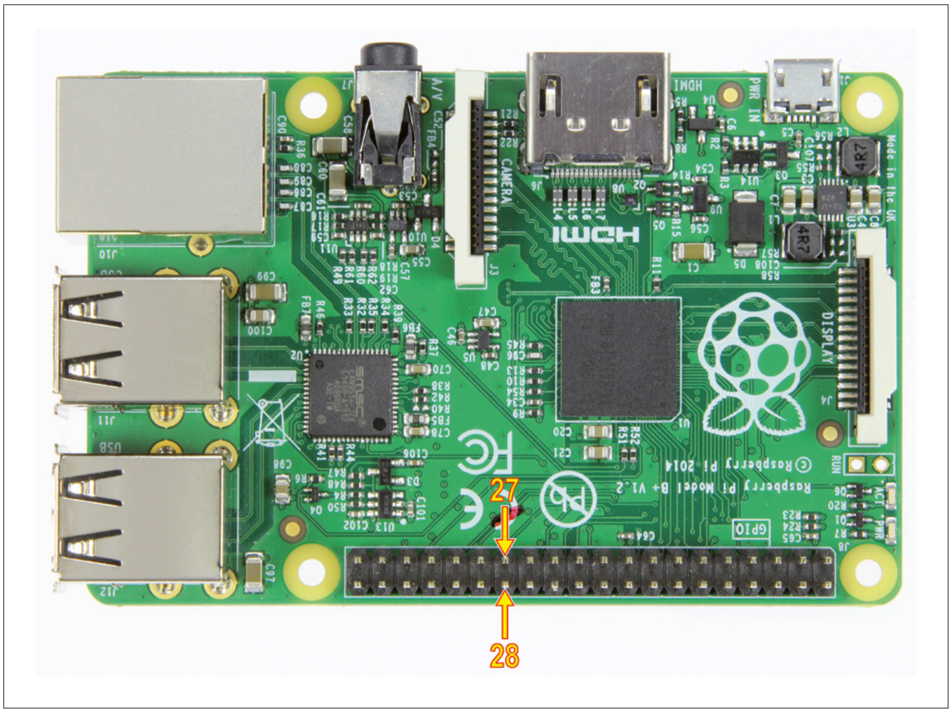


Figure 1. The GPIO connector is longer on the B+, which can cause two of the pins to be in the way on some 26-pin connectors.

> The original software isn't (completely) compatible with the B+, meaning that it won't boot up. Fortunately, you can get round these problems fairly easily!

for your user name and password. The defaults for these are:

```
name     pi
password raspberry
```

Note that nothing will be displayed while you type in your password, not even stars! It appears as if nothing is happening, but rest assured that it's working normally.

The conversion: stage 1

This stage can be used for all SD cards that have the Debian Wheezy distribution. Use Putty to log on to the Raspberry Pi and enter the following command:

```
sudo apt-get update
```

You can copy this command (and all other commands in this article) from the text file 'commands.txt' in the download [4], and paste it in the Putty window by clicking the right-hand mouse button. Then press Enter. This command updates the tables in the Raspberry Pi, so it knows where certain files and programs can be found on the Internet. This can take several minutes. Once this has completed, enter the following command to start the actual upgrade:

```
sudo apt-get upgrade
```

You should answer Y (yes) to all the questions. Note that this command can take a long time to complete (45 minutes isn't unusual) and requires a lot of power. If you have a weak power supply it's quite possible that the RPi will hang during this process. For this reason the use of a 1-amp supply is recommended. When it has finished, reboot the Raspberry Pi with the command:

```
sudo reboot
```

If you're using Putty you will lose the connection at this point. You should wait until the Raspberry Pi has started up and then log in again.
The conversion is now complete and the micro SD card is suitable for use in the Raspberry Pi B+. Most of the software will still work, but there are exceptions (as you'll see from the modifications described below). It is recommended that any programs you've installed yourself are tested thoroughly and adapted if necessary. The exact changes required will depend on the software you're using.

The conversion: stage 2

As an example we'll show you what changes are required to adapt the micro SD card used with the book 'Raspberry Pi - Explore the RPI in 45 electronic Projects'. Anything that isn't mentioned will work without modifications. All of the source code from the book will still be on the micro SD card.

1. Extra pins

When you use (IDC) connectors to attach electronic projects to the Raspberry Pi B+ there is a chance that they will no longer fit because the GPIO connector has become larger. Fortunately, the first 26 pins are identical. The next two extra pins (**Figure 1**) may be in the way (this depends on the type of connector you use, some types with solder connections will fit without requiring any changes). These pins are ID_SD (pin 27) and ID_SC (pin 28), and are exclusively for use with a Pi-Plate I²C ID EEPROM. In the future, any Pi-Plate (expansion module) that is plugged in can be automatically recognized. It's possible to bend these pins out of the way (or even cut them off) if you don't intend to use Pi-Plates in the future. However, a better solution would be to use the proper, larger connectors.

2. Keyboard

During the upgrade the keyboard settings will have been overwritten. The following command calls up the software configuration tool:

```
sudo raspi-config
```

From here you should use 'internationalization options' to select your locale and keyboard. Note that the new keyboard will only be active once the Raspberry Pi has been restarted (using `sudo reboot`).

3. Sound

It's possible that the amixer will be muted after the upgrade, so you won't hear any sounds. You can turn the mute off using the following command:

```
amixer set PCM unmute
```

4. IdleX

The file 'Idle' is overwritten during the upgrade, with the result that 'Idle' starts up instead of 'IdleX'. Open 'Idle' in the nano editor with the command:

```
sudo nano /usr/bin/idle
```

The cursor will be at the start of the file (**Figure 2**). Press the DEL key and hold it down until all the text has been deleted from the window. Copy the text from **Listing 1** (which can also be found in the download) and paste it into the editor using the right-hand mouse button. If you're controlling the RPi directly via a keyboard and screen, rather than via Putty on a PC, you will have to carefully type in this listing yourself. Don't try to use the mouse for scrolling; you should use the cursor keys for this.

Save the file using Ctrl-O, Enter, and Ctrl-X.

That's it!

The new micro SD card for the Raspberry Pi B+ is now ready for use. Stop the Raspberry Pi B with the command:

```
sudo shutdown -h now
```

Remove the micro SD card from the adapter, insert it into the Raspberry Pi B+, turn on the power supply and you're ready to go!

(140342)

Web Links

[1] www.raspberrypi.org/downloads/

[2] www.elektor.com/raspberry-pi

[3] www.elektor.com/sd-card-with-software-belonging-to-the-book-raspberry-pi-129025-81

[4] www.elektor.com/140342

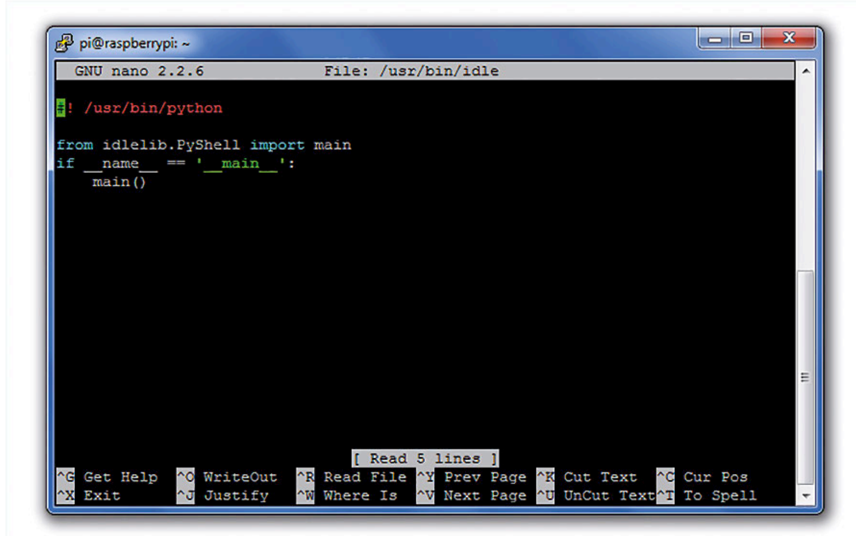


Figure 2. The (incorrect) Idle file, opened in Putty. This text should be removed and replaced with that from Listing 1.

Listing 1.

```
#! /usr/bin/env python

# Launch IdleX
import sys

def show_error():
  if sys.version < '3':
      import Tkinter as tk
      import tkMessageBox as messagebox
  else:
      import tkinter as tk
      import tkinter.messagebox as messagebox

  root = tk.Tk()
  root.withdraw()
  messagebox.showerror(title='IdleX Error',
     message=('Unable to located "idlexlib".\n' +
       'Make sure it is located in the same directory ' +
       'as "idlexlib" or run setup.py to install IdleX.\n' +
       '   python setup.py install --user'))

try:
  import idlexlib
except ImportError:
  show_error()
  sys.exit(-1)

from idlexlib.idlexMain import main
main()
```

350-V Step-Up Converter

High voltage from a 30 V bench power supply

By **Martin Ossmann** (Germany)

Adding this push-pull converter to the output of a standard bench top power supply gives a DC output up to 350 V DC. The output voltage and current of this 50 W booster design is controlled by the voltage and current limit controls of the bench supply.

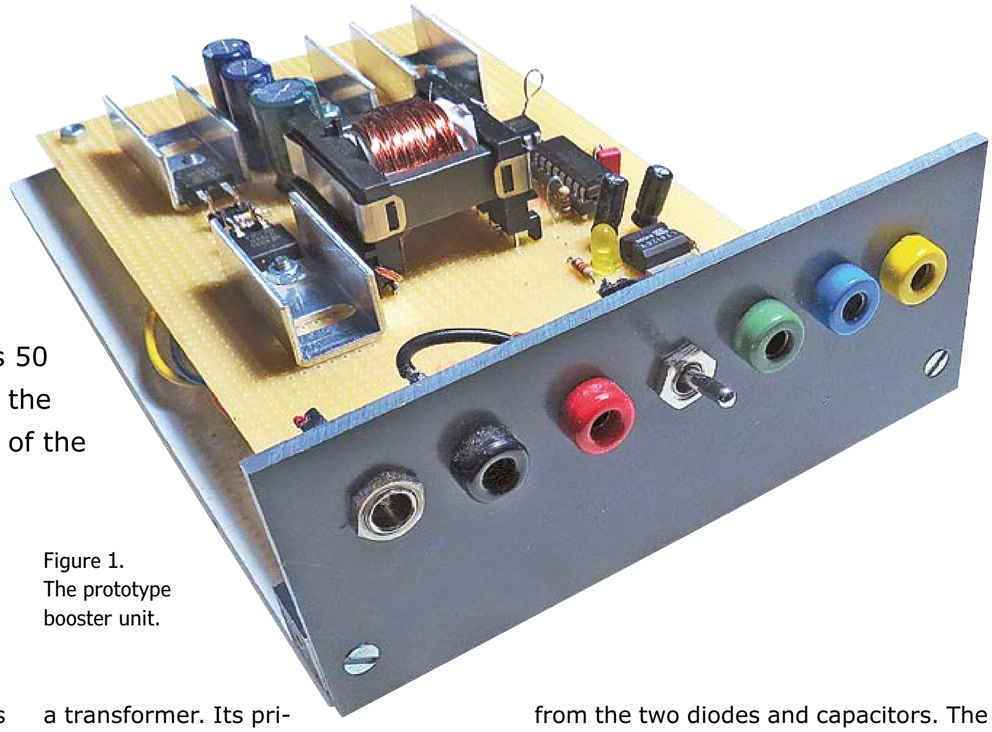


Figure 1. The prototype booster unit.

A benchtop variable DC power supply is probably one of the most useful items of equipment on any test bench. There are many different designs to choose from, a typical supply gives an output up to 30 V and a current of 3 A. In contrast, high voltage power supplies are few and far between; their high price makes it difficult to justify the outlay.

This low-cost DC booster should help plug the gap; it multiplies the output from a bench supply by a factor of 6 or 12. The design is good for 50 watts output power at a maximum voltage of 350 volts. To simplify the circuit, the booster's output voltage is controlled by adjusting the bench supply's output voltage, likewise with current limiting. To add to its versatility the input and output voltages are electrically isolated.

The schematic shows that the booster design consists mainly of a simple and robust push-pull converter. We go on now to look at some background theory. Using the same principles, you can then adapt the design to suit other applications.

The push pull principle

The circuit's operating principle is shown in **Figure 2**. The central component is a transformer. Its primary side consists of two windings N_p while the secondary windings are labeled N_s.

The switches S1 and S2 alternately switch the input voltage U_{in} through the two primary windings. The output voltage induced in the secondary winding is rectified by a voltage doubler circuit made from the two diodes and capacitors. The output voltage $\pm U_{out}$ is given by:

$$U_{out} = N_s / N_p \times U_{in}$$

One advantage of the push-pull converter is that with a low loss design the output voltage is largely independent of load. Under no load, the output is also stable so there is no requirement for a voltage regulator.

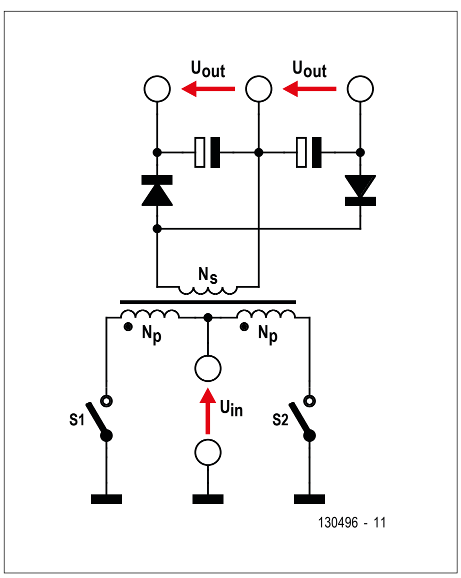


Figure 2. The push-pull converter principle.

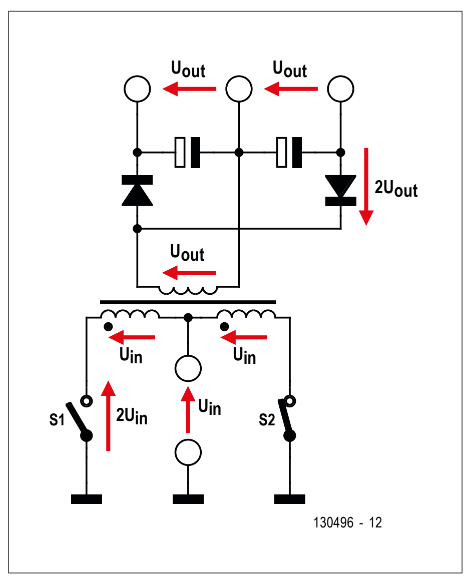


Figure 3. Push-pull waveforms.

Voltages

Looking at the voltage waveforms present on components around the circuit gives an appreciation of the booster design and its performance characteristics.
With switch S2 closed (**Figure 3**) the primary winding on the right of the diagram is switched to the input voltage U_{in}. Its close coupling on the transformer produces an induced voltage in the left hand winding. This results in double the input voltage across this switch (Transistor). It's important to keep something in reserve to withstand voltage spikes caused by signal overshoot. We have specified an IRFL540 rated at 100 V so an input voltage of up to 40 V is well within its capabilities.

This picture makes it clear what would happen if the second switch is closed at the same time: it would result in a short-circuit of this (twice the input) voltage. For this reason it's essential that the switches are controlled so that their ON times do not overlap.

