



Q's-4-2day
- What's an FPGA then?
- What can it do 4 me?
- What applications pls?
- Price? Free dev tools?

FPGA

taming the beast

● 1970's NAGRA IV Tape Recorder ● Arduino AC Grid Analyzer | Electronics 4 Starters

Hyperactive LED Xmas Tree | DIY FPGA Dev system

● Internet of Things ● **What's cooking @ E-Labs** | Missing Ground Plane | **PWM to Phase Shifter** ● Elektor World



Crystal-free 8-bit USB PIC® microcontrollers cut system costs and power consumption

0.25% clock accuracy enables USB connectivity, eliminating the need for external crystal



Microchip's lowest-cost and smallest-form-factor USB microcontrollers (MCUs), feature pin counts of 14 to 100 pins and are the first 8-bit MCUs to integrate LCD control, battery-backed RTCC, and USB on a single chip.

Microchip's latest USB PIC® MCUs feature internal clock sources with 0.25% clock accuracy to enable USB connectivity with no external crystal. They are also the first USB MCUs to combine pin-counts ranging from 14 to 100, with high peripheral integration and up to 128 KB of Flash. The eXtreme Low Power (XLP) technology also keeps power consumption down to 35 μ A/MHz in active mode and 20 nA in sleep mode.

Lowest-cost and smallest-form-factor

The PIC16F145X MCUs give you USB connectivity and capacitive touch sensing, in addition to a wide range of integrated peripherals with footprints down to 4x4 mm.

High-performance touch-sensing with USB

With an integrated Charge Time Measurement Unit (CTMU) and 1.8 V to 5 V operation, PIC18F2X/4XK50 MCUs are pin-compatible with legacy PIC18 MCUs, giving an easy migration to higher-performance.

USB plus LCD control and a RTCC with Vbat

The PIC18F97J94 family gives you USB connectivity with LCD control, and a battery-backed real-time clock calendar (RTCC), all on a single 8-bit PIC® microcontroller.

GET STARTED IN 3 EASY STEPS:

1. Choose a peripheral mix and pin count to suit your application
2. Use the free USB stacks and software drivers for faster design
3. Start developing with low-cost development kits

For more information, go to: www.microchip.com/get/eu8bitUSB



Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

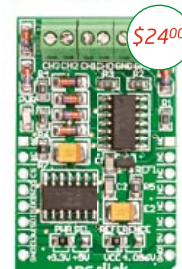
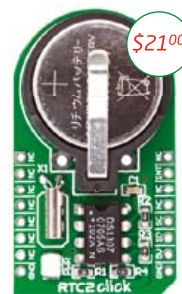
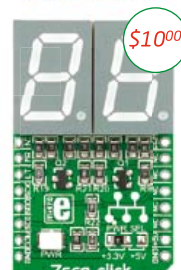
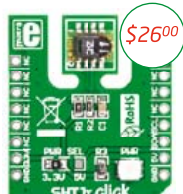
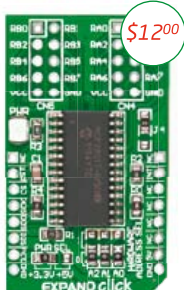
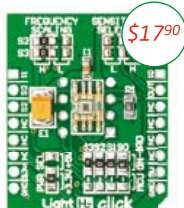


one SOCKET



for ALL boards

Just place the **Click™** board you need into the innovative new **mikroBUS™** socket and it's ready to work right away.



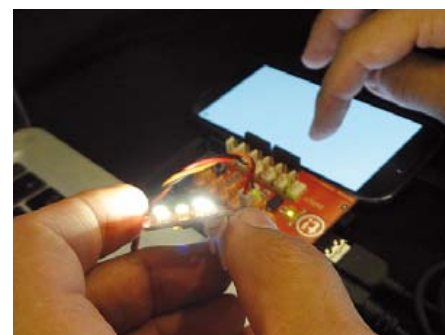
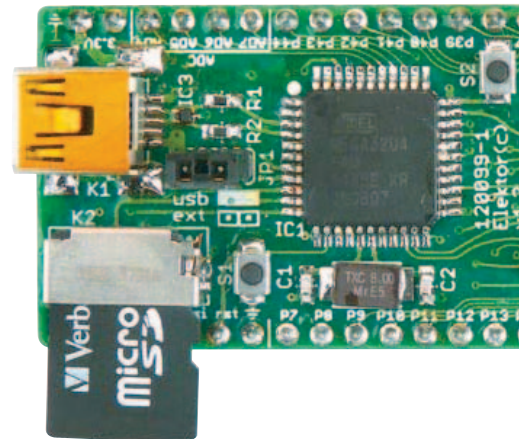
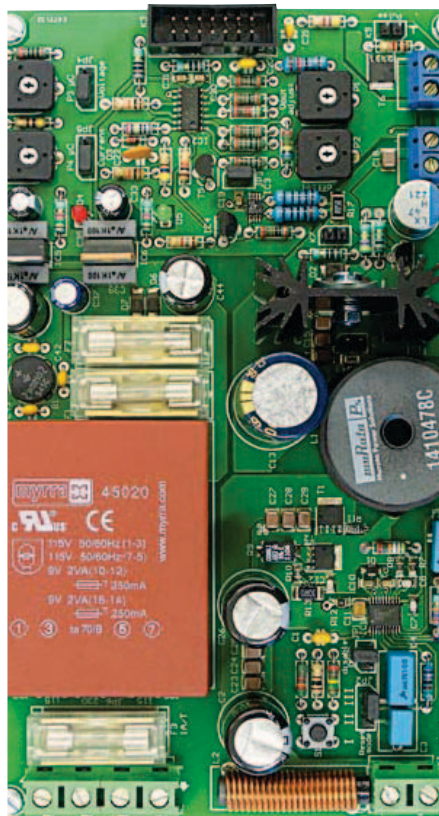
MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

www.libstock.com

GET IT NOW
www.mikroe.com

NOTE: Two 2x8 male headers are included in the package with each click board.

elektor Magazine



● Community

8 Elektor World

- Elektor Improved Radiation Meter on ConSat-1... soon
- Elektor Improved Radiation Meter @ Oxford University... soon 2
- Elektor Monster Formant hits YouTube
- Lost in the Electronic Maze
- Circuit Cellar 25th (19hex) Anniversary Party
- RF Poetry



● Labs

12 Elektor hot Labs

Clemens Valens on what's brewing, smelling and burning at Elektor Labs, including their new website.