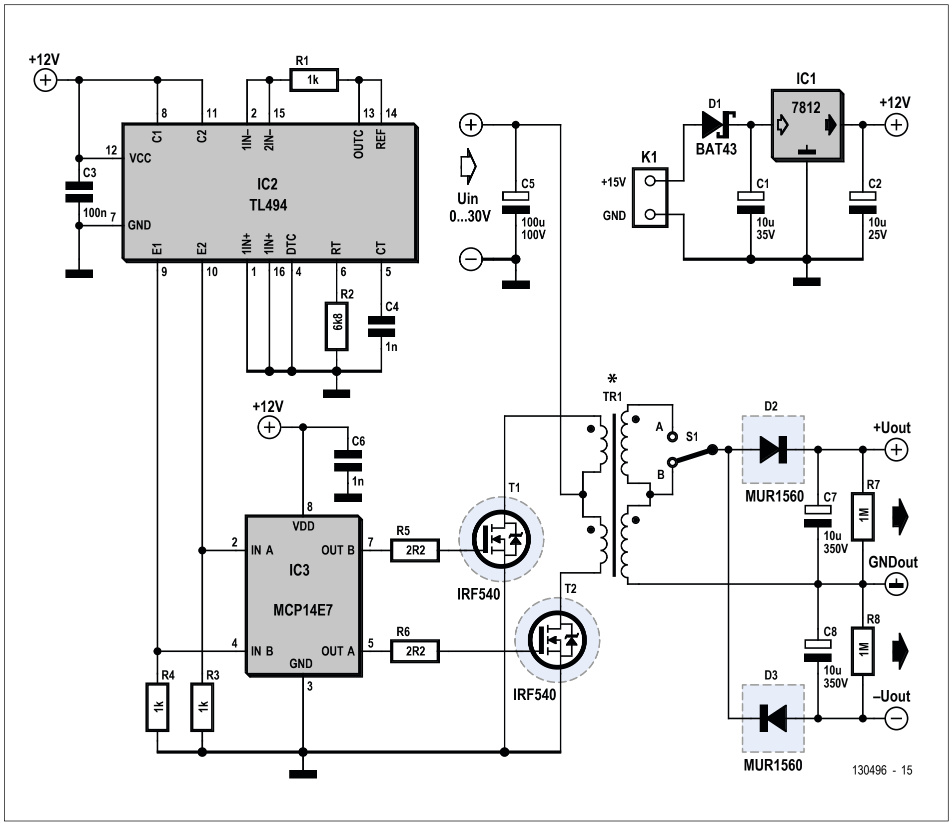


Figure 6. The DC booster schematic.

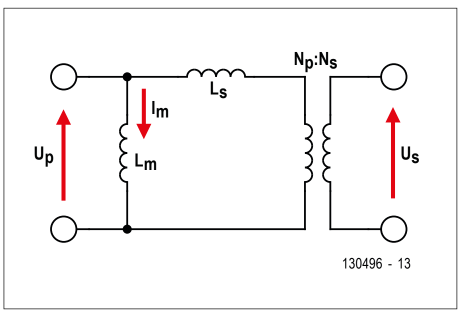


Figure 4. The transformer equivalent circuit.

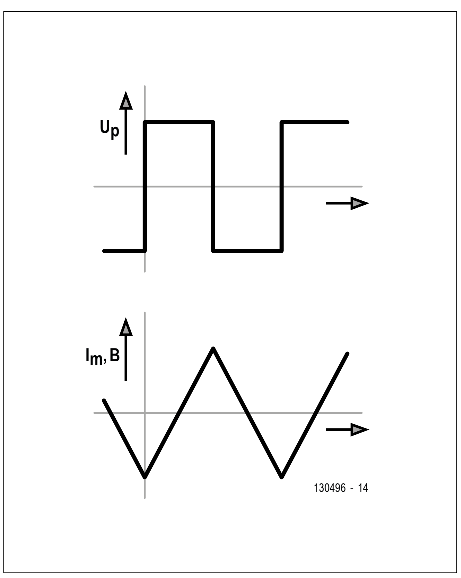


Figure 5. Voltage, excitation current and induction.

Moving to the secondary side of the transformer, when S2 is closed the diode on the left conducts and the right hand diode has $2U_{out}$ reverse voltage across it. In this design we use a MUR1560 with a maximum reverse voltage of 600 V. This allows rectification of an output voltage of ±250 V (i.e. 500 V total).

Transformer saturation is dependant on the voltage. **Figure 4** shows an equivalent circuit diagram. The waveform U_p at the primary winding is a square wave of value $\pm U_{in}$ with a mark-space ratio of 1:1. The voltage induces an excitation current I_m in the main inductor L_m, represented in **Figure 5** by the triangular waveform. The corresponding core flux density B_m has a triangular waveform with a peak value of B_{pk}. From the induced voltage (the input voltage) and applying the law of induction we can find the change of $\Delta B/\Delta\tau$ with time:

$$\frac{\Delta B}{\Delta \tau} = \frac{2\,B_{pk}}{T\,/\,2} = \frac{U_{in}}{N_p\,A_e}$$

... Where $T = 1/f$. The square wave period of frequency f and A_e is the effective cross sectional area of the core. From this:

$$B_{pk} = \frac{U_{in}}{4 N_p\,A_e\,f}$$

For our purposes we have used an ETD29 core, with A_e = 76 mm². The frequency f = 80 kHz with N_p = 8 turns. With an input voltage of 30 V:

$$B_{pk} = 150 \text{ mT}$$

Typical core material (N30 or similar) can handle a peak flux density of around 300 mT without saturating. At 30 V input there is still enough in reserve but at 50 V we start to approach the saturation threshold.

Current

Whenever current flows through a circuit it is subjected to a series of losses. A current of 5 A flowing through the MOSFETs and the primary winding results in a power of 50 W at a voltage of 10 V. Current flows through the left MOSFET for half the time and through the right MOSFET for the other half. Altogether it has the same losses as if the same current flowed continuously through just one MOSFET and the primary winding. The MOSFET on-resistance is 0.077 Ω, giving a loss of around 2 watts shared between the MOSFETs or 1 watt per MOSFET. This is not excessive and will be easily dissipated using a small heatsink for the TO220 package. According to the manufacturers data a single winding on the ETD coil former has an average length L_w of 52.8 mm, the resistance of a single primary winding amounts to 0.014 Ω, resulting in a loss of around 0.4 W.

An output of 50 W at 120 V results in an output current of approximately 0.4 A, indicating a 0.1 % loss in the secondary winding. Power dissipation in the diodes works out at 0.7 V × 0.4 A = 0.28 W divided between the two diodes, which is well within their specified rating.

Construction

The complete schematic for the DC Booster is shown in **Figure 6.** The switching waveforms are generated by a TL494 (IC2). This device ensures that there is no timing overlap of the waveforms. The gate driver IC3 produces the necessary drive current to cleanly switch the MOSFETs, ensuring low power losses. The MOSFETs drive the transformer primary and switch S1 selects the transformer's turns-ratio. With the switch in one position the booster acts as a six-times multiplier to produce ±90 V from the 30 V input and in the other position it works as a twelve-times multiplier, giving ±180 V.

The gate signal waveforms driving the MOSFETs are shown in **Figure 7.** Close inspection shows that there is no overlap of the two 12 V signals. In place of the Logic-Level MOSFETS you can substitute MOSFETs with a 4 to 5 V gate-threshold voltage.

Component layout for the circuit is not critical. It's important to observe correct polarity of the two primary windings (marked by the spots on the schematic). Small heat sinks are only required for the power devices. **Figure 1** shows the author's original proto-

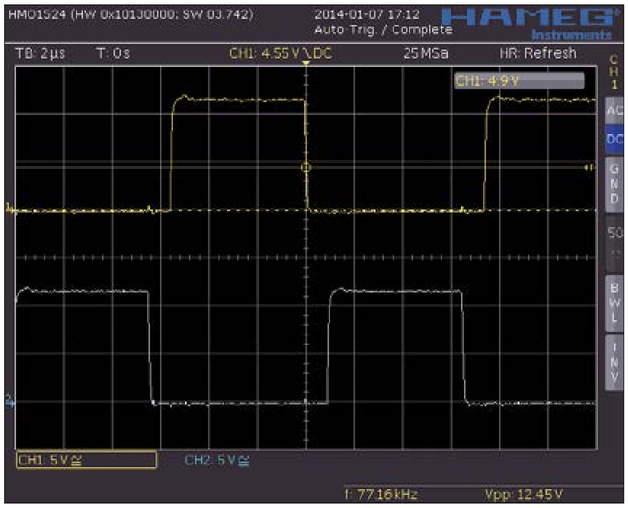


Figure 7. The gate control.

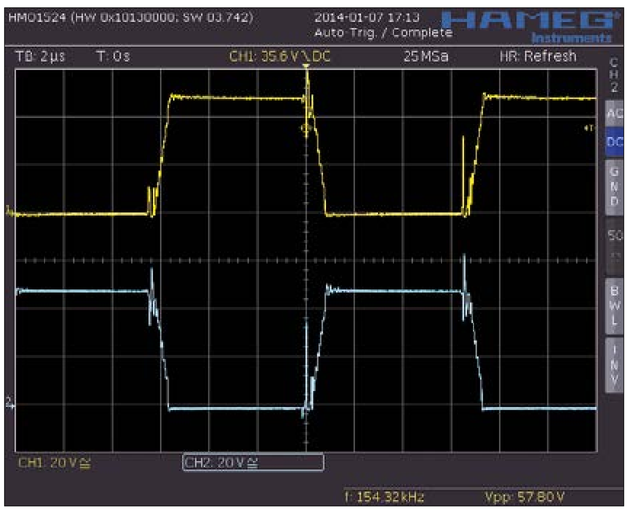


Figure 8. Drain waveform at the primary.

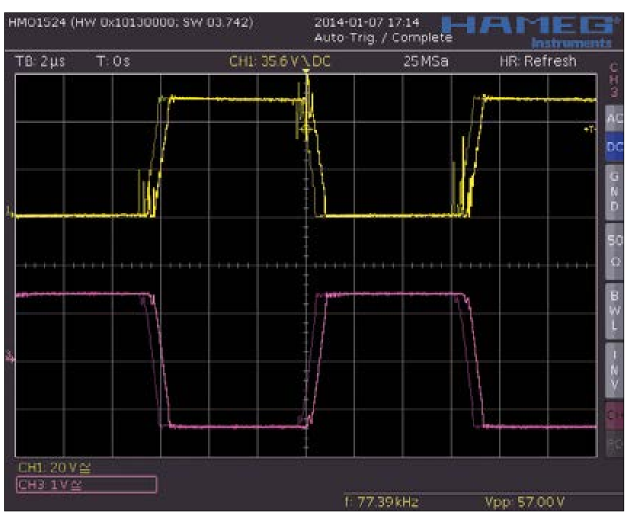


Figure 9. Voltage waveform at the secondary.

type. **Figure 8** gives the MOSFET drain waveforms with an input voltage U_{in} of approximately 25 V. You can see that the waveform has a peak value of $2 \times U_{in}$ with no overshoot.

The upper trace in **Figure 9** shows the drain waveform (U_{in} = 25 V) and beneath it is the voltage at the secondary winding (measured with a 100:1 probe). This shows the ±150 V output voltage, produced from six times 25 V.

Transformer winding

A ETD29 core without air gap is used for the transformer. Any of the power ferrite materials (3F3, 3F4, N27, N30 etc.) are suitable. It is important to ensure that both primary windings are closely coupled. For this reason they are wound on the coil former together (i.e. bifilar), they will take up exactly one layer. The primary wire is made from four lengths of 0.4 mm enameled copper wire wound together. This is best achieved with the help of a cordless drill. Take four lengths of wire of a suitable length and twist them together using the drill. You now need to wind two lengths of these twisted cables around the coil former to make the two sets of eight turns which make up the primary windings. The resulting winding can be seen in **Figure 10**. If these two primary windings are not closely coupled it produces unsymmetrical operation which leads to core saturation and poor efficiency.

The secondary winding consists of 2 x 24 turns, this time the wire is made from 3 strands of 0.4 mm enameled copper wire (#26 AWG) twisted together. Use a layer of insulating tape between the primary and secondary windings and also over the finished transformer. **Figure 11** shows how the finished transformer should look.

Operation

Finally we take a look at how the

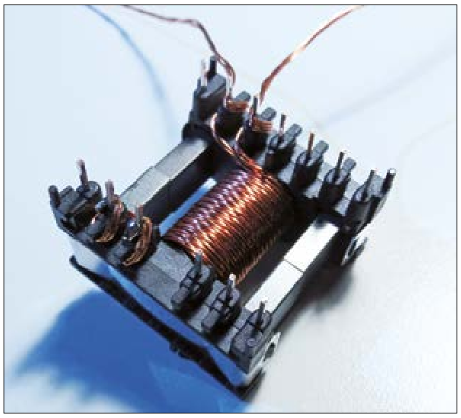


Figure 10. The transformer primary winding.

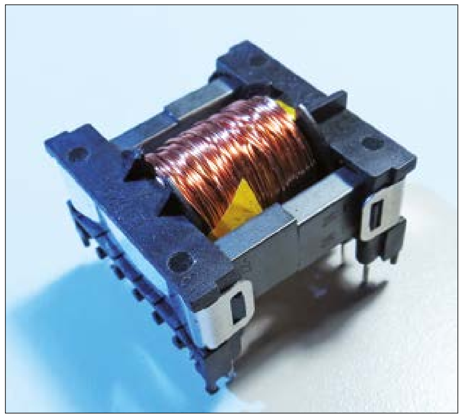


Figure 11. The finished transformer.

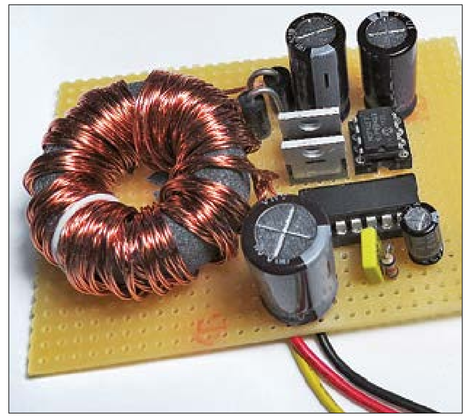


Figure 12. A compact unit for vacuum tube experimentation.

booster is used. For simplicity an external power adapter is used to power the control part of the booster circuit. A standard bench top power supply provides the input voltage to be multiplied. Before powering the booster circuit for the first time, turn down the bench supply output to zero and connect the mains adapter to the control supply input. Connect the bench supply to the circuit input and watch the bench supply current reading as you turn up the voltage. It should only take a few tens of milliamps, if it's much higher; there is a fault in the circuit, most likely a mix up of the primary windings polarity. Use a DVM to measure the high voltage output. The bench supply current limit control can be used to control the booster's output power.

With no ammeter in the secondary to measure load current, keep an eye on the primary supply current. The bench supply current limit facility ensures that the DC booster is short-circuit proof.

When the DC booster is used to provide a supply for experimentation with vacuum tubes, for example (from 12 V) then a compact module like that shown in **Figure 12** can be built. It may then be useful to add another secondary winding to the core to provide a low voltage tube heater supply.

The circuit can also be adapted to power devices such as CRT type oscilloscope tubes. A higher output voltage can be achieved by increasing the number of turns of the secondary transformer winding. As detailed earlier, ensure that the reverse breakdown voltage of the multiplier diodes at the output can handle the higher voltage. Alternatively you use more than one secondary winding each with a diode and buffer capacitor to achieve the required voltage. In both cases take care to ensure there is sufficient isolation between the primary and secondary windings to prevent any possibility of insulation breakdown and voltage leakage.

(130496)

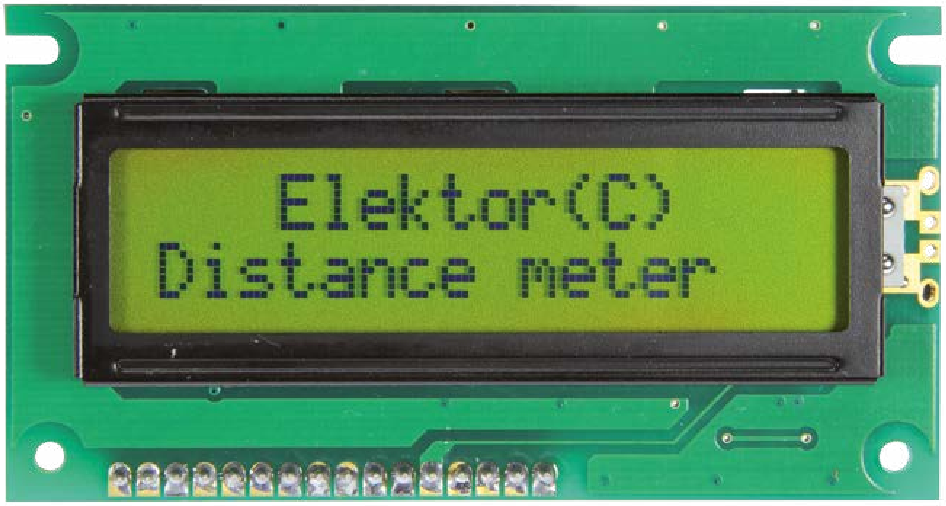
Caution.

This design produces a lethal, high DC voltage. Take every precaution to ensure that no part of your body can accidentally make contact with this voltage, both during construction of the unit and its subsequent use as a high voltage bench supply. High voltage alone is dangerous but this design also supplies a reasonable level of current!

Ultrasonic Tape Measure

Accurately measure up to 13 ft / 4 m

By **Rolf Gerstendorf** (Germany)



OK, so a conventional tape measure would do the job as well but in the 21st century we expect something a bit more sophisticated. Need to measure your room? With this device you can measure all the distances without leaving the safety of your armchair. It shouldn't take long to build and use a low-cost intelligent ultrasonic sensor from Malaysia.

The HC-SR04 ultrasonic sensor is produced by the Malaysian company Cytron Technology [1]. They specialize in sensors for robotic applications in the educational sector. The sensor does not just consist of an ultrasonic transmitter and receiver but also contains a little bit of 'intelligence' that makes it easier to interface with a microcontroller board such as an Arduino. That isn't the only reason that the sensor is very popular amongst the robot maker communities, at around four Euros the HC-SR04 is a very reasonable price.

The sensor consists of a 45 × 20 mm PCB with the transmitter and receiver unit (and a 4 MHz crystal) on one side (**Figure 1**) and the signal conditioning circuitry and controller ICs on the other. According to the data sheet the HC-SR04 has a range from 2 cm to 4 m with a resolution of 0.3 cm. Its ±22.5° beam angle gives the sensor good sensitivity.

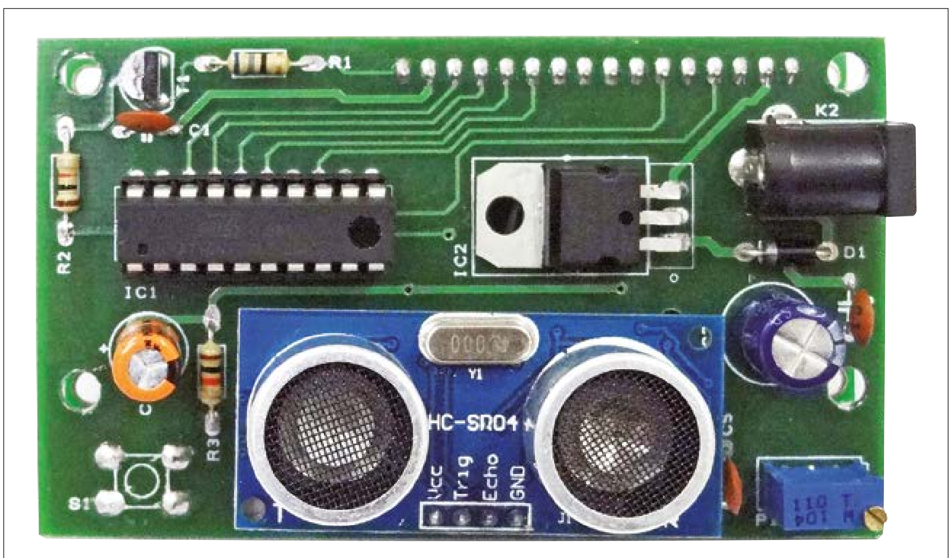


Figure 1. Front face of the ultrasonic sensor unit.

Cytron makes the claim that 'the sensors operation is not affected by sunlight or black non-reflective surfaces' but sound absorbing material such as clothing is another matter!

The HC-SR04 operates from a 5 V supply drawing just 2 mA in quiescent mode and 15 mA in operational mode. Connections for the supply take up two of the four connector pins of the sensor board, the other two are for the trigger input (to the controller pin 14) and the echo output (Pin 6) at TTL level. The sensor timing waveforms can be seen in **Figure 2**. After the 'Initiate' trigger input signal (>10 µs) the sensor produces an 8-period 40 kHz burst and then waits listening for the reflected echo. The sensor firmware then calculates the delay (providing some degree of error correction) and generates a signal at the Echo output, the length of which is equivalent to the distant between the ultrasonic transmitter and the object which produced the reflection. The rule of thumb applied now is that the pulse length divided by 58 gives the distance in centimeters (or by 148 to give the inch equivalent). The signal length will vary between 150 µs and 25 ms to span the complete sensor range. An output pulse of 38 ms indicates that no reflections were detected.

Add more Intelligence

As you can see from the circuit diagram in **Figure 3** apart from the Ultrasonic sensors it consists of not much more than a microcontroller and an LC display connected to port B of the microcontroller. The 'user input device' is just a simple pushbutton that performs multiple functions. A preset pot is also included in the design to allow display contrast adjustment. The circuit also uses a 7805 type voltage regulator together with a diode D1 to protect against incorrect supply polarity.

The Software was originally developed for the ATtiny2313 microcontroller using the BASCOM AVR development environment. This original version only displays the distance in millimeters and has none of the 'bells and whistles' of the latest version. Anyone interested in this beta version can find it at [2] to download. The software has since been developed and the user can now choose to show the distance in millimeters, centimeters, inches, feet or yards with a resolution of two decimal points on the 16×2 LC display. It also allows the user to define a reference point from which the measurements are made. Memory space in the original 2313 design proved insufficient for the fully featured version so the pin compatible ATtiny4313 was chosen instead.

Either software version has the same basic structure and begins after initializing with the controller with the Main Loop **Dist** to generate a trigger pulse each second from pin PB2 of the controller to trigger the sensor and initiate a measurement cycle.

The **Interrupt Routine Isr_lbl** is triggered by a rising edge on the interrupt input (0) pin PD2. It measures the length of the output pulse and passes the value back to the main routine. Here it is converted into the chosen format and sent out to the LC display. The timing is based on the 8 MHz crystal.

There is also a **Setup-Function** to set up the operating mode: In **Calibrate-Mode** the distance to the first reference reflecting surface is stored as the zero value. The **Default-Mode** resets the selected units back to their default value and uses the sensor's front face as the reference surface for all measurements.

The function **Timer 1** takes care of the display back lighting. A press of the button turns on the back light and to save energy turns it off again after eight seconds if the button hasn't been pressed in the meantime.

Put it together

Both the LCD and sensor board are already complete units and need no further work so this leaves a small amount of soldering to mount components on the PCB shown in **Figure 4**. The microcontroller is fitted using an IC socket and the voltage regulator is mounted flat to the board surface. The project was developed with Designspark [2] and the

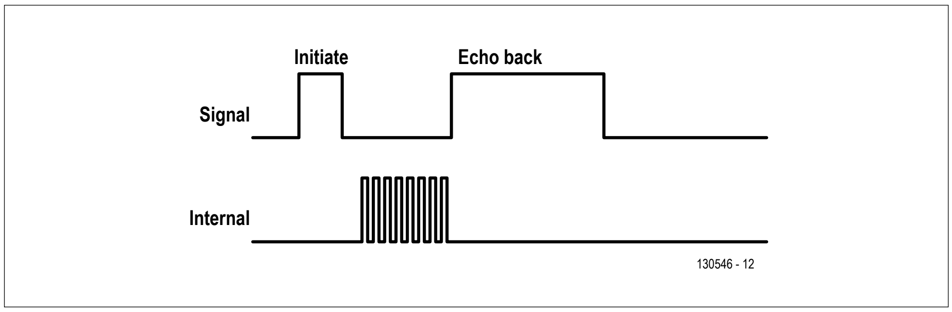


Figure 2. The sensor timing diagram.

It shouldn't take long to build and use a low-cost intelligent ultrasonic sensor from Malaysia.

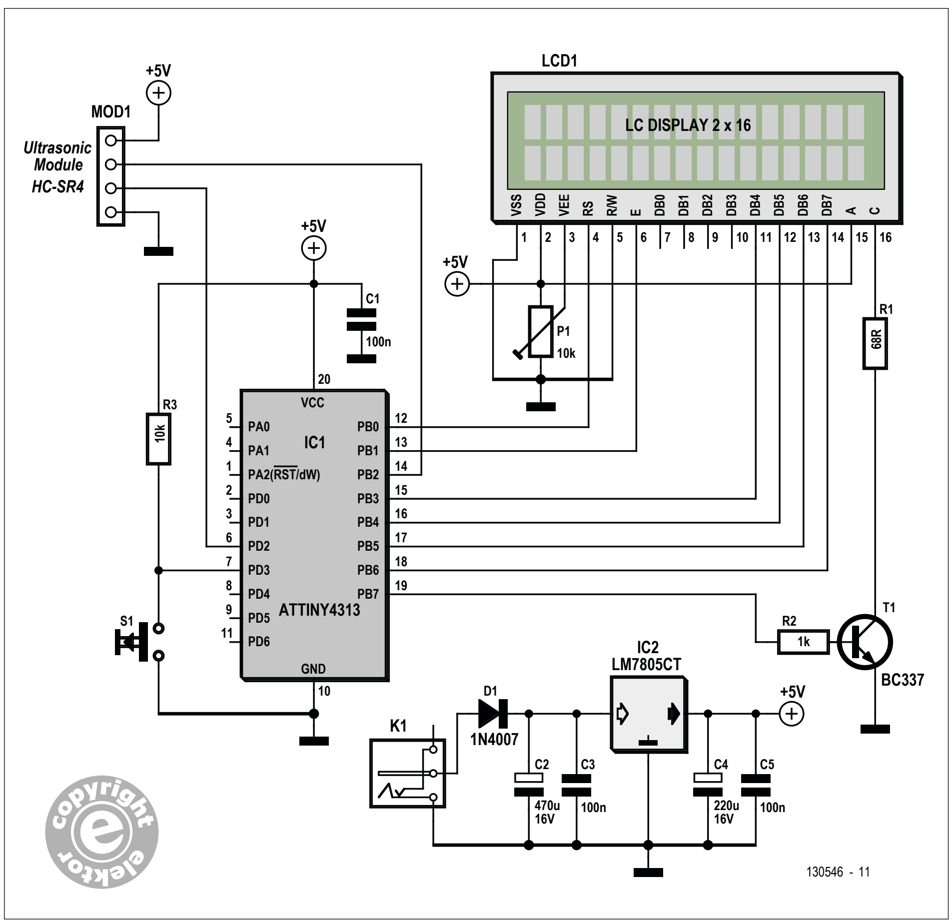


Figure 3. A controller and display, nothing to worry about there.

Component List

Resistors
R1 = 68Ω
R2 = 1kΩ
R3 = 10kΩ
P1 = 10kΩ multiturn trimpot (Bourns 3296W-1-103LF)

Capacitors
C1,C3,C5 = 100nF
C2 = 470µF 16V, 10mm pitch
C4 = 220µF 16V, 7.5mm pitch

Semiconductors
D1 = 1N4007
T1 = BC337-40
IC1 = ATtiny4313-20PU, programmed, Elektor Store # 130546-41 (see text)
IC2 = LM7805

Miscellaneous
LCD1 = LCD Module 2x16 characters, Fordata FDCC1602N-FLYYBW51SE
MOD1 = 4-way SIP connector to match:
HS-SR04 ultrasonic sensor
K1 = socket for roll pin
S1 = pushbutton, Alps SKHHALA010
PCB # 130546-1

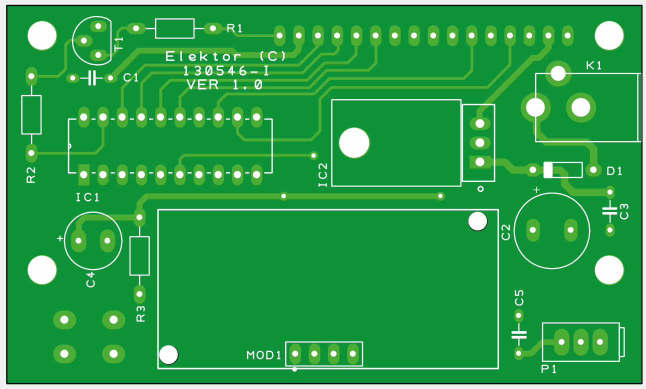


Figure 4. The PCB can be assembled in no time.

Available in the Elektor Web Store

controller software (Full-Version) can be downloaded free of charge from [3]. To make things even easier you can order a ready-programmed ATtiny4313 controller from the Elektor shop where you can also order an (unpopulated) PCB for this project [3].

The controller board is sandwiched between the display board and the sensor board (see main photo). The display board can simply be connected to the controller board using short lengths of wire or alternatively make it pluggable with a length of pin headers/sockets. The sensor board already has a four way SIP pinheader to plug into a corresponding socket mounted on the controller board. The HC-SR04 data sheet emphasizes (in red) that the GND pin must make contact **before** power is applied to the V_{cc} pin. This means that the board must first be plugged in before power is applied. To avoid the risk of accidentally plugging into a powered board, clip off a few millimeters from the V_{cc} pin so that it always makes contact after the GND pin.

At switch-on the controller will supply measurements in millimeters. By pressing the pushbutton the units change to centimeter, inch, foot, yard and then back to millimeters. A longer press puts the controller into setup mode. This allows you to define the measurement reference point: the default point is the front face of the ultrasonic transducers but you can define another reference point for measurements (calibrate). In the second case the sensor measures the distance to a target and a long press of the pushbutton makes the controller set this distance as the reference (in calibrate mode) for further measurements. A short press on the pushbutton toggles between the two modes and a long press selects the corresponding option.

(130546)

Weblinks

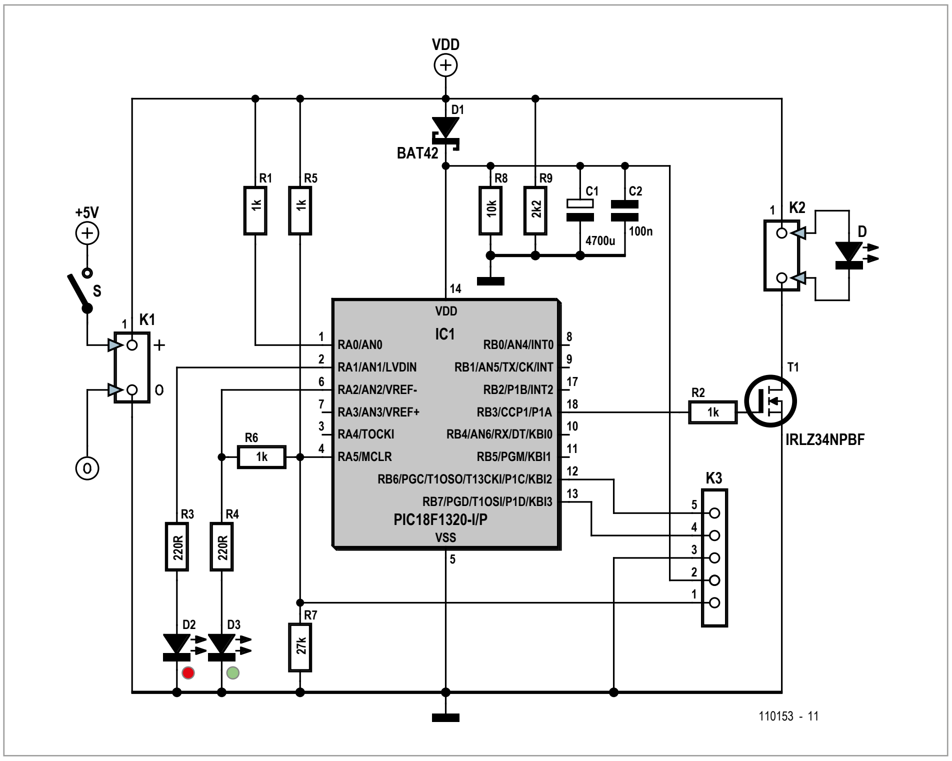
[1] www.cytron.com.my/

[2] www.rs-online.com/designspark/designshare/eng/projects/184/view/stage/design/

[3] www.elektor-magazine.com/130546

Intelligent LED Dimmer

By **Christian Wachsmann** (Germany)



LEDs are undoubtedly becoming more prevalent in domestic lighting applications. Low voltage operation and small outlines mean they are used in ever more imaginative places (check out the LED tile cross lamps which fit in grouting channels). The chances are that any LED lighting installation will benefit from a dimmer controller. Standard phase-fired lighting dimmers are not suitable for mains powered LED units. To reduce power losses when dimming LEDs it is more efficient to use a PWM waveform.