14 MIC502 & MAX6643

Raymond's pick of the month in terms of components.

16 Tiny but not insignificant

The story of stray capacitance in a FET preventing a microprocessor from being programmed.

17 Missing ground plane!? Check your board with PCB Visualizer

Through a slip-up in one of our circuit board designs we learned about this great PCB preproduction checking tool.

18 What's Cooking?

Not Elektor's new SMPSU we hope! A look at an early prototype.

● Projects

20 Taming the Beast

An upbeat introduction to FPGAs, followed by a description of Elektor's DIY FPGA Development System.

28 Arduino AC Grid Analyser

This circuit performs a rigorous quality check on the AC line voltage from the wall outlet. It's got FFT and a mini spectrum analyser, courtesy of an Arduino board.

36 Hyperactive LED Xmas Tree

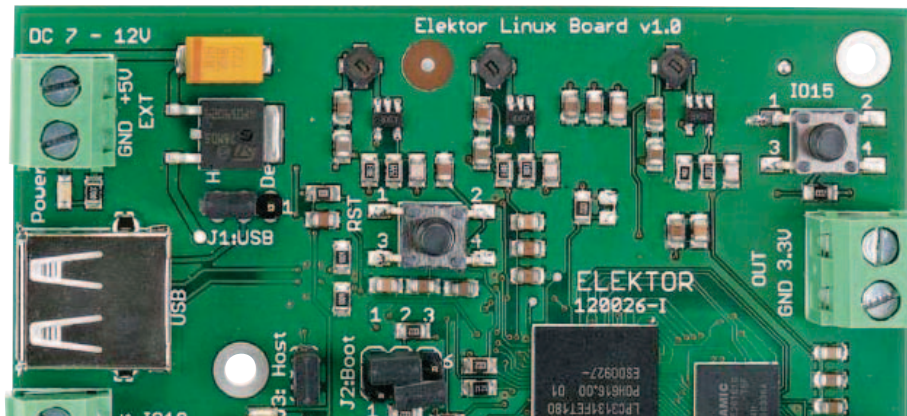
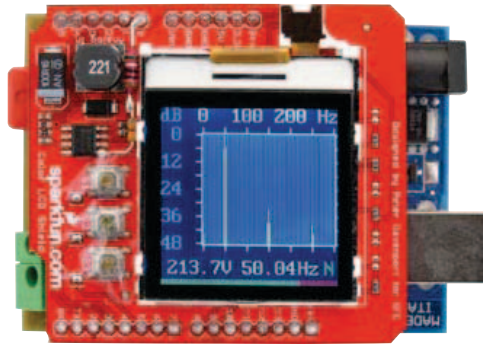
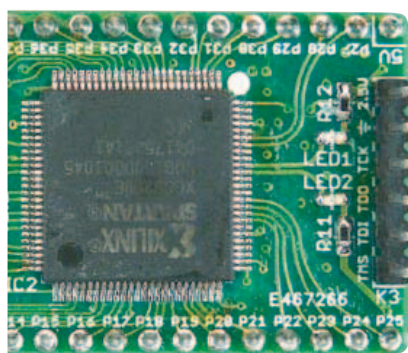
There's no end of lively visual patterns generated by this PIC controlled gizmo, especially if you build two.

40 Embedded Linux Made Easy (6)

This month the Elektor Linux board gets connected to a typical home network using a server application we write ourselves.

46 Arduino on Course (3b)

Can your Android smartphone read sensors directly? Probably not, and that's where an Arduino Mega comes in handy.



● Industry

52 Electronics for Starters (10)

In this tenth and concluding part we look at basic circuits operating at radio frequency (RF).

58 Transconductance Amplifier

The principle of converting a differential voltage into a proportional current allows a versatile instrumentation amplifier to be designed and built.

64 PWM to Phase Shifter (Design Tip)

Sure, the function can be carried out by a few flip flops and gates.

64 Improving the Pick-Up Angle of an Infrared Satellite Receiver Remote (Design Tip)

Just add a second infrared detector, they said, but how?

65 Audio Amplifier in Dinner Format (Design Tip)

The TDA7050 has great potential for making a very compact audio amp.

66 Creating a Low-Leakage Rectifier

Using a Self-powered Op Amp (Touchstone Semiconductor)

67 News & New Products

A monthly roundup of all the latest electronics products and components.



● Tech the Future

70 The Internet of Things (IoT)

Three experts in electronics industry and research discuss the origins of IoT and its impact on today's electronics.
Series Editor: Tessel Renzenbrink.

● Magazine

74 Retronics: The Nagra IV Tape Recorder

Once the most celebrated portable tape recorder in the world the Nagra was a professional device, developed for professional transportable applications; for broadcast use or picture-synchronized audio recording; for movies and TV.
Series Editor: Jan Buiting

78 Hexadoku

Elektor's monthly puzzle with an electronics touch.

80 Next Month in Elektor

A sneak preview of articles on the Elektor publication schedule.

Volume 38, Number 432, December 2012
ISSN 1757-0875

Publishers:

Elektor International Media,
78 York Street, London W1H 1DP,
United Kingdom.
Tel. +44 (0)20 7692 8344
www.elektor.com

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

Elektor is published 10 times a year with a double issue
for January & February and July & August.