Nothing could be easier I hear you say, all you need is an NE555, maybe with a power transistor at the output. What may be the simplest technical solution is however complicated in other respects i.e. by installation. Such a circuit would also need cabling to a variable resistor to control the dimmer. Long wires are prone to interference pick-up which can affect stability of the switching threshold. In the age of digital electronics the author thought that it was about time to consign the analogue pot to the junk box of history.
So he set out to replace the variable-resistor function with a small low cost microcontroller. To generate a PWM output is not a problem for a microcontroller and if the dimmer setting can be digitised then there will be no need for any analogue values and not a pot in sight. With a little thought it was possible to devise a method by just using the simple on/off switch…
How it works: When the dimmer is switched on it remembers the last light level setting and sets the LED brightness accordingly. To change the brightness setting it is necessary to turn the dimmer off and back on within a time period of two seconds. The dimmer now begins to slowly cycle through the brightness levels from maximum to minimum. Turn the dimmer off and on quickly to store the current brightness value. Turning off without turning back on changes nothing and the dimmer value reverts back to the previous stored value when it is next switched on.
The operating and change setting procedure is both simple and logical. There is certainly no need for an operator's manual and the behaviour can be very easily implemented in a microcontroller's software. The circuit diagram shows that there is not too much circuitry external to the PIC18F1320 microcontroller. The 5 V power adapter power supply is not shown but plugs in to connector K1. Schottky diode D1 is in series with the power line to the microcontroller and also provides a path to charge the reservoir capacitor C1. This stores enough energy to keep the microcontroller running for 2 s after power is turned off by switch S. R8 provides a high impedance path for C1 to fully discharge when the power is off.
Resistor R1 connected from the supply to input pin 1 indicates when the power supply has been turned off or on. Red LED D2 indicates that the unit is on and green LED D3 lights up to indicate the two second window within which the processor can be switched on again to store the new dimmer setting. Resistor R5 pulls the reset input (IC1 pin 4) high. During the two second programming window when the green LED is on resistor R6 ensures that the reset remains high even though the supply voltage through R5 has fallen. Only after this period can the reset input be pulled low.
MOSFET T1 is driven by a PWM signal at its gate terminal via resistor R2. If you substitute an alternative MOSFET ensure that it can be driven fully into conduction with a 5 V gate voltage. With load currents up to 2 A T1 will not require any heat sink. It is important to note that the on/off switch is placed in the low voltage output from the AC power adapter. If the switch were used to turn off the adapter mains input instead, its 5 V output would turn off far too slowly for this circuit to be effective. The pin out of K3 is compatible with the standard Microchip PICkit 2 programmer connector.

The Elektor web page for this project [1] contains both the software HEX file and the C code source file which allows you to study the program more closely and make changes. A ready-programmed microcontroller is also available giving you the possibility to build the circuit even if you do not own a programmer.

(110153)

Internet Link

[1] www.elektor-magazine.com/110153/

ARM CMSIS Design Contest

$10,000 for winners plus 400 free developer boards

ARM and microcontroller manufacturers ST, NXP, Freescale and Infineon, together with Elektor, announce a fantastic competition for developers. Turn your interface application into reality with the CMSIS driver library from ARM and select one of four well-equipped development boards! ARM/Keil will support contestants for 6 months with a free license to use their powerful MDK ARM development environment.

The ARM controllers of the Cortex-M series are powerful and versatile, yet they cost no more than 8-bit controllers. Semiconductor firms such as NXP, Freescale, Infineon and ST have licensed the cores from the British processor foundry, enabling them to offer controllers backed up by well-equipped development boards. As you would expect, the software support includes powerful development environments with integrated compilers, libraries and sample programs.

Among the crucial ingredients for success are low-level drivers, which greatly simplify access to interfaces such as U(S)ART, I2C and USB. There's no longer any need to spend time studying data sheets in order to find the relevant Register; instead developers can simply make use of well-documented Functions in the C programming language. In this respect, an inspired idea from ARM is the CMSIS driver standard, which unifies the driver features across manufacturers. Consequently the learning curve for developers who work with differing ARM controllers is reduced significantly. At the same time firmware can be ported far more easily from one controller to another. There's plenty more information at www.keil.com/cmsis-driver.

Total of $10,000 to be won

This competition specially for developers is powered by ARM and the microcontroller manufacturers ST, NXP, Freescale and Infineon, in association with Elektor. The CMSIS driver library will simplify your task, whichever of the four differently featured boards you choose from the four separate suppliers! ARM/Keil will support contestants with a free license for the MDK professional ARM development environment, good for six months. Your project must reach us by **June 30, 2015** and a condition is that the software must be published under the very straightforward BSD 3-clause Open Source License.

An expert jury will select the best projects from all the entries. Developers will score points not only for ingenious, useful and preferably innovative project ideas, but also for the excellence of their coding. Software must take advantage of the CMSIS driver library, also be simple to maintain and easily portable. Contestants should make use of as many features as possible of the respective boards, minimizing as far as possible the need for additional — costly — electronics.

Up for grabs are:

1st Prize. US$5,000 in cash
2nd Prize. US$3,000 in cash
3rd Prize. US$1,000 in cash
4th Prize. US$500 in cash
5th Prize. US$500 in cash

Sign up immediately with your idea — we have 400 well-equipped developer boards waiting for contestants!

Registration, contest rules and further info:

http://armcontest.elektor.com

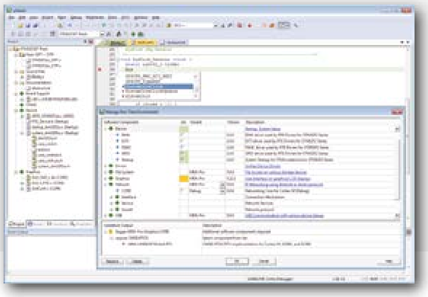
Sample project and video tutorial for the MDK-ARM:

www.keil.com/contest

(140384)

MDK-ARM by ARM/Keil

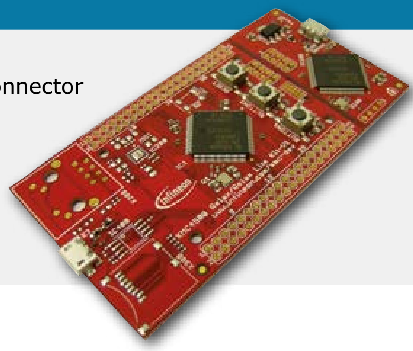
- Support for Cortex-M, Cortex-R4, ARM7 and ARM9 controllers
- Industrial grade ARM-C/C++ compiler
- µVision4 IDE for code development, debugging and simulation
- Keil RTX RTOS (with source code!)
- TCP/IP Network stack
- USB device and USB host stacks
- GUI library
- Comprehensive collection of sample projects



www.keil.com/mdk5

XMC4500 Relax Kit from Infineon

- XMC4500 Controller (Cortex M4, 1 MB Flash, 160 kB RAM)
- USB port
- 2 user-assignable LEDs
- 2 user-assignable push buttons
- 32 Mbit Quad SPI Flash
- Crystal for RTC
- Ethernet interface
- Micro SD card slot
- Extension header connector with CAN, DAC etc.



www.infineon.com/relaxkit

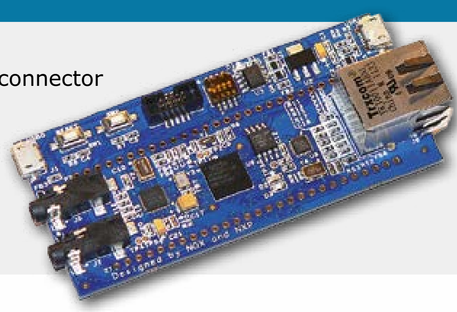
Freescale Freedom Development Platform

- MK64FN1M0VLL12 MCU (Cortex M4, 1 MB Flash, 256 kB RAM)
- USB OTG port
- RGB LED
- 2 user-assignable push buttons
- FXOS8700CQ Accelerometer/Magnetometer
- Ethernet interface
- SDHC slot
- Arduino-compatible I/O connectors

www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-K64F

LPC4330-Xplorer-Board by NXP

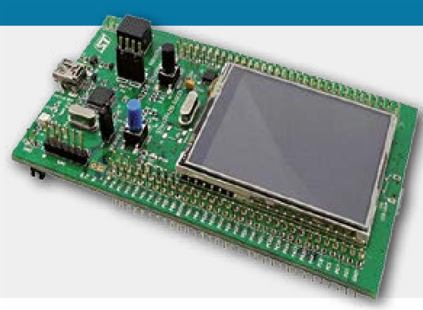
- LPC4330 (Cortex M0/M4, external Flash, 264 kB RAM)
- 2 USB ports
- 2 user-assignable LEDs
- 1 user-assignable push button
- Onboard audio codec and audio jacks
- 32 Mbit quad SPI Flash
- Ethernet interface
- Micro SD card slot
- Extension header connector



www.nxp.com/demoboard/OM13027.html

Discovery kit for STM32 from ST

- STM32F429ZIT6 (Cortex M4, 2 MB Flash, 256 kB RAM)
- USB-OTG-Port
- 2 user-assignable LEDs
- 1 user-assignable push button
- 2.4” touch-screen QVGA TFT LCD
- 64 Mbit SDRAM
- L3GD20, motion detector and 3-axis gyroscope
- Extension header connector



www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090

What People Want

By **Gerard Fonte** (USA)

To have a successful product you have to give people what they want. This is so elemental and so obvious it seems silly to have to say it out loud. And yet people regularly seem to think that this rule can be ignored. Or perhaps, they're just not thinking. I often refer to human nature and this is a fundamental aspect of human nature. Fighting human nature is a battle lost before it starts.

Failure is the Only Option

It doesn't matter how brilliant the concept is. It doesn't matter if it's a totally new technology. It doesn't matter that it's inexpensive. It doesn't matter who invented it or who promotes it. If people don't want it, they will not buy it. Period. End of story. And yet this tune is re-played over and over and often by people who should know better. Hopefully you will not join this sing-a-long.

Steve Jobs left Apple and developed the NeXT computer. It was, by far, the most powerful personal computer of the time but cost over three times more than the competition. People didn't want to spend $6500 for any computer (which was half the price of a new Ford Mustang).

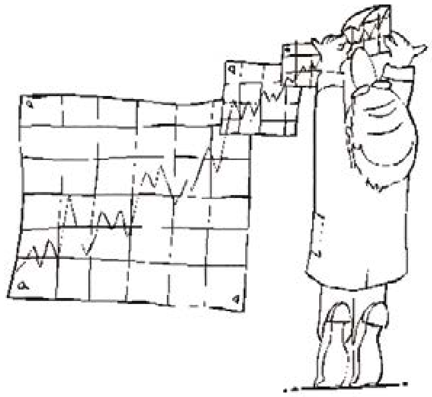
The Segway (the two-wheeled, stand-up scooter) was billed as revolutionary mode of transportation. Perhaps it was. But most communities ban motorized vehicles from sidewalks (except wheelchairs) and using it in the street was scary. It cost $5000, weighed one hundred pounds and had a top speed of 12 mph (bicycles weigh less than 30 pounds, cruise at 12 mph and can reach 25 mph). People don't want portable things that can't be stored easily or carried up and down stairs, or something that makes them look lazy. Especially expensive things.

In 1985 Coca Cola changed the 100 year-old formula for their drink and introduced the New Coke in place of the original. This was based on taste tests that showed that people preferred a sweeter drink (like Pepsi). There was a huge backlash. People don't want their favorite things changed. As Coca Cola president Keough said of the debacle: "Our research could not measure or reveal the deep and abiding emotional attachment to original Coca-Cola felt by so many people."

The main problem here is that the developers only focused on the "new and better" aspects of their product. They weren't able to step away from it to get a better perspective. This is an easy thing to do. We generally think other people are just like us. That's another aspect of human nature. That is, these examples show that different facets of human nature can conflict with each other.

Success is Not an Option

Just as you can't make people want something, you can't make people stop wanting something. Generally this is more of a social problem rather than a marketing problem. Although if you are introducing a product that closely competes with an existing product that is well-known and well-liked, you may face this situation. (Do you think Fonte-Cola has a chance?)



Alcohol, cigarettes and recreational drugs are the three big items in this category. Religious and political leaders have tried on many occasions to eliminate alcohol. In 1920, the eighteenth amendment to the US Constitution was enacted which banned alcohol. It gave rise to "speakeasy clubs" and organized crime. It lasted until 1933 when it was repealed by the twenty-first amendment.

The "War on Drugs" was initiated in the US by President Nixon in 1971. And despite 1.5 million drug-related arrests and 500,000 people sent to prison PER YEAR, there hasn't been much reduction in drug use in the US. (today WoD is a fine rock band , *Ed*.)

There has been considerable success in reducing the number of smokers in the US. But it has not been easy. There are massive public awareness campaigns, a ban on tobacco advertising and huge taxes on tobacco products. Then there's that unfortunate fact that a third of tobacco users will die when using the product as directed. Even with all that, there are still 16,000,000 Americans who smoke. And now there are e-cigarettes.

Up the Flagpole

Many times it's difficult to know what people want. Steve Jobs and Steve Wozniak were pretty sure their Apple II would sell to the growing number of computer hobbyists. I don't think they expected the explosion of interest from the general public. Their affordable solution of integrated hardware and software was exactly what people wanted. It was the original plug-and-play computer. Anybody could use it.

Google-Glass didn't fly. They stopped production on Jan 15, 2015. There were lots of security and privacy concerns. But those issues generally don't limit sales too much. Perhaps they were too advanced or didn't work as well as advertised. Of course, it doesn't really matter what the reasons were. They didn't sell because people didn't want them.

So, what do people want? I suggest food that contains no calories. Make a taco-flavored chip from non-digestible plant fiber. (There are many artificial flavors to choose from.) People need more fiber in their diets. It promotes health and helps to reduce weight. There's a huge number of people who want to lose a few pounds. And, add in a few vitamins as well. You can have a healthy lunch of a no-calorie drink and no-calorie food. You feel full and it tastes good. In ten years, this can be a billion dollar industry. You heard it here first!

(150050)

LEARN DESIGN SHARE

By Jaime Glez Arintero jaime.glez.arintero@eimworld.com

Welcome, Circuit Wanderer

We try to make every issue of Elektor (e-)special. But as you may have noticed, this one is more special than ever as a lot of things have changed in the structure of your magazine.
At some point we realized that there are three basic *operations* in the life of every electronics enthusiast that he or she keeps doing all the time: learn, design, share. You got it, that's our brand new motto. And although due to physical boundaries we're always forced to arrange our contents in one way or another, these actions may or may not follow any specific order. Moreover, they just happen. You may learn something new, and feel the need to design. Some things are too good to be kept in the dark, so you share them. Alternatively you can learn while you design, share while you learn, or even design while you share. The order of the activities does not dictate the outcome — let your creativity decide for you. Exchanging ideas has always been the base of human progress. In electronics, this exchange usually brings projects to life in record time. It makes you see things from different perspectives, simplifies the tasks, and is a strong binding factor in communities. Sharing is not only caring, but a vastly efficient and possibly the smartest way of working. Now, more than ever, feel free to **share** your projects, ideas, designs, or tips. Here in the SHARE section we're all ~~transducers~~ ears.

"Wanderer, there is no road, the road is made by walking." - Antonio Machado

Watts up?

I C new sections here…

Err-lectronics

To err is human, they say.
Programmers call them "patches," but since we have no means of updating your printed copy of Elektor Magazine, we have to resort to publish a regular section with corrections and updates. *Err-lectronics* is one of the brand new sections, and your input is more than welcome. Engineers love updates and tweaks as much as the original product. Did you spot an error, or anything subject to improvement? Go ahead, it's never too late, at least in electronics! (they also say that, right?)

Escaped from the Labs

And lived to tell!
Trust me, the space called Elektor Labs is not a dangerous place, as long as Jan Visser refrains from baking pizzas with an SMT oven, or lighting up LEDs in an improvised sausage circuit (*true story, bro*). If that's the case, our High Risk Specialist/Reporter Thijs Beckers will tell us what's going on there. With every issue you may expect an amusing or triggering electronic engineering story, as always without forgetting the human touch. Hopefully you'll recognize yourself at times!

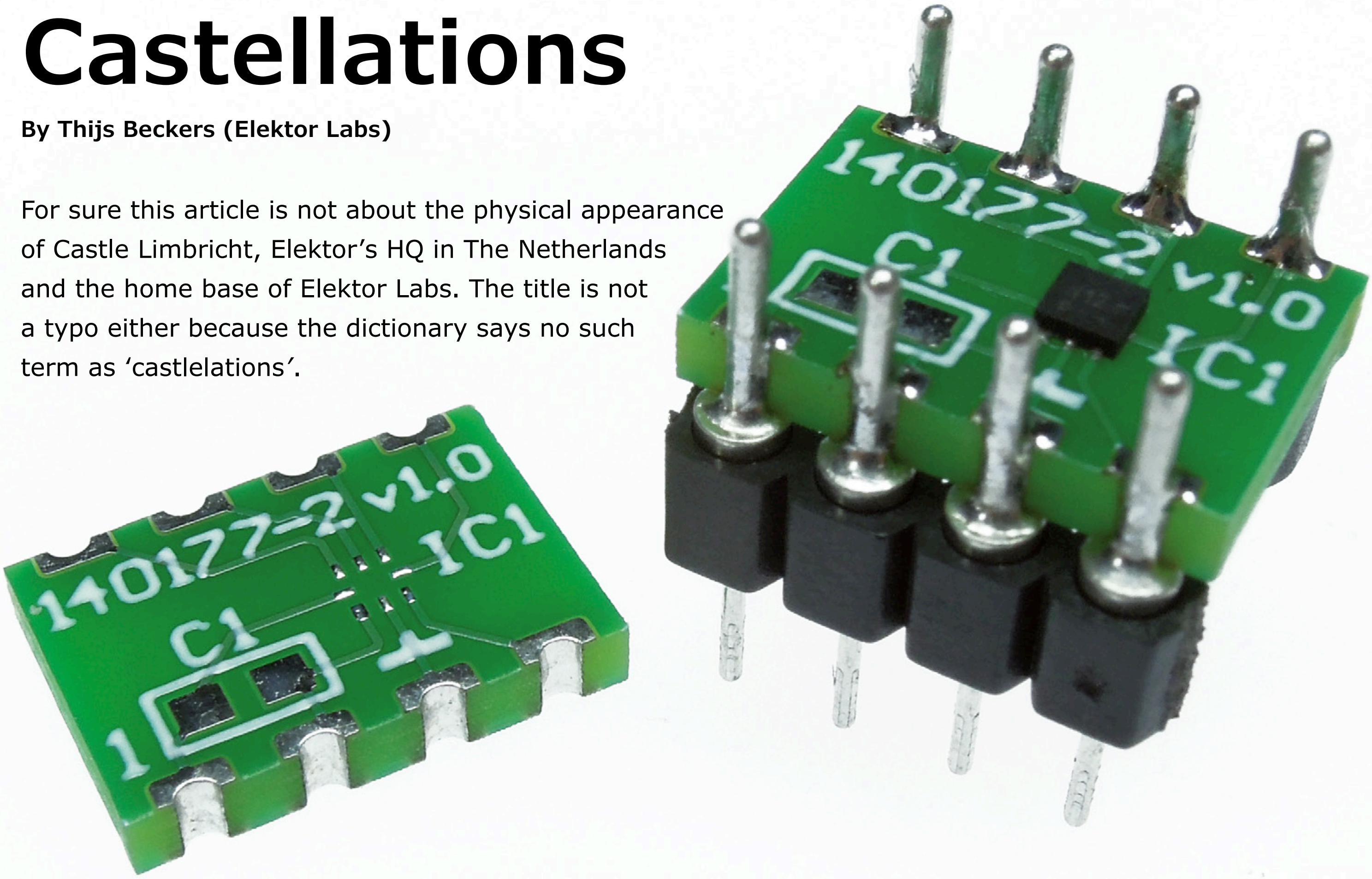
Webscouting

But isn't that what we all do daily?
As the poet T.S. Eliot pointed out, *"where is the knowledge we have lost in information?"* With the Internet we're exposed to massive amounts of information daily, so like in all fields of interest, the good stuff keeps hiding somewhere. However, our Dutch editor Harry Baggen will provide a compilation of the best of the *interwebz*, putting the spotlight on selected sites he believes every electronics enthusiast should visit as a minimum. Did you find any links that are worth sharing? Let us know!

Castellations

By Thijs Beckers (Elektor Labs)

For sure this article is not about the physical appearance of Castle Limbricht, Elektor's HQ in The Netherlands and the home base of Elektor Labs. The title is not a typo either because the dictionary says no such term as 'castlelations'.



Castellation in our neck of the woods is a term for vias or through ('thru') holes in PCBs that are cut right through the middle to create semi holes or even partial holes. In practical terms: the edge of the PCB is 'castellated' with copper connections. Another term frequently used is *PTH on edge*, where 'PTH' stands for Plated Through Hole. *Plated Half-holes* is also being used.

Castellations are used for various purposes. The most popular application is probably the linking of two boards side by side. This type of linking is often used to mount pre-assembled modules on a larger motherboard, while simultaneously providing a convenient inspection point to check the electrical connection and tap into signals there. Joining the PCBs together directly also makes the whole system considerably thinner than a comparable connection with multi-pin connectors.

The castellation technique can also be used where case pins need to be connected to the side of a PCB. We've done this in the past, for example with our cracker *FT232R USB/Serial Bridge/BOB* [1], to the amazement of some who complained about badly cut boards. Now this technique is applied once again with an upcoming project from Elektor Labs. The main reasons to go for castellation for this particular project are:

- Space. This solution allows for the smallest possible footprint of the board holding the NXP NTAG-IC. It also allows the board to be soldered directly onto the main board conveniently.
- Function. For optimal performance the antenna must be as far away as possible from conducting surfaces, e.g. copper planes and traces.

Fabrication of castellations requires extra attention from PCB manufacturers, as extra steps need to be taken during production of the PCB. An extra pass is needed in the drilling department after the *direct metallization* and *dry film imaging* of the outer layer processes. To ensure correct manufacturing, the data preparation department needs to pay extra attention

About the author:

Thijs joined Elektor back in 2005 as a freshman in the Dutch editorial department, immediately following graduating for his Bachelor degree in Electrical Engineering. After switching to serve the English editorial team for a short time while part-timing for Elektor Labs, he has now been fully appointed to the Elektor Labs team for over a year. He is responsible for setting up the production and ensuring the accurate manufacturing of Elektor Kits and Modules. His personal interests are electronics design (with a focus on audio-related electronics), loudspeaker design, repairing electronics and (e-)drumming, where e-drumming can be the ultimate combination of all three other interests.

to the setup of the files for the production machines. And of course the designers will have to indicate their wishes before the manufacturing process starts.

Here at Elektor Labs we work closely together with Eurocircuits for the production of our prototype boards. To make sure this 8-pin DIP sized proto-PCB (see **photo**) got manufactured correctly, we adhered to the following guidelines when uploading the data:

- Indicate the board outline in the copper layers as well as in a separate layer (in Altium Designer we use Mechanical 1 for the board outline, see **screen shot**). It needs to be in the copper layers for correct positioning. The separate layer then indicates special attention needs to be paid during processing. Yes, it looks like a giant short circuit, but the production preppers at Eurocircuits recognize this as the instructions for castellation and remove the short from the copper layers prior to production.
- When ordering the PCB, make sure to tick the *PTH on the board edge*-option (see screen shot).
- Include a separate text file with the note that the PCB is supposed to be castellated.

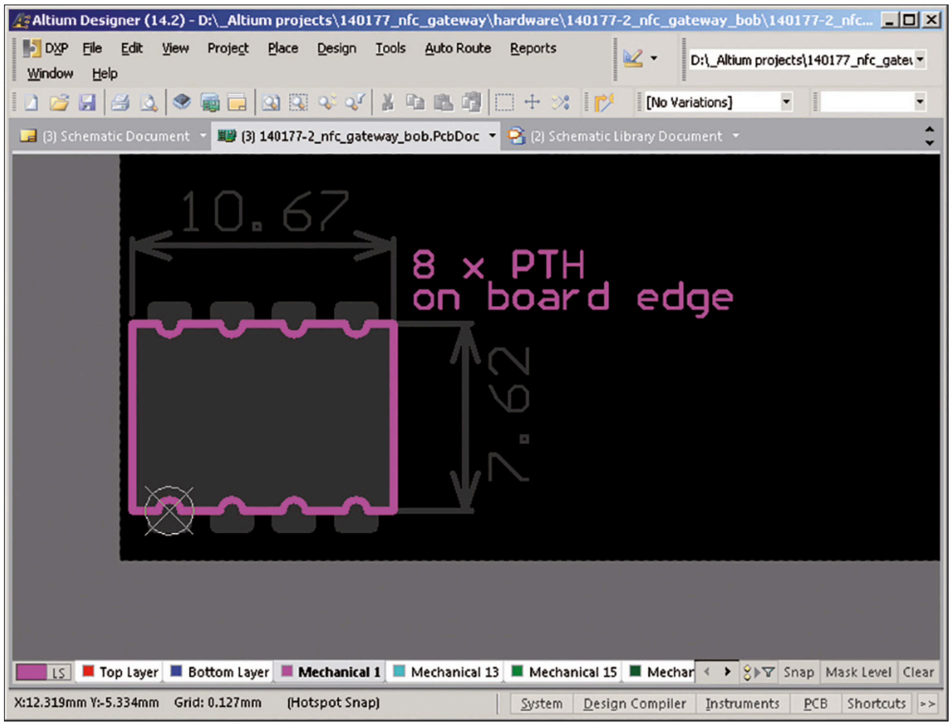
On top of that, the following recommendations are in order:

- Use the largest hole size possible, 0.80 mm is the recommended minimum.
- Use the largest outer layer pad possible, both top and bottom sides
- Locate pads on the top as well as on the bottom layer to securely anchor the plating to the PCB.
- If possible, place inner layer pads to anchor the hole barrel. This will also help reduce burring during the castellation process.
- Ensure there is enough spare space on the edge of the PCB to hold the PCB in the production panel during manufacturing. If you need castellated holes on all four sides, special attention needs to paid to the design and it is advisable to contact your PCB manufacturer and discuss your proposed profile as early in the design process as possible.
- All surface finishes are possible, but for smaller sizes selective gold-over-nickel is generally preferred.

If you stick to these guidelines, you should be able to produce perfect PCB production files (Gerber files) for castellated boards. Happy designing!

Just in case you wonder: The IC in the photo is only 1.6 by 1.6 mm by 0.5 mm high! And yes, we soldered it ourselves, using our EC-Reflow-Mate [2].

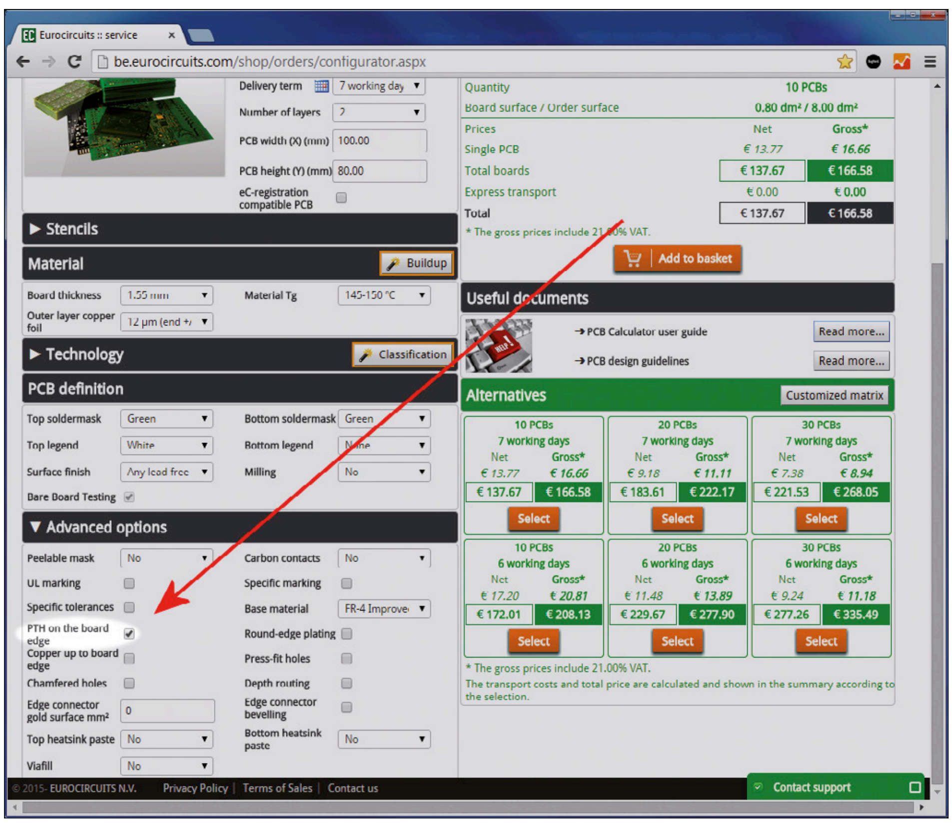
(140527)



Web Links

[1] www.elektor.com/110553-91

[2] www.elektor.com/ec-reflow-mate



HP 400H VTVM Restoration
Part (1)

By Chuck Hansen (USA)

A few years after the introduction of Hewlett-Packard's first commercial product, the 200A audio oscillator [1], there was an obvious need for a means to accurately measure their 200-series oscillators as well as any signals from the output of audio equipment that was being tested with those oscillators.