Subscriptions:

Elektor International Media,
78 York Street, London W1H 1DP,
United Kingdom.
Tel. +44 (0)20 7692 8344 ,
Internet: www.elektor.com/subs
Email: subscriptions@elektor.com

Head Office:

Elektor International Media b.v.
P.O. Box 11 NL-6114-ZG Susteren
The Netherlands.
Telephone: +31 (0)46 4389444,
Fax: (+31) 46 4370161

Distribution:

Seymour, 2 East Poultry Street,
London EC1A, England.
Telephone: +44 (0)20 7429 4073

UK Advertising:

Elektor International Media b.v.
P.O. Box 11 NL-6114-ZG Susteren
The Netherlands.
Telephone: +31 (0)46 43 89 444,
Fax: +31 (0)46 43 70 161
Email: j.dijk@elektor.com
Internet: www.elektor.com
Advertising rates and terms available on
request.

Copyright Notice

The circuits described in this magazine are for domestic
use only. All drawings, photographs, printed circuit board
layouts, programmed integrated circuits, disks, CD-ROMs,
software carriers and article texts published in our books
and magazines (other than third-party advertisements)
are copyright Elektor International Media b.v. and may
not be reproduced or transmitted in any form or by any
means, including photocopying, scanning and recording,
in whole or in part without prior written permission
from the Publisher. Such written permission must also
be obtained before any part of this publication is stored
in a retrieval system of any nature. Patent protection
may exist in respect of circuits, devices, components etc.
described in this magazine. The Publisher does not accept
responsibility for failing to identify such patent(s) or other
protection. The submission of designs or articles implies
permission to the Publisher to alter the text and design,
and to use the contents in other Elektor International
Media publications and activities. The Publishers cannot
guarantee to return any material submitted to them.

Disclaimer

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

© Elektor International Media b.v. 2012
Printed in the Netherlands

All change — watch the colored dots

This edition of Elektor UK .Magazine
(*dot magazine*) marks the start of
a freshly structured publication.

The changes are not just cosmetic
as is obvious from the categories
called .Community, .Projects,
.Labs, .Industry, and .Magazine you
will have noticed on the restyled
contents pages, but actually
support our vision that the way
electronics is experienced on and
from paper does have a future, and
is online's best friend in many ways.

Our new membership models introduced last month and progressively
implemented in the course of next year enable you to enjoy these, our
printed pages as much as your PC screen showing the boiler room called
elektor-labs.com. The publication model is simple, too: on the website, feel
free to interact with your fellow readers on all design aspects of projects.
Next, if the Elektor editors and lab workers think it's really good, the project
and its OP are up for *elektorizing*, which equates to casting into a prestigious
article printed on paper.

In good e-engineering tradition the machine does not stop there as the
published project hopefully gets directed back to the .labs website, resulting
in feedback, adaptations, discussions, improvements, threats & threads,
FAQs, compile errors, tweaks, and unheard of applications devised by
members from any of the 50 countries and 6 language areas served by
Elektor. That's the way I'd like to see it work and happen, mainly because I
am into electronic engineering deep enough to say that the stuff that really
matters comes from engineers like you.



Jan Buiting, Managing Editor

The Team

Managing Editor:

Jan Buiting (editor@elektor.com)

International Editorial Staff:

Harry Baggen, Thijs Beckers, Eduardo Corral, Wisse
Hettinga, Denis Meyer, Jens Nickel, Clemens Valens

Design staff:

Thijs Beckers, Ton Giesberts, Luc Lemmens,
Raymond Vermeulen, Jan Visser

Membership Manager:

Raoul Morreau

Graphic Design & Prepress:

Giel Dols, Mart Schroijen

Online Manager:

Daniëlle Mertens

Managing Director:

Don Akkermans



United Kingdom
Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com



USA
Hugo Vanhaecke
+1 860-875-2199
h.vanhaecke@elektor.com



Germany
Ferdinand te Walvaart
+31 46 4389417
f.tewalvaart@elektor.de



France
Denis Meyer
+31 46 4389435
d.meyer@elektor.fr



Netherlands
Harry Baggen
+31 46 4389429
h.baggen@elektor.nl



Spain
Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es



Italy
Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it



Sweden
Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com



Brazil
João Martins
+55 11 4195 0363
joao.martins@editorialbolina.com



Portugal
João Martins
+351 21413-1600
joao.martins@editorialbolina.com



India
Sunil D. Malekar
+91 9833168815
ts@elektor.in



Russia
Nataliya Melnikova
+7 (965) 395 33 36
Elektor.Russia@gmail.com



Turkey
Zeynep Köksal
+90 532 277 48 26
zkoks@beti.com.tr



South Africa
Johan Dijk
+27 78 2330 694
j.dijk@elektor.com



China
Cees Baay
+86 21 6445 2811
CeesBaay@gmail.com

Our network



VOICE COIL

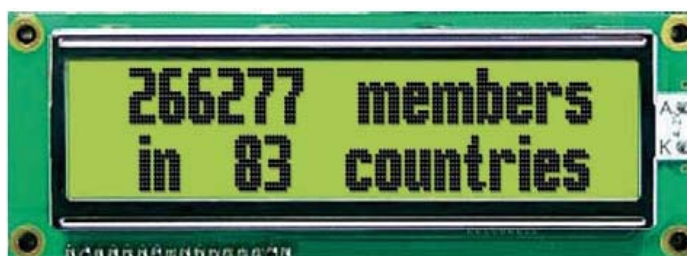
CIRCUIT CELLAR
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION



audio X PRESS



Connects you to



Supporting Companies



Beta Layout
www.pcb-pool.com39



DesignSpark
www.designspark.com63



EmbCode
www.embcode.com10



Eurocircuits
www.elektorpcbservice.com69



EzPCB
www.ezpcb.com73



Jackaltac
www.jackaltac.com73



Labcenter
www.labcenter.com84



Microchip
www.microchip.com2



MikroElektronika
www.mikroe.com3



National Instruments
www.ni.com11



Pico Technology
www.usbmso.com/PS17061



Reichelt
www.reichelt.co.uk15



Schaeffer AG
www.schaeffer-ag.de39

Not a supporting company yet?