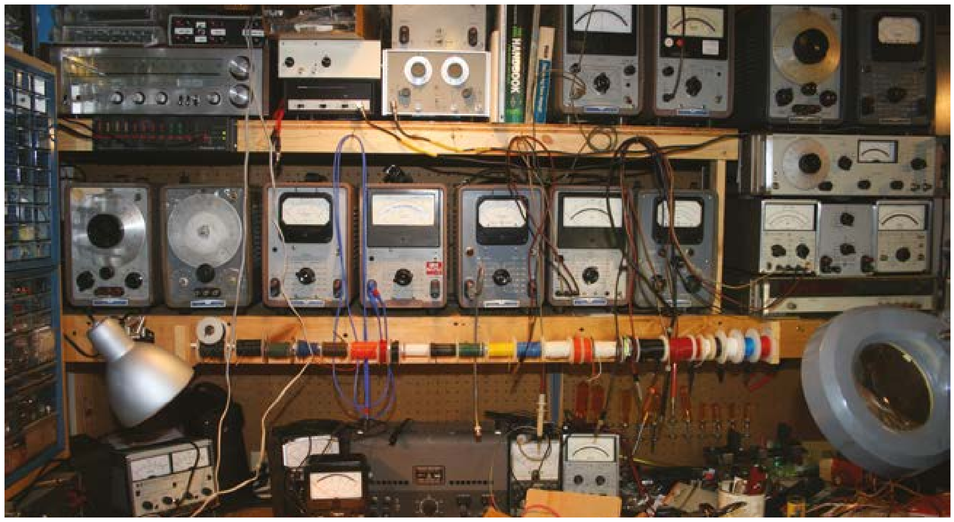


Image courtesy Steve Byan. HP 400H is instrument with red sticker.

The original passive voltmeters used a moving-coil galvanometer, which employs a small high resistance coil of fine wire suspended in a strong magnetic field. When an electric current is applied, the galvanometer's indicator rotates and compresses a small flat-coiled spring. The angular rotation is proportional to the *current* through the coil.

One of the design objectives of any measurement instrument is to disturb the circuit as little as possible. A voltmeter should draw a minimum of current to operate. A voltmeter application requires a series resistor so that the angular rotation becomes proportional to the applied voltage. This is achieved by using a sensitive galvanometer in series with a high resistance.

The sensitivity of such a meter can be expressed as "ohms per volt", or the resistance in series with the meter divided by the full scale measured voltage (V_{fs}). For example, a 1-volt meter with a sensitivity of 1000 ohms per volt would draw 1 mA at full scale deflection (Ohm's Law).

Moving-coil instruments with a magnet field respond only to direct current. Measuring alternating voltage requires a rectifier in the circuit so that the coil deflects in only one direction. Another method is to use an iron-vane meter movement, where a coil wrapped around an iron cylinder induces a magnetic field into the cylinder. Another concentric iron cylinder section within the outer cylinder rotates by mutual magnetic repulsion. A pointer is attached to the inner cylinder and indicates the current on a logarithmic meter scale, whose non-linearity compresses the readings at the low end.

Electrostatic voltmeters use the mutual repulsion between two charged plates to deflect a pointer attached to a spring. Meters of this type draw negligible current and work with either alternating or direct current.

Vacuum Tube Voltmeter (VTVM)

The sensitivity and input resistance of a voltmeter can be increased if the current required to deflect the meter pointer is supplied by an amplifier and power supply instead of by the circuit under test. An electronic amplifier between the input signal and the meter gives two benefits; a more rugged moving coil instrument can be used, since its sensitivity need not be as high, and the input resistance can be made higher, reducing the current drawn from the circuit under test. Amplified voltmeters have a fixed input resistance of 1 to 20 MΩ, independent of the selected voltage range.

The vacuum–tube voltmeter concept was developed during WW I in the UK by R. A. Heising to minimize a passive voltmeter's adverse influence upon the circuit being measured. The first practical VTVM was invented by E.B. Moullin of the University of Cambridge in 1922, and was put on the market by the Cambridge Scientific Instrument Company.

Harold Black devised a negative feedback circuit in 1927 through the study of the carrier telephone amplifier designed at Bell Laboratories. His circuit improved the stability of the amplifier and its nonlinear nature, paving the way towards more accurate measurements.

Alan Blumlein in the UK invented the vacuum tube long-tailed differential pair circuit in 1936, and a more stable DC amplifier was realized. All the circuit bits and pieces needed for a practical vacuum-tube voltmeter were finally available. Moreover, the meter became easier to read because the negative feedback allowed for a linear meter scale.

VTVM advantages

The VTVM has a much higher sensitivity than electromechanical meters. Every

ESTD 2004 www.elektor.tv

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

voltmeter has an input resistance that forms a voltage divider in conjunction with the impedance of the circuit being measured. This is of no consequence when measuring power line voltage with its fractions of an ohm source impedance. But when measuring signal voltages in low-level vacuum tube circuits with hundreds of kilo-ohms (k-ohms; kohms; kΩ) of plate resistance, low sensitivity becomes a major cause of reduced accuracy. I've seen vacuum tube service schematics that specify the make and model of passive VOM that was used to make voltage measurements, so the results can be duplicated.

Another advantage of the VTVM is that AC signal measurements can be made in the presence of the high DC voltages in vacuum tube amplifiers. The high-voltage blocking capacitor prior to the input attenuator circuit allows accurate measurement of very low AC input signals over a wide frequency range.

The VTVM, with its vacuum tubes, is more tolerant of accidental overloads. The specification for the HP 400A states that "Occasional overloads of 100 times normal will not damage the meter movement". Solid-state voltmeters can be less forgiving of voltages in excess of their maximum input ratings.

Hewlett-Packard VTVM history

At the time of their introduction, HP VTVMs offered unprecedented reliability for their price. The meter scale was calibrated in both RMS (root-mean-squared) AC volts, and decibels referred to 1 mW into 600 Ω (today called dBm). The first HP VTVM, the 400A was designed by Dave Packard in 1941 (**Figure 1**). It was introduced early in 1942 and manufactured until 1958. Notable for its stability (it was one of the earliest vacuum tube voltmeters to need no initial adjustment for zero and none for drift), its high input impedance, and over 1 MHz bandwidth, and the 400A became an industry classic. A key to achieving its performance was application of large amounts of negative feedback.

The 400 series also has a pair of amplifier output banana jacks that provide 0.15 VAC at full scale meter deflection on all ranges, with a 50-Ω output impedance. This low impedance output can be connected to an oscilloscope so the typical scope 1-MΩ input impedance will not further load the circuit being measured.

The input stage cathode follower circuit provides an input impedance of 1 MΩ, into a low output impedance cathode follower step-attenuator with a total resistance of approximately 6.3 kΩ. The cathode attenuator is switched, alternately with the two series 1000:1 ratio input resistors, to change the meter voltage ranges. The output stage full wave rectifier actuates a 1 mA meter movement. The wideband amplifier is substantially flat from 10 cps to 1 Mc [2]. Because the amplifier employs negative feedback, it is extremely stable. The accuracy of the meter reading is virtually independent of line voltage changes and tube characteristics. It can also serve as a wideband amplifier by using the output terminals.

HP 400 series VTVM family picture

The **HP 400A** rms AC voltmeter/amplifier, described above, has nine voltage ranges from 0.03 V_{fs} (volts full-scale) to 300 V_{fs}, with an overall meter accuracy within

Figure 2. HP 400H Front Panel.

±3% below 100 kc and ±5% from 10 cps to 1 Mcps. Line voltage variations from 105 volts to 125 volts or changing tubes will affect the reading by less than 3% at all frequencies below 100 kc. Input impedance is 1 MΩ shunted by 16 µµF [2] for 30 V and below, 3 MΩ on the 100-V range and 2.4 MΩ on the 300 V range. It offers service as a wideband amplifier, using the output terminals on the upper right side of the case.

The **HP 400B** rms voltmeter/amplifier has the same slope-front panel as the 400A. It also has nine ranges from 0.03 volt full scale to 300 V_{fs}. However, its wideband frequency capability is restricted to 2 cps to 100 kcps. Input impedance is 9 MΩ

Figure 1. The original: the HP400A VTVM.

HP400A Ye Olde Sales Pitch

This excerpt from the HP instrument catalog describes its features:

The HP Model 400A vacuum tube voltmeter, a recent development of HP Laboratories, possesses all the important desirable features. It is one of the best available yet its accuracy is unexcelled and it has extreme sensitivity over a wide frequency range. One of the outstanding features of the HP 400A is that the voltage indication is proportional to the average value of the full [sine] wave. This is a feature not found in most electronic meters on the market today.

AC voltages as small as 0.005 and as high as 300 volts can be simply and directly measured without any precautions over a frequency range of 10 cycles to 1,000,000 cycles. Accuracy of readings is assured because the high input impedance does not disturb the usual circuit under test. Furthermore, the calibration error of the instrument under all conditions is less than 3% to 100 kc and less than 5% to one megacycle.

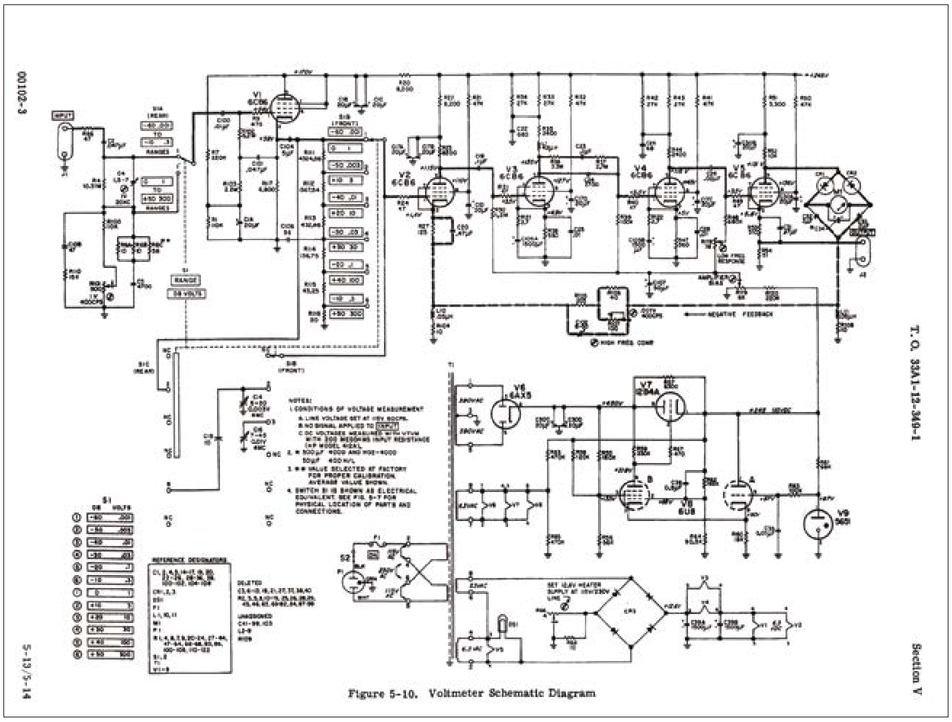


Figure 3. HP 400H schematic diagram.

at 100 cps to 4 MΩ at 100 kcps. It was available from 1950 to 1952.

The **HP 400AB** rms voltmeter/amplifier, introduced in 1955, is the first of the tall case flat-front panel VTVMs, and has a wider voltage range of 0.003 V_{fs} to 300 V_{fs} in 11 ranges. The dB ranges are –60 to +50 dBm. The frequency range is 10 cps to 600 kcps. Input impedance is 10 MΩ shunted by less than 25 µµF.

The **HP 400C** is a wideband rms voltmeter/amplifier with the same slope-front panel as the 400A and 400B. It is capable of reading from 0.001 V_{fs} to 300 V_{fs} over 12 linear voltage ranges. Its wideband frequency response is from 10 cps to 2 Mcps. It was available from 1950 to 1952.

The **HP 400D**, introduced in 1955, has the tall case, but with the smaller meter from the HP 400A and 400B. It has 12 linear rms voltage ranges from 0.001 V_{fs} to 300 V_{fs}, with a frequency range of 10 cps to 4 Mcps. The meter also has 12 log dB ranges from –72 to +52 dBm, where 0 dBm is 1 mV into 600 Ω. There is a dB conversion chart in the Manual for other impedances. The meter input impedance is 1 MΩ in parallel with 15 µµF from 1 V to 300 V, or 25 µµF from 0.001 V to 0.3 V. Rack-mounted versions were also available as the **400DR**.

The **HP 400E**, **HP 400F** and **HP 400G** are small solid-state AC voltmeters with linear voltage and log dB ranges and a mirrored scale to compensate for parallax. They all have 10 MΩ input impedance. The **400EL** and **400FL** have linear dB scales and log voltage ranges. The 400E has twelve ranges from 0.001 V to 300 V_{fs}. The **400F** has fourteen ranges from 0.1 mV_{fs} to 300 V_{fs}. The **400G** is a dB only meter with eight ranges from –80 to +60 dBm.

The **HP 400H** VTVM, the subject of this Retronics article, has the same specifications as the 400D, but has a larger meter with a mirrored scale. It has better accuracy than the 400D and 400L (**Figure 2**). Rack-mounted versions were also available as the **400HR**.

The **HP 400L** has the large mirrored scale meter, but without the mechanical zero adjust found in the 400D and 400H. It also has a log voltage scale and a linear dB scale. Rack-mounted versions were also available as the **400LR**.

HP 400H circuit description

The schematic diagram from the *HP 400H Operating and Service Manual* is shown in **Figure 3**. This manual also covers the HP 400D and HP-400L VTVMs.

The input voltage is fed through DC blocking capacitor C2 to the input divider/attenuator, with all the resistors and C4 frequency compensating cap. The signal at the grid of the V1 cathode follower is obtained directly from the input jacks for input ranges from 0.001 to 0.3 V_{rms}. S1A switches to a 1/1000 divider so the maximum input voltage is limited to 0.3 V_{rms} at input ranges of 1 V_{rms} and higher. V1 provides a constant input impedance of 10 MΩ for all twelve meter ranges. Three trimmer capacitors are provided to ensure flat response up to 4 Mcps from the input divider.

The cathode of V1 feeds the S1B stepped attenuator with a total resistance of about 6.3 kΩ, and reduces the measured voltage level so 1mVrms or less is applied to V2, the meter amplifier input stage. The cathode follower provides a gain of 0.95.

The four-stage broadband voltage amplifier section consists of V2-V5. The amplifier provides between +55 and +60 dB of gain, with about 55 dB of negative feedback at mid-frequency. The meter rectifiers and the meter are part of the feedback loop (the dashed line in the schematic between the meter bridge and the cathode of V2). Various fixed inductors, trim-pots and trimmer caps adjust the tube bias and feedback to limit low and high frequency response for stable operation from 10 cps to 4 Mcps.

The full-wave meter rectifier circuit is connected as a bridge, with the meter connected across its mid-point. The current through the meter is proportional to the plate voltage of V5, with a magnitude proportional to the rms value of a sine wave.

The power supply consists of V6-V9. The high voltage power transformer winding is full-wave rectified by V6 and filtered by C30C-C30D. V9 is an 87 VDC cold-cathode glow-discharge voltage reference tube that is connected to the grid of V8A. V8B and V7 create a linear series regulator that reduces the +450 VDC voltage from rectifier V6 down to the +245 VDC bus for the plate and screen supplies of V1-V5.

The heaters for V1-V4 are connected in series-parallel across a selenium rectifier full-wave bridge which is heavily filtered by C39A-C39B. This DC heater supply reduces the amount of AC power line "hum" introduced into the input stage V1 and the broadband amplifier. Only the final amplifier stage V5 is powered by an AC filament winding. The high voltage rectifier and regulator sections V6-V8 have their own separate AC filament winding.

My HP 400H — a closer look

This particular 400H came with a pallet of other electronic equipment that the company I work for bought to outfit the test lab. The case is a bit worn, the leather handle has deteriorated and the dark gray paint is worn off in places. I found it strange that the yellow calibration seals were intact, but there was no calibration date sticker. There were also two typed paper labels, one on top said “Property of AT&T” and one on the front said “ENG LAB”.

Like all the tall-case bench-top chassis HP units, when you remove two screws above the rear panel power cord opening you can easily slide off the chassis cover. The heads of the two Phillips-head screws on this unit were stripped and I had to remove them with locking pliers. They were difficult to remove and the ends of both threads were damaged, as if the wrong screw threads might have been used.

Inside, it looks like it has the original tubes since most had “hp” logos. There is what appears to be a trail of something that leaked out of the C105 can capacitor on the right near octal rectifier tube V6. This trail goes down past V3 below the can, down behind the large power transformer, and onto the bottom rail of the case (**Figure 4**). The upper right corner of the chassis also shows evidence of cooked dust from overheating on the chassis near V9 and V7 at the upper-right.

Since the interior of the chassis is also dusty, perhaps from long-term storage, I removed the tubes one-by-one to clean them off; dust can’t be good for proper heat dissipation. When I went to remove V9, the 5651 glow-discharge voltage regulator tube, the bulb broke cleanly away from the base (**Figure 5**), which explains the white powder at the top. V9 is the 87 V reference for the +245 VDC regulated power supply, and with it open circuit the voltage may have increased above the 450 VDC rating of HV filter cap C30, degrading the dielectric. It might also have put the rectified +450 high voltage onto all five 6CB6 amplifier tube circuits for who knows how long.

Further evidence of the cascading failure was that the socket for the V7 12B4A HV regulator tube was dark brown rather than light brown like the other tubes. In light of these visible problems I decided not to power up the HP 400 until I did some further investigation.

I cleaned up the chassis with isopropyl alcohol (IPA) and cotton swabs. Fortunately, the 5651A and the 20/20/20/20 µF C30 can capacitor are available from Antique Electronic Supply; the cap being even a bit higher in voltage at 475 VDC. The other +450 VDC caps, C1 and C17, are in plate or screen grid circuits and were protected by series resistors.

There are no overheated parts evident on the left side section of the chassis (**Figure 6**). I always worry when I see a selenium rectifier. CR3 is used to make DC for the four input amplifier tube filaments, but it still looks to be in good shape. The triple 1500 µF 15 VDC filter cap C39 for the filament voltage also looks okay. My manual schematic shows that only C39A and C39B are used, but my HP 400H has C39C connected to the negative end of CR3. This was probably a later change that was not listed in the Manual Backdating Changes list in the back of this manual. Our Calibration Lab people at Bendix were always very good at installing any manufacturer recommended updates into our test equipment.

Next time we continue this story with the initial testing of the instrument — and discover more issues to deal with.

(140534)

References

[1] Hewlett Packard Model 200AB Audio Oscillator; Retronics, Elektor March 2014, pp. 78-82

[2] Note: I used the electrical unit notations that were used back in the era of the HP 400 series VTVMs; frequency in cycle per second (cps) rather than Hz; and input shunt capacitance in µµF rather than the contemporary pF.

Figure 4. Capacitor leakage and overheating.

Figure 5. Cleaned chassis with V9 broken off at the base.

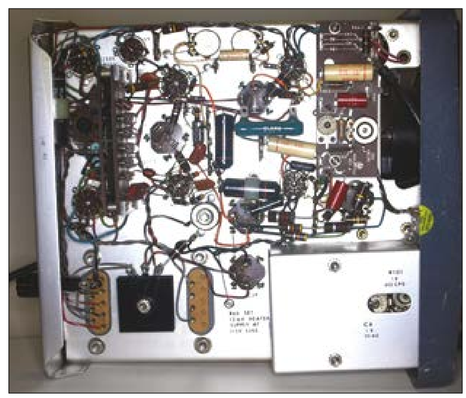


Figure 6. Left side of the HP 400H chassis.

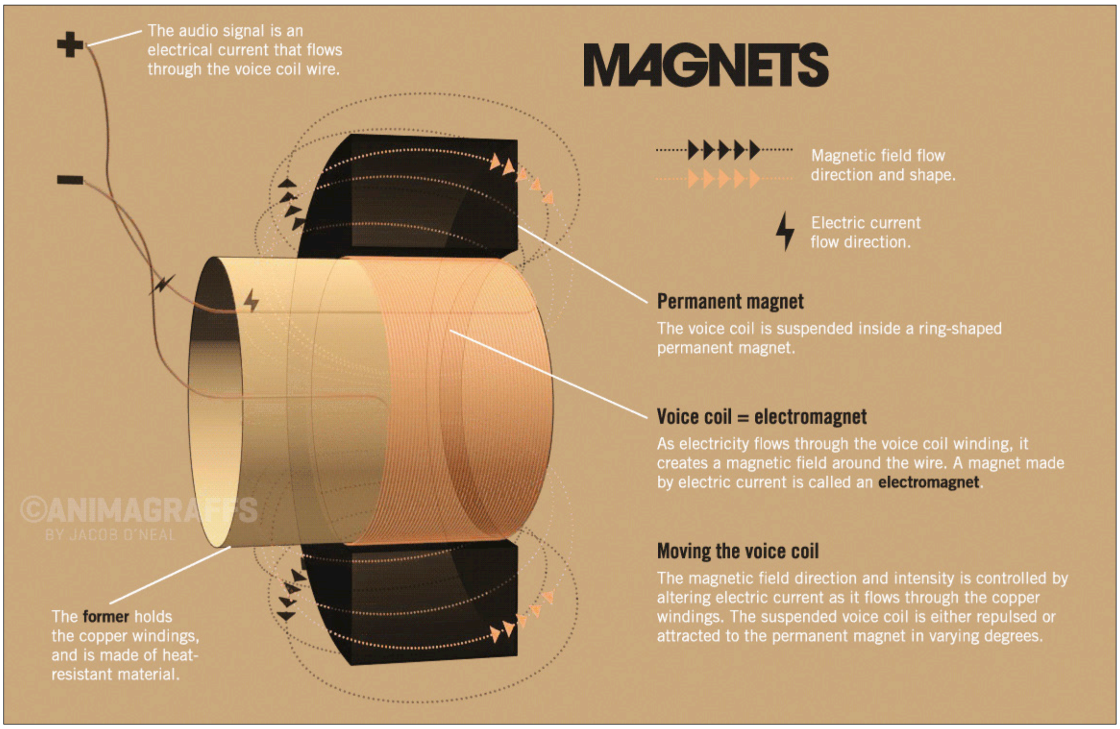
The Author

Chuck Hansen is an Electrical Engineer and holds five patents in his field of engineering, and works as a consultant in the aerospace industry. He has written two books for Audio Amateur Publications, and has over 260 magazine articles to his credit. Chuck began building vacuum-tube audio equipment in college. He enjoys sailing and playing jazz guitar. He likes to modify guitar amplifiers and effects to reduce noise and distortion, as well as building and restoring audio test equipment.

Animated Electronics

One animation is worth more than 1 Kwords

By Harry Baggen (Elektor Netherlands Editor)



Trawling the Internet for websites that explain how electronic components and circuits work will quickly find multi-page text descriptions which, together with a few illustrations, attempt to explain what is going on. This is very interesting and instructive indeed, but in these hectic times even electronics engineers (who are well-known for their immeasurable patience and perseverance) sometimes have the desire for a more concise and speedier information transfer. This is where an animation is entirely appropriate.

Unfortunately many things from the electronics domain are not so easily explained with the use of moving pictures. This is much more easily done with mechanical things, where you can see the individual parts actually move in the animation and the connection between different components becomes clear. In our quest for such animations we have nevertheless found a few that will be very interesting to our readers. Some explain how semiconductors work, others (the majority) deal with electric motors. And the most splendid... you can see above! More about this later.

Semiconductors

The first animation that we present here shows how a MOSFET works [1] (**Figure 1**). While there is not much that moves in the image, the nice aspect here is the interactive nature of the design. Using the mouse, you can set different drain-source and gate-source voltages and you will see the electron flow and the corresponding operating point in the ID/VDS characteristic. There is even a small quiz where you can test yourself whether you have understood everything correctly.

A similar design can be found at [2], which shows how electrons move through the layers of a bipolar transistor. Here too you can adjust a few things with the mouse. You can turn the power supply voltage on and off and (the most important) using a slider you can set the base-emitter voltage between 0 and 0.7 V. From 0.6 V electrons will begin to flow between the emitter and collector, provided you turned on the power supply voltage first. The accompanying text explains what is happening. It is also possible to see the descriptions of the various parts when the label function is switched on.

A third semiconductor example is the operation of the 555 timer [3] (**Figure 2**). Although there is nothing here that you can adjust (except the speed of the 'simulation'), the animation will nevertheless quickly show you how the well-known 555 operates internally. You can follow how the capacitor that determines the time is charged and discharged, and how the internal opamps switch over at certain voltage values and subsequently change the state of the flipflop at the output. If you have to explain this in words then you will be busy for a while!

Electromechanical parts

Animations on many websites explain how different types of electric motors work. Some sites achieve this with very simple moving images, such as [4]. While the images are minimal-

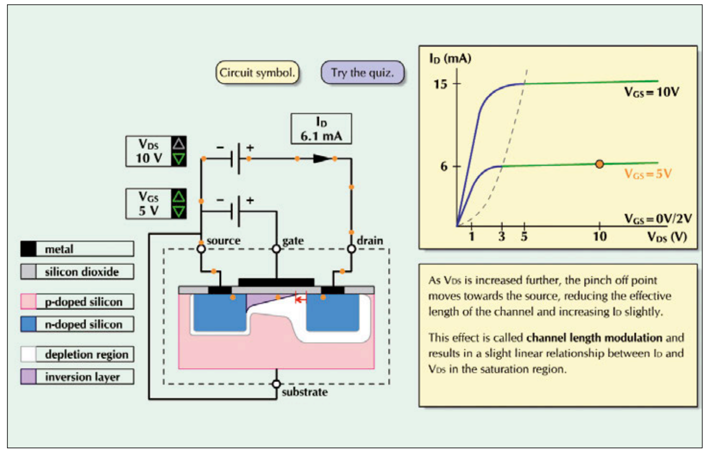
ist the operation of various motors is nevertheless explained very well. DC and AC motors, induction motors, three-phase motors, loudspeakers, practically everything that moves with the aid of coils is covered here.
For those who are looking for an animation that explains a brush-less DC motor, will find a nice example of a 4-pole version, with corresponding drive signals, at [5] (**Figure 3**). Unfortunately it is not possible to change the speed of the animation, but it is possible to stop it and step through one step at a time.

Highlight

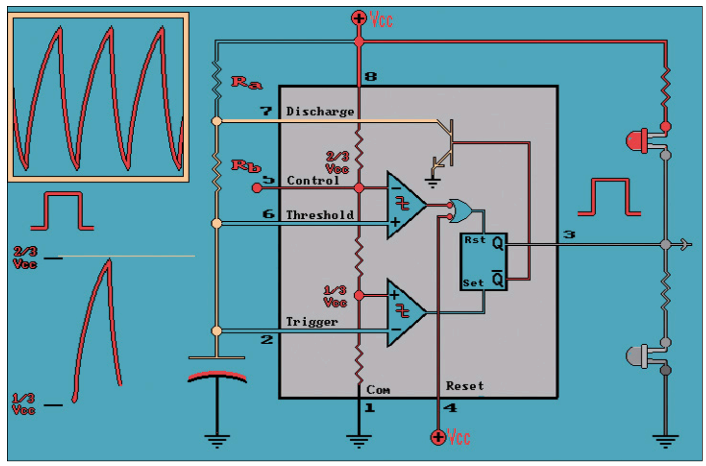
To conclude we present a website with the most splendid of animations that we've ever seen in this domain. This is the website 'Animagraffs' by the graphic designer Jacob O'Neal. He shows very extensive animations of various devices on his website, such as an aircraft jet turbine, a car engine and a pistol. But what interests us the most are the animations of a loudspeaker [6] (see introduction photo) and a flat screen display [7]. It is unbelievable how much time he has invested in this. This is not just one moving image, but a complete explanation which shows that he has thoroughly familiarized himself with each subject. The animations are truly fantastic. The loudspeaker animation not only shows the movement of the voice coil with the cone, but also the behavior of the magnetic field lines between the voice coil and the magnet, the effect of the suspension on the cone, the spreading of the air molecules by the movement of the cone and much more. The animation for the flat screen display does not only show the operation of a complete panel, but also examines separately the operation of the backlight, the display matrix and the behavior of the liquid crystals between the electrodes. Once you have seen these animations you will most likely also explore the other subjects on this website, even though they have nothing to do with electronics. This certainly deserves an A+ if we were to grade such websites!

(140529)

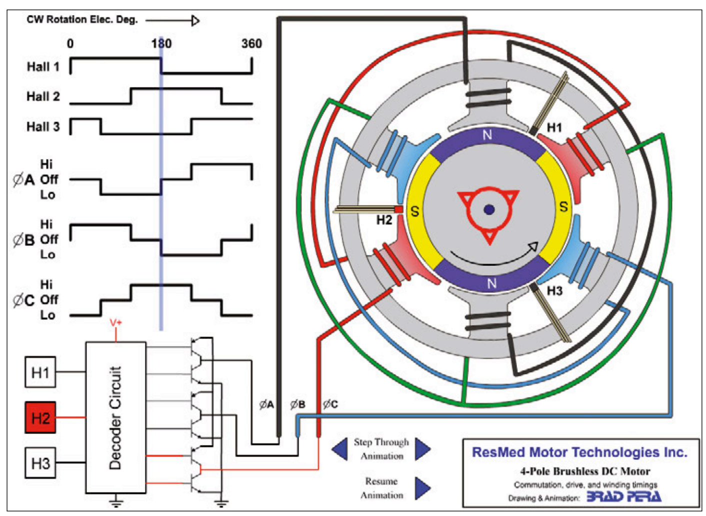
The fantastic animations by Jacob O'Neal deserve an A+



1



2



3

Web Links

[1] www-g.eng.cam.ac.uk/mmg/teaching/linearcircuits/loader.swf?device=mosfet.swf

[2] www.learnabout-electronics.org/Downloads/Fig316dl_bjt_operation.swf

[3] www.williamson-labs.com/pu-aa-555-timer_med.htm

[4] www.animations.physics.unsw.edu.au/jw/electricmotors.html#DCmotors

[5] http://educypedia.karadimov.info/library/4-pole_bldc_motor.swf

[6] http://animagraffs.com/loudspeaker/

[7] http://animagraffs.com/flat-screen-displays/

I have a Dream

10,000 registrations and counting

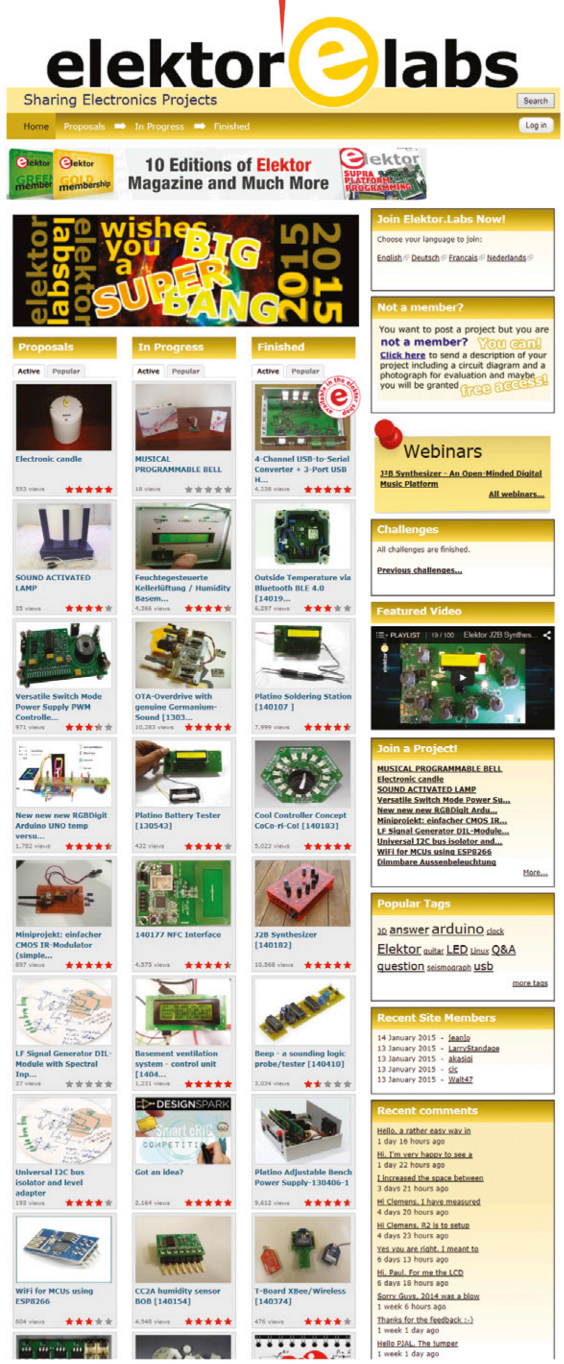
By Elektor Clemens Valens (Elektor.Labs)

Since the launch of the Elektor.Labs website almost three years ago the number of people who took the time to create an account has steadily risen to the magic barrier of 10,000 (10 K). It happened on our servers last Christmas.