Contact Johan Dijk (j.dijk@elektor.com, +27 78 2330 694)
to reserve your own space for the next edition of our members' magazine

Elektor World

By Jan Buiting, Managing Editor

Elektor Improved Radiation Meter on ConSat-1... soon

Hello. I am Michelle Boyce, affiliated with SpaceConcordia.Ca. We have recently constructed a Cube Satellite (ConSat-1) for the Canadian Satellite Design Challenge.

Onboard, one of the payloads is a Solid State Radiation Detector based on articles from your magazine:

- Measure Gamma Rays with a Photodiode Radiation Detector using a BPW34, by Burkhard Kainka, June 2011.
- Improved Radiation Meter Counter for Alpha, Beta and Gamma Radiation, by Burkhard Kainka, November 2011.

The project was part of an Educational Outreach Program with Space Concordia.

Kind regards, Michelle Boyce (Canada)

Our Radiation Meter project was a triumph from day #1. On reading this message from Canada, Elektor PR staffers performed a global search for the wording 'global coverage' in all corporate documents and replaced it by "global coverage *and beyond*". Meanwhile Michelle has been requested to write an article for Elektor.

Elektor Improved Radiation Meter @ Oxford University... soon 2

Hello Thijs, I work at the University of Oxford Physics department. We have been building a circuit similar to yours to go into a helium balloon to measure cosmic rays. We bought the radiation detector circuit (we have tried a few different kits) with hopes to test it in our lab. We have access to many different sources but have not got your kit working. I think I have built and shielded it correctly, but can't work out what's wrong with it. It currently triggers with the comparator set to L68 (above doesn't trigger). The signal on K4 looks to be about 500 mV_{pp} on a scope.

We are quite short of time now but I would still personally like to see this kit working. If you would like us to test this circuit with our sources it possible you have a working circuit we could bor-

row to test or could we send the one I have built back (it's in a metal box)?

Jeff Lidgard (UK)

What, YOU and 'not working'? A working prototype is on its way to you Jeff.

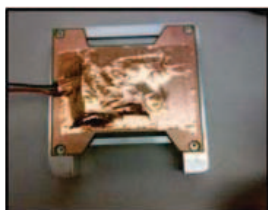
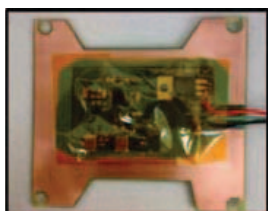
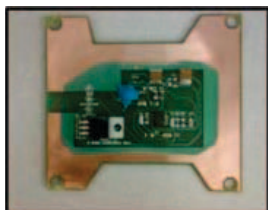
Elektor Monster Formant hits YouTube

Siegfried Brückner is likely to have the world's largest constellation of Elektor Formant modules on the planet and in space. We estimate the instrument is about 10 feet wide and 4 feet tall. It is linked to a sequencer, a bass pedal and a keyboard. A total of about 110 Formant modules, 500 controls, and dozens red patch cords produce that inimitable Formant synthesizer sound [1].



Siegfried wrote to us: "Phase #3 of the project comprises over 160 modules from Elektor's Formant books 1 & 2, as well as specials and discrete circuits. Everything is connected through a module wall. An 8-way MIDI interface was incorporated, making the Formant controllable over MIDI and CV/Gate.

The new Formant keyboard now has a 4th and 5th octave keyboard and additionally a 13-key pedal. Each keyboard / pedal has a C/V interface.



Rommedal and puzzle creator Sadettin Commert.

Hi friends at Elektor,
I am not sad or mad about not winning, but totally lost about the answer given as

Input 2. It is possible, but with my little brain knowledge I find that Input 1 also is a possible solution?

This part is taken from Wikipedia, and I hope we can agree on the fact that a Zener is a normal diode in the arrows direction :-)?

The picture shows a hand drawing of my Input 1 solution. Would it be possible for you to fast draw like mine the Input 2 solution and verify that I'm right or not, and if not, please show me where is "sleep".

Sincerely, Klavs Rommedahl (Denmark)

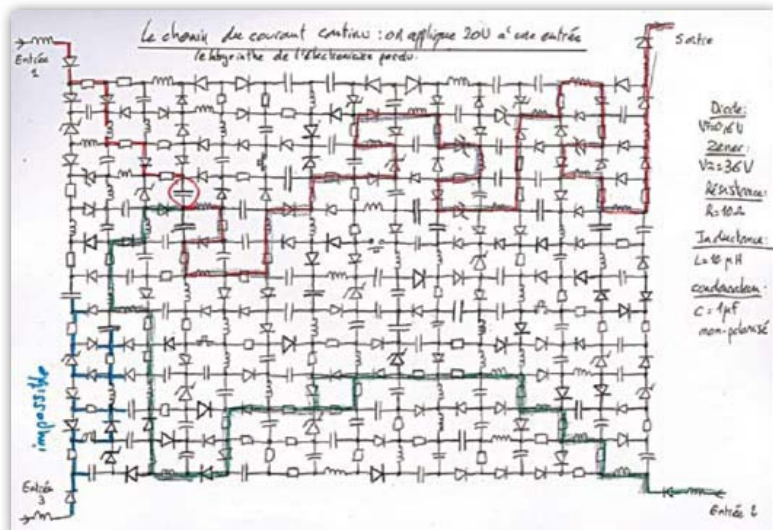
(translated from the French)



Dear Jan and Klavs

As it turned out the printed version of the maze was an early one which contains an error, see the attached pdf showing the correct version. One resistor should have been a capacitor! The erroneous drawing thus allows two possible solutions.

KR, Sadettin Commert (France)



To be able to process other sounds on the Formant, there additional MIDI interfaces were built in. These allow the keyboards to be played mono as well as polyphone fashion (expander). Also, the expander sounds can be modified as required. In April 2012 I was able for the first time to demonstrate my Monster Formant to fellow specialists at the Happy Knobbing event in Rheinland Pfalz, Germany, the theme of the year being: sound generation using a modular system [2]. The sound recording unfortunately is below standard. My next performance is in Munich on November 17." Elektor's own Formant was rescued from a dumpster and gets played occasionally in Elektor Cellar [3]. Headphones on!

[1] http://youtu.be/_5cRr7og4Vc

[2] <http://youtu.be/ynQyGT6t9tU>

[3] <http://youtu.be/JOhxLE3-eCI>

Lost in the Electronic Maze

To create a puzzle that has a single solution is as hard as solving it. See page 78 in the July & August 2012 edition. Over to reader Klavs

 Circuit Cellar 25th (19_{hex}) Anniversary Party



September 28 and 29 2012 marked the twenty-fifth anniversary of Elektor's truly awesome sister magazine *Circuit Cellar*. Your editor was fortunate to be invited over to New Castle NH, USA for din-

ner, speeches and chats not just with current and former CC staffers, but also with many of their renowned columnists and project editors including Ken Davidson, Dave Tweed, Robert Lacoste, Ed Nisley, and Jeff Bachiochi. With CEOs, SFOs etc. well out of hearing range the conversations soon turned electronic & embedded. Sincere thanks go out to party hosts Steve Ciarcia and Hugo Van haecke.

 RF Poetry

My thanks go out to the crew at the Robert Frost Farm, Derry, NH, for awarding me a pencil for no more than citing one line of RF poetry: "One could do worse than being a swinger of birches". Back at my desk at Elektor HQ I am using the pencil to scribble edits as well as draw the odd circuit diagram when our main servers are down.

Advertisement

ROBUST SOFTWARE COMPONENTS FOR YOUR EMBEDDED SYSTEM!

FREE SD DRIVER!

PRODUCTS



FREE DOWNLOAD

Store massive amounts of data on SD memory cards!

- Card support: MMC / SD / SDHC / SDXC
- Interface support: Bit-bang / SPI / 1-bit / 4-bit

Free download available!



Store files and folders in flash, SD cards and USB sticks!

- FAT12/16/32
- Long file names
- Multiple open files
- Multiple threads
- Partitions
- TRIM
- Multiple drives
- Cache



Skip the serial port and use high speed USB!

- Embedded USB drivers: COM port, Stream and/or Packets
- Windows USB drivers: Signed binaries and full source
- Composite support

Embedded software as **clear, readable, platform independent C source**

Full documentation available online (no login required)

Free **1 hour consultation** with every purchase and **60 days trial** period

EmbCode.com

+46 31 780 4421 / info@embcode.com
www.embcode.com

WITH WORKING, RELIABLE SOFTWARE COMPONENTS YOU CAN FOCUS ON YOUR APPLICATION!

The Trusted Leader for Any Measurement



NI data acquisition products are the most trusted computer-based measurement devices available, offering superior accuracy and maximum performance. NI CompactDAQ systems feature more than 50 measurement-specific modules and complete breadth and depth of I/O. Coupled with the analysis and signal processing capabilities of NI LabVIEW software, this platform puts the power of graphical system design at your command so you can build any measurement system faster.

LabVIEW offers powerful analysis functions for sensor and signal data, simplifies GUI development and helps you program the way you think—graphically.



>> Accelerate your productivity at ni.com/measurements-platform

01635 517300 | uk.ni.com

See what you missed at **NIDays 2012**: download presentations, view videos & pre-register for 2013 at uk.ni.com/nidays

Follow us on



Search **niukie**



By Clemens Valens
(Elektor .Labs)

Don't miss the hype

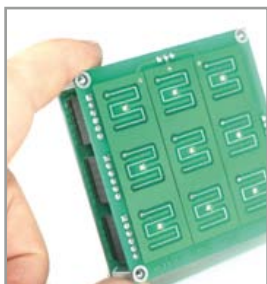
When I was a student the teachers often said that it was hard to keep up with advancing technology. I thought "well, is it?" as I did not find it difficult at all to consume all the information I could find. But now that I am about to lose the last hair on my head and my muscles are getting weak, I must admit that it is not an easy task to constantly know what is going on in every corner of techno land. Luckily I am not alone and some people are even worse off than I.

The other day I shared a ride with an even older colleague. Since he was driving I was watching the road signs and when we got close to our destination I pulled out my phone to have a look at Google Maps. The phone was sleeping so I had to activate it first and for this I have to slide my finger over the display in a special pattern, like a signature. This is called a pattern lock. My colleague saw me do this and was so amazed that he almost crashed into another car. Back on the road again he asked me if what I just did was the latest in cool gadgets. I said that it wasn't and that I had seen people using it for several years now. Actually, I said, we have a project on the .LABS website that shows you how to build your own pattern lock. My driver was deeply impressed.

Funny though, just three days later I received a message from the guy that was doing this project on .LABS, announcing that it was ready. He had even posted a promotional video, did I get a chance to see it? I hadn't been to his project page in a while, so I looked it up again and now it was time for me to be deeply impressed. Nice clean write-up, high-quality photographs, excellent video, exactly as we like it. His presentation immediately made me want to build one myself. You will not be surprised to read more about this project soon in the printed magazine. Of course it would be great if all the projects on .LABS were so well presented, but it is not at all mandatory. If you don't have the equipment to take studio-quality photos, then use your phone. If you don't have the time to draw a circuit diagram in a professional CAD package, simply sketch it on a piece of paper and upload a scan. The only thing we care about is that you take the time to come up with neat projects and great ideas. Blow us over and we will reward you.

Have a look at the pattern lock project and build your own:

www.elektor-labs.com/project/android-style-capacitive-sensing-pattern-lock.12407.html



The origin of a species

I just love Wikipedia, don't you? Look what I discovered some time ago: *"Psocoptera are an order of insects that are commonly known as booklice, barklice or barkflies. They first appeared in the Permian period, 295–248 million years ago. They are often regarded as the most primitive of the hemipteroids. Many of these species have only been described in recent years.*

The largest suborder are the Psocomorpha, with about 3,600 species in 24 families, ranging from the species-poor Bryopsocidae (2 species) to the speciose Psocidae (about 900 species). Psocomorpha are notable for having antennae with 13 segments. They have two- or three-segmented tarsi, this condition being constant (e.g. Psocidae) or variable (e.g. Pseudocae-cilliidae) within families."

If, like me, you didn't understand most of that, rest assured that it's about bugs. So why mention them here? Because recently the suborder of Psocomorpha has been extended with a new species, the noisiest of all, the PSo-CaMorph music synthesizer.



It was first discovered on .LABS, and I think that this is very cool. The goal of the project is to design an old-fashioned analog music

synthesizer using modern Programmable System-on-Chip (PSoC) technology. As you have surely understood from the first bit of this article, bug families are highly modular; the species are constructed from wings, leg and antenna segments and other body parts. Analog music synthesizers are also modular; they consist of one or more VCO, VCF and VCA modules and some other parts. Insects can be collected and stored in special envelopes; analog synthesizers have envelope generators. Chips, like bugs, are small and annoying but they have more legs. Psocoptera appeared some 250 million years ago, analog synths are vintage too. So a modular synthesizer in a chip is almost a bug, right? Anyway, what I wanted to tell you is that U2 can now get into insectology by joining the PSoCaMorph project on .LABS. If you do so, you may even win a PSoC 5 Development Kit, all you have to do is contribute.



The PSoCaMorph lives here:

www.elektor-projects.com/project/the-psocamorph.12269.html



Editor's Choice

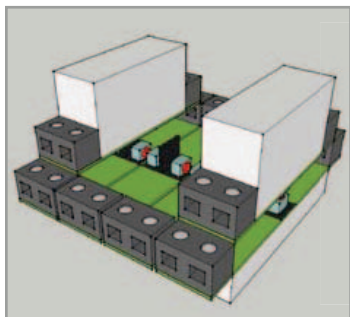
**EDITOR'S
CHOICE**

A number of .LABS projects have been selected by our editors and should be up for publishing in the near future. For some of these projects, sadly we found that the original poster (OP) does not reply to our messages. Therefore, if you posted a project, please check on a regular basis the email account you used to access .LABS. We will not get you in publication if we cannot get in contact with you. Here is a selection of projects that we think are interesting and that we would like to publish in the printed magazine:

Geiger Counter Data Logger with WLAN Interface

OP Marcus Vohburger is working on a complete system for measuring and monitoring radioactivity. According to him, building a Geiger counter is actually pretty simple. Because it is so simple he added some extra stuff.

www.elektor-projects.com/project/geiger-counter-data-logger-with-wlan-interface.12264.html



Mains Gate:

Programmable Relay & Energy Monitor

The OP John Hind writes about this project: rather than some US political scandal, this is an idea for "mashing up" five of the circuits from the July & August 2012 double issue to produce a really flexible "mains gatekeeper" circuit combining a multitude of control, measurement, protection and logging functions at minimal cost.

www.elektor-projects.com/project/mains-gate-programmable-relay-energy-monitor.12461.html

Simple MIDI Wind Instrument

This is based on early experiments done by the OP that ended in an overly complex system. If we publish this project it would be a modernized and simplified version. We could do with a bit of help from you to come up with a usable wind instrument slash air pressure sensor.

www.elektor-projects.com/project/simple-midi-wind-instrument.12277.html



Wind Direction and Speed Instrument

Sandwiched between two spray-can caps sits a little PCB with only four parts: a purposely overheating transistor and three NTC resistors. Wind will blow the column of hot air produced by the transistor out of the centre; this change of position is picked up by the NTC resistors. Neat, don't you think?

www.elektor-projects.com/project/wind-direction-and-speed-instrument.12480.html

Component Tips

By Raymond Vermeulen (Elektor Labs)

MIC502 and MAX6643 Fan-Speed Controllers

For this month's instalment I researched two ICs that provide stand-alone functionality as fan-speed controllers. This can be very handy when you have a system which is dissipating a lot of power, but is also built into some kind of enclosure. This makes forced-cooling with a fan a necessity. If you would like to make the control of this fan dependent on the temperature, but don't want to delegate this task to the existing microcontroller, then a separate control IC will be very useful. You only need to connect a sensor and a fan. One striking detail is that the ICs presented here do not follow the standard 4-pin, 25 kHz PWM fan speed control, but switch the power supply voltage directly at much lower frequencies.

(120569)

MIC502

The MIC502 made by Micrel uses an NTC or PTC as the temperature sensor, with the option of having a second sensor. The sensors are connected to inputs VT1 and VT2. A voltage from about 30% to 70% of V_{dd} produces a duty cycle of 0% to 100%. The highest of the two inputs takes precedence, which is a very nice feature if you use one sensor to measure the ambient temperature and use another to monitor the temperature of a component that is likely to get hot. A voltage can be applied to the V_{slp} pin, which causes the chip to go into a sleep state when both inputs VT1 and VT2 drop below this value. When either VT1 or VT2 go above this value then the IC will be reactivated. This is mainly to avoid the fan stalling at a duty cycle that is too low for it to operate properly. A timing capacitor is connected to pin CF, a value of 100 nF is recommended for a frequency of 30 Hz. You can, however, also set a higher frequency. According to the data sheet the range is from 15 to 90 Hz.

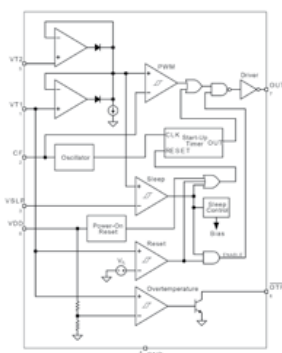


Figure 1. Block diagram of the MIC502.

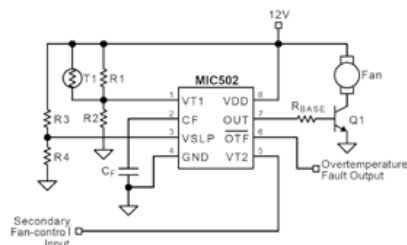


Figure 2. Application example using the MIC502.

MIC502 datasheet: www.micrel.com/_PDF/mic502.pdf

MAX6643

The MAX6643 uses a diode-connected transistor to do the temperature measurement. This IC has three different temperature control settings. The temperature is set (between 60 and 100 °C) with the 'Overtemperature Threshold' inputs (pins OT1 and OT2) and the OT-output will be activated when this temperature is exceeded. The 'High temperature Threshold' inputs (pins TH1 and TH2) determine at which temperature the duty-cycle of the PWM-signal increments by one step. The 'Low temperature Threshold' inputs (pins TL1 and TL2) determine the temperature at which the duty cycle of the PWM signal goes down by one step. For each of these temperature-thresholds a number of different values can be set by connecting the corresponding pins to V_{dd} or GND or leave open (refer datasheet).

There is also a 'FULLSPD' pin, to force a duty cycle of 100%. This can be useful when an excessive temperature has been detected. In addition there is an input for the tachometer signal from the fan, which allows the 'FANFAIL' pin to indicate when there is a problem with the fan.

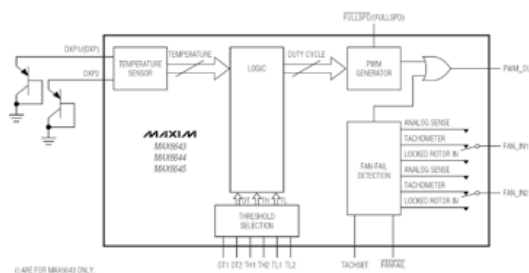


Figure 3. Block diagram of the MAX6643.

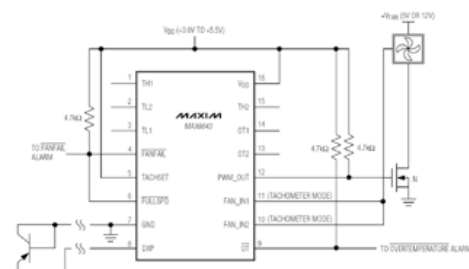
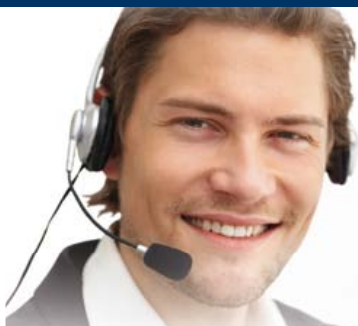


Figure 4. Application example using the MAX6643.

MAX6643 datasheet: <http://datasheets.maximintegrated.com/en/ds/MAX6643-MAX6645.pdf>



- ✓ more than 40 years experience
- ✓ more than 40000 products in stock
- ✓ no minimum order charge
- ✓ fast 24 hour shipment



+49 (0)4422 955-333



+49 (0)4422 955-360

**Professional quality
@ discount prices!**

Housing Technology

Measuring Technology



System Housing

- Suitable for Eurosize printed circuit boards
- 3 height units.

GEH SG 1-10 **46.65** (~ £37,60)

pro Ma

10" opening: 203.8 mm
19" opening: 427.8 mm

GEH SG 1-19 **66.50**

Hammond Series 1550W, IP66

- Aluminium die casting housing in rugged design
- Self-adhesive foam sealing, 4 stainless steel screws



	L x W x H (mm)
1550WQ	4.10 60 x 55 x 30
1550WA	4.10 89 x 35 x 30
1550WB	5.40 115 x 64 x 30
1550WD	6.10 115 x 64 x 55
1550WC	6.95 115 x 90 x 55
1550WE	10.65 171 x 121 x 51

High-quality etui housing

- Dust-proof
- Jet-proof



	L x W x H (mm)
BOPLA ET-205	4.80 52 x 50 x 37
BOPLA ET-206	7.15 65 x 50 x 37
BOPLA ET-208	8.95 98 x 64 x 38



More choice: <http://rch.it/6F>

Small Plastic Casing

- Standard casing
- 6 fastening nipples for circuit boards

	L x W x H (mm)
GEH KS 21	0.98 72 x 50 x 21
GEH KS 28	0.98 72 x 50 x 28
GEH KS 35	1.10 72 x 50 x 35
GEH KS 42	1.10 72 x 50 x 42



Connectors

Power Supply



NEUTRIK NE-8FDP

9.65

(~ £7,78)



Flange Bushing

- Ethernet
- RJ45 on RJ45

Standard RCA plug

- With kink protection

CSP RT	red
CSP GE	yellow
CSP WS	white

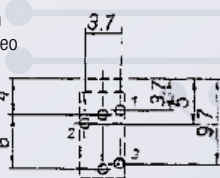
each 0.09
(~ £0,07)



All RCA plugs <http://rch.it/6G>

High-quality Jack socket

- 2.5mm
- Stereo



LUM 1501-03 **0.69**

Lumberg Product range <http://rch.it/6H>

Professional lockable Sockets

- 6.35 mm
- Mono/stereo

NJ-3FPB



NEUTRIK NJ-3FP **5.50** Metal casing
NEUTRIK NJ-3FPB **5.95** black chrome-plated housing

Network & PC Technology

Development Tools

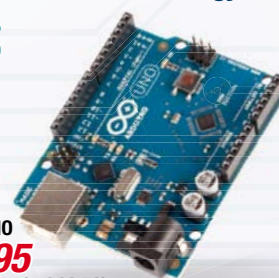
Sat and TV Technology



ARDUINO UNO

27.95

(~ £22,52)



Microcontroller Board

Open Source Prototype Platform based on Atmel ATmega328

- 14 digital I/O interfaces, 6 analogue inputs
- 16MHz quartz oscillator, ICSP header, reset



Microcontroller Boards

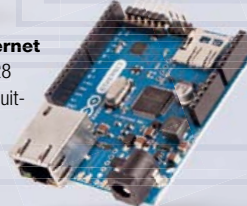
- Based on ATmega 2560
- 54 digital I/O pins
- 16 analogue inputs
- 16MHz crystal oscillator
- 4 UARTs, ICSP header, reset



ARDUINO MEGA

49.50

(~ £39,89)



WizNet W5100 Ethernet

- Based on ATmega 328
- 13 digital I/O pins (4 suitable for PWM output)

ARDUINO ETHERNET

49.50

(~ £39,89)

Professional Raspberry Pi Housing

- Convenient design – perfect assembly
- Intuitive cable connection
- Status display
- 4 housing feet
- Circuit board size: up to 85 x 56mm



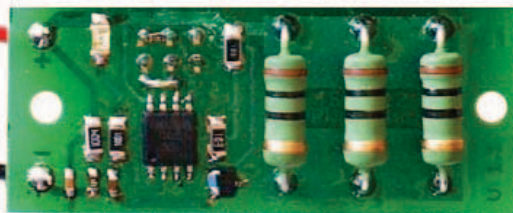
TEK-BERRY

4.95

(~ £3,99)

Circuit board not included in scope of delivery

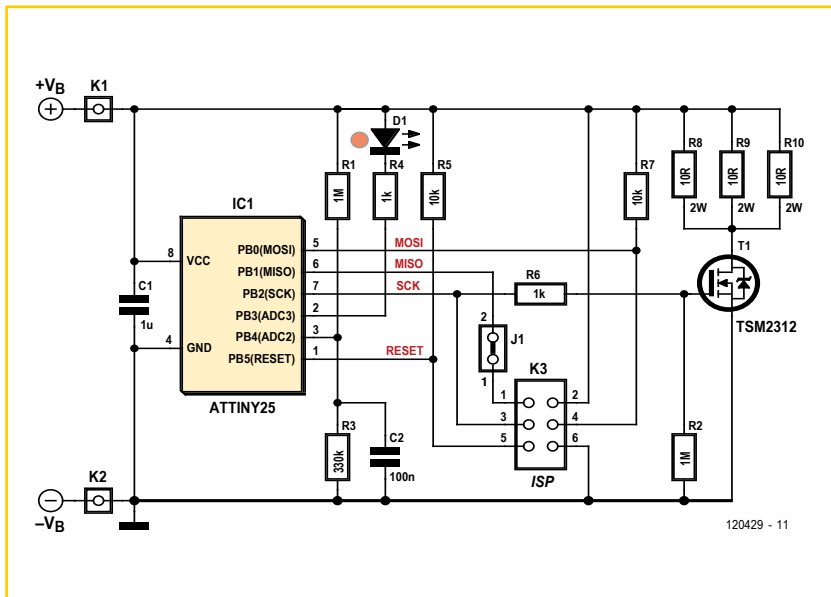
Tiny but not insignificant!



by Dr. Thomas Scherer (Germany)

This circuit refuses to be programmed by an AVRISP MKII programmer; can anyone see why? I must admit to spending too many hours recently trying to answer that simple question. First some background: My plan was to make a simple, flexible but precision balancing circuit for my newly acquired LiFePo battery pack. Off-the-shelf solutions using expensive specialist ICs would do the job but I like to take a more practical approach. This seems like an ideal application for a tiny low-cost AVR microcontroller. I am already familiar with the ATtiny25 (see [1] and [2]) so I was feeling quite confident as the design took shape. Little did I suspect that I would fall at the first hurdle when I tried to flash the firmware to the controller.

My first thoughts turned to the layout but after checking and double-checking the PCB tracks



and component placement the finger of suspicion began to point to the microcontroller. Maybe the manufacturer had sent out a batch with the SPIEN fuse bit disabled? That would of course prevent any attempt to flash using the ISP/SPI route. My use of an SMD outline microcontroller made it difficult to directly connect to a STK500 programming board to work round the serial interface and try 'high voltage' programming. Once again, with leads directly soldered from the HV pins on the STK500 to the PCB programming failed even though HV programming should always be possible. The plot was definitely starting to thicken; to simplify things even more I soldered flying leads onto the SMD ATtiny25 package to directly connect it to a DIL socket on the STK500. This time programming was successful and the SPIEN fuse on the chip was found to be correctly set all along. There must indeed be something wrong with the circuit, but what?

Perhaps the loading of the 1 kΩ resistor R4 in series with LED D1 was affecting a programming signal? But no, that was also a red herring; pin 2 (PortB.3) doesn't even have a connection with K3. Now my attention focused on the switching MOSFET T1, although small in stature, this could in fact have a significant amount of capacitance associated with its input; maybe even enough to load the digital programming signal? A quick look at the data sheet confirmed my suspicions: $C_{iss} = 900 \text{ pF}$... bingo!

To reduce its influence, resistor value (R6) was increased to 10 k Ω . Now when I plugged the AVRISP cable into K3 and clicked on 'Program' in AVR studio everything ran as it should and the flash was successful. I am really quite happy to admit to my errors; after all as someone once said, experience is just the name we give to our mistakes.

(120429)

- [1] www.elektor.com/magazines/2011/january/all-soft-555.1642210.lynkx

- [2] www.elektor.com/magazines/2008/january/anti-standby-switch.321677.lynx

Missing ground plane!?