Ten thousand People who followed the Star of Elektor to be guided to the Elektor.Labs website, an online shed where electronics projects are born, nourished and raised before being paraded to the world in… (drumroll) … Elektor Magazine. Ten thousand People, filled with joy and delight, came to Elektor**.**Labs and brought with them an impressive amount of offerings and gifts in the shape of brilliant ideas, great suggestions, exquisite contributions, sharp comments and wonderfully designed projects for sharing amongst all of them, and more. They all sat down and started to pass around tiny objects made of sand and rare materials, and skillfully attached them to emerald-colored boards decorated with intricate designs drawn in precious metals like silver, copper and gold using strangely glowing and humming hot sticks, liberating captivating scents and thin curls of smoke. Then, when the People had finished sculpting those fragile works of art, they attached them carefully to the source of life, always respecting polarity, and… the works worketh as expected!

— Hey, Clemens! Wakey-Wakey!
— Huh? Wha… Where am I? Oh, was it all just a dream? A beautiful dream? Bummer.

Well, I guess the "worketh as expected" was an illusion, but the 10 KPeople is true. Post a project and reach ten thousand e-people or more. Isn't that a nice audience? 10,000 is quite a number. People often find it difficult to appreciate large numbers. They tend to count like 100, 1,000, 10,000. They dream of how they will sell one thousand pieces of the gadget they just came up with, or, why not, ten thousand? However, before you get to ten thousand you must first pass two thousand. It's often already pretty hard to sell the first thousand pieces and then you have do it again to get to two thousand; repeat that five times to reach ten thousand. The number that follows 1,000 is 1,001, not 10,000. One thousand already is a huge number.



Think of it this way: If you have one client or registrant a day, every day of the year, you would have to wait for more than 27 years to get to ten thousand. Elektor**.**Labs reached this number in one thousand days, which averages to ten newly registered users every day. That's something that doesn't stop to amaze me. Every time I look at the Recent Site Members block on the Elektor.Labs homepage I see new members and it makes me wonder if it will ever stop. Of course it will. Even Michael Jackson, several times elected world's most famous person, is not known by every person on the planet although he got pretty far. It is reasonable to assume that there are fewer Elektor readers, or even fewer electronics enthusiasts than people that like to listen to Michael Jackson songs. However, this doesn't stop me from dreaming; dreaming of passing the 11,000 barrier or maybe 100,000? 1,000,000? 1,000,000,000? A smile appears on my face, a drop of saliva dangling from a thin wire slowly descends from the corner of my mouth… I love dreaming.

(140533-I)

Web Link

[1] www.elektor-labs.com

Err-lectronics

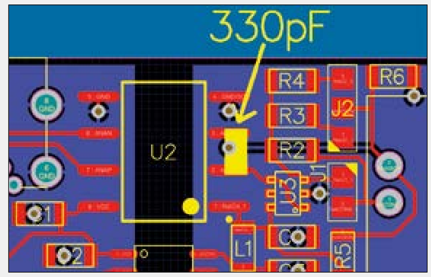
Corrections & Updates to published articles

Compiled by **Ralf Schmiedel** & **Jaime González-Arintero**

What's electronics without typos, bugs, and wrong pinouts? We love robots but we're not planning to become part of *their community* yet, and thus, unfortunately our imperfections keep slipping into projects and articles. Sorry about that! However, this here section is not about errors *per se*, but also on everything up for improvement or updating in one way or another. Look at the bright side…

A Patch for the Probe

As seen in *Isolated Oscilloscope Probe*, Elektor 9/2014, page 46 (130297).



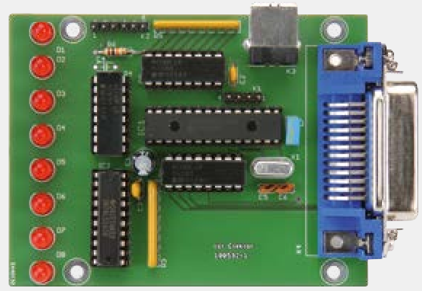
The gain of the AMC1200's insulation transformer is actually fixed by default at 1:8. However, in reality, with some specimens it becomes unstable, and can only be operated at 1:7.6 (max.). In that case, it's useful to implement a 330-pF ceramic capacitor between the signal inputs VINp (pin 2) and VINn (pin 3). A standard 1206 or 0805 capacitor does the job, soldered directly between the pins of the AMC1200. The picture shows the "patch" with the capacitor mentioned.

Besides, we found out that 10-ohms "emergency" resistor R1, intended to limit the current of the USB in case of a failure, was unfortunately overdimensioned. At a low USB voltage, the voltage on the resistor could drop, causing the DC/DC converter of the ADuM5242 to malfunction, affecting the isolated supply voltage. Thus, it's highly recommended to use a 1-Ω resistor instead, or even to omit R1.

Erik Lins (author of the original article)

Gee-Whizz, a GPIB-to-USB Converter

As seen in Elektor 7&8/2012, page 48 (100592)

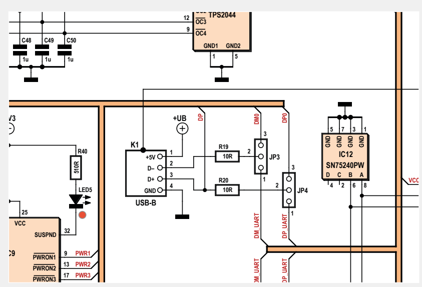


On Elektor's English forum, author Anders Gustaffson is helping other readers with updates for his Gee-Whizz GPIB project. GPIB converters are rare and costly, no wonder why this project has staunch followers. Anders has kindly released new software, so don't miss the conversation this time!

[j.mp/Gee-Whizz]

USB Hub feat. RS-232/RS-422/RS-485

As seen in Elektor 11/2014, page 10 (140033)

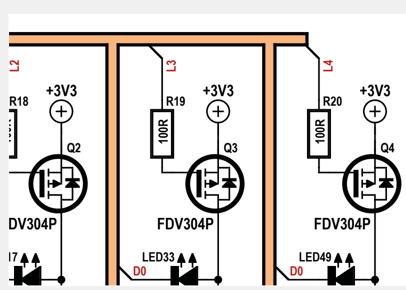


Oops! Unfortunately, there was an error in the schematic, regarding the USB-B connector K1. In reality, pin 1 in the USB-B (+5V) is connected to +UB. Pin 2 in the USB-B (–D) is connected to JP3 via R19. Bus line 'TXD_C' with pin 5 of IC8 and pin 48 of IC1 should be labeled TXD_D. The corrected schematic (140033-SCH.pdf) including all changes may be downloaded at [www.elektor-magazine.com/140033].

The project PCB and the ready-populated module are not affected.

Programmable Christmas Tree

As seen in Elektor 12/2014, page 28 (140371)



In the original schematic, the four FDV304P MOSFETs were drawn incorrectly, their drain and source pins were swapped. The corrected schematic is printed here.

Besides, although in the article we mentioned that the source code as well as the hexadecimal file could be downloaded in the project site, unfortunately we were not allowed to publish it freely for not owning the rights to one of the libraries used. Once again, our apologies!

Compiled by **Beatriz Sousa**

CRAZY X-MAS SALE: *CRAZY - CRAZIER - INSANE*

Last year we went all the way with a Crazy Christmas Campaign, doubling traffic and turnover in our four stores. We granted over €150,000 in discounts with rebates up to 58% on our best liked products. Unsurprisingly, the 62-LED Christmas Tree was the top seller. The Raspberry Pi Advanced Programming book and the Elektor Green Membership also ranked among the most popular. During this period we sent 47 good-news messages to 165,000 readers of our Elektor. POST newsletter and achieved a 73% open rate. More than 4,000 people also enjoyed our online Advent Calendar with about 100 prizes in stock.

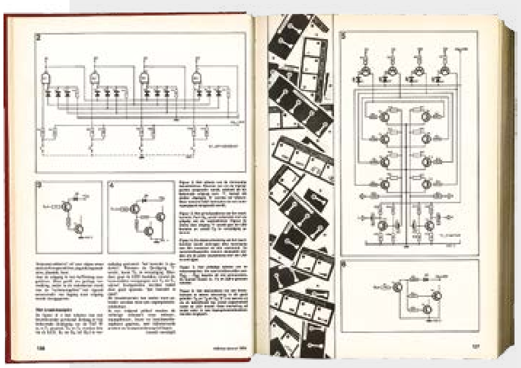
GREEN MEMBERSHIPS GROWING FAST

"...Altough I prefer my Elektor books in print, I have chosen a Green, online membership for the mag because of the great projects that have been published over the years. This and the 10% member discount in the Elektor Store was my main reason to join Elektor. Having to download my magazine content is okay with me; it's better for the environment too..." One of the responses from our members we got as a result of a survey. From every 10 new members, 6 take out a Green Membership. If you want to change your Membership from Gold to Green or you want to join Elektor as a 'fresh member', go to **www.elektor.com/green-membership**.

READ ONLY MEMORY

Elektor magazine and its parent publishing company boast a long and rich history. In this space we picture a gem from the past.

Touchscreens are very innovative and hot? Duuhhh ...



In December 1974 issue Elektor published a set of schematics for a contraption called TAPKAST. The TAPKAST article aimed to show how to make black & white touchscreens for amplifiers.

Elektor now also available in three Asian languages ...

Our English digital weekly newsletter Elektor.POST can now be read by more than 3,000 Chinese, Japanese and Korean students, teachers and engineers.
Perfect time to appoint a special publisher — John Moore — who has been operating since November 2014 from our Tokyo office. John is a UK citizen, but has lived and worked for over 26 years in Asia as a renowned MAKE Publisher and President. The first result of John's work for Elektor is a license contract for Elektor to be presented and distributed in Beijing by China Machine Press, China's number two trade publisher. Now Korea, Taiwan and Japan are also negotiating to launch a selection of Elektor's magazines, specials and books in their markets.
Elektor's author network is vastly expanding to include the best experts from Asia, which has resulted in the acquisition of a popular title *3D Printing Projects* (Softbank Publishing).

For more information, contact Raoul Morreau on: raoul.morreau@eimworld.com

PEOPLE NEWS • Producer Dave Ridgway has been appointed Book Promotor in our Global as Client Executive • Cumhur Cakmak has started producing ELEKTOR.POST in Turkish PgHHHE about his new book 'Électronique pour les débutants', more on www.elektor.fr/debut special about DRONES • German E-marketeer Muhammed Sökük is preparing an Arduino

Team ● Nicole Crombag joined the German team ● Rémy Mallard has launched a video: goo.gl/ ● Editor Dré de Man is working on a summer 'crazy campaign' for the spring ●

EXPERT PROFILE

Elektor works closely together with more than 1,000 experts and authors for the publicatin of books, articles, DVDs, webinars en live events. In each installment of Elektor Word News we put one them in the limelight.

Name:
Professor Dr. Dogan Ibrahim

Age: **57**

Education: **First Honours class in Electronic Engineering**

Publications:
> 66 on microprocessors, microcontrollers, and PC based real-time system design

Training: **> 50 short courses on computers, software engineering and related fields**

Who is Professor Dr. Dogan Ibrahim?
I am 57 years old, married with two children: a boy and a girl. Currently I am a Visiting Lecturer at the Near East University in Cyprus and hold PhD degree from the City University in London in the field of Digital Signal Processing.

What is your experience?
I have many years of mini-computer experience using PDP11 series from DEC and VAX series of hardware. Over 30 years with languages ranging from assembler to C# and Python. In addition to many years using popular microcontrollers.

Who is your biggest role model in electronics?
The person to admire in electronics is certainly genius Thomas Edison who was a prolific inventor holding 1,093 US patents to his name and many European patents.

What will be the most key electronics development?
Electronics grows so fast that it is difficult to predict the future. I think the Internet of Things (IoT) could be the next big development in electronics, especially IoT based robotics. It provides comfort and help to our daily lives.

What topics will you be writing about in the future?
It is hard to estimate the future, but I think most of my future technical books will be based on intelligent systems and their applications in everyday life.

Suppose Elektor gives you £100. What would you buy? Why?
If I were given £100 I would probably buy an Intel Edison Breakout board. This small board with its 500-MHz Atom CPU and cloud connectivity seems to be the ideal environment for the future IoT projects.

What was THE development of electronics in the 1980s?
I was in my 20s when the first handheld calculator arrived and it was a fascinating piece of electronics at that time. This was an 8-digit LED calculator with 4 functions. Before that we were using the slide rule in our engineering calculations.

Hexadoku

The Original Elektorized Sudoku

With Hexadoku already in the Electronics Hall of Fame there's no need really to tell you what's found on this page. Here's a freshly made puzzle for you to have a crack at. Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of five Elektor book vouchers.

The Hexadoku puzzle employs numbers in the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.
Correct entries received enter a prize draw. All you need to do is send us **the numbers in the gray boxes**.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for five Elektor Book Vouchers worth **$70.00 / £40.00 / €50.00 each**, which should encourage all Elektor readers to participate.

Participate!

Before March 1, 2015, supply your name, street address and the solution (the numbers in the gray boxes) by email to:

hexadoku@elektor.com

Prize winners

The solution of the December 2014 Hexadoku is: **D7085**.

The €50 / £40 / $70 book vouchers have been awarded to: Stefanie Kalkbrenner (Germany); Alfred Hoks (Netherlands); Joe Young (Canada); Joseph Reding (Switzerland); Harjeet Singh (India).

Congratulations everyone!

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

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

Looking to combine 2D multi-touch & 3D gesture recognition in one PC peripheral?

Microchip's 3DTouchPad gives you the first 2D/3D input sensing Development Platform

Microchip introduces the 3DTouchPad, a production-ready development kit and reference design which combines 2D tracking of up to ten fingers, with 3D air gesture recognition for fast development of advanced input sensing for PC peripherals and other applications.

Based on Microchip's GestIC® technology, 3DTouchPad's robust and innovative 3D gesture recognition technology offers a detection range of up to 10 cm, whilst the highly responsive 2D projected-capacitive multi-touch input sensing supports up to 10 touch points and multi-finger surface gestures.

The integration of Microchip's new MTCH65X high-voltage capacitive touchscreen line driver enables robust projected-capacitive touch performance as well as larger sensor sizes and a thicker cover material by increasing the Signal-to-Noise Ration (SNR).

As a plug-and-play PC peripheral, 3DTouchPad connects to a PC using a single USB cable and includes a free Graphical User Interface (GUI), Software Development Kit (SDK) and Application Programming Interface (API). It also offers out-of-the-box driverless features to enhance the user experience for Windows® 7/8.X and MacOS®.

KEY FACTS

- 3DTouchPad 2D/3D input-sensing Development Kit: DM160225
- MTCH65X 2D projected-capacitive touchscreen line driver
- GestIC® technology for advanced 3D gesture recognition

For more information: www.microchip.com/get/eu3DTouchPad

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless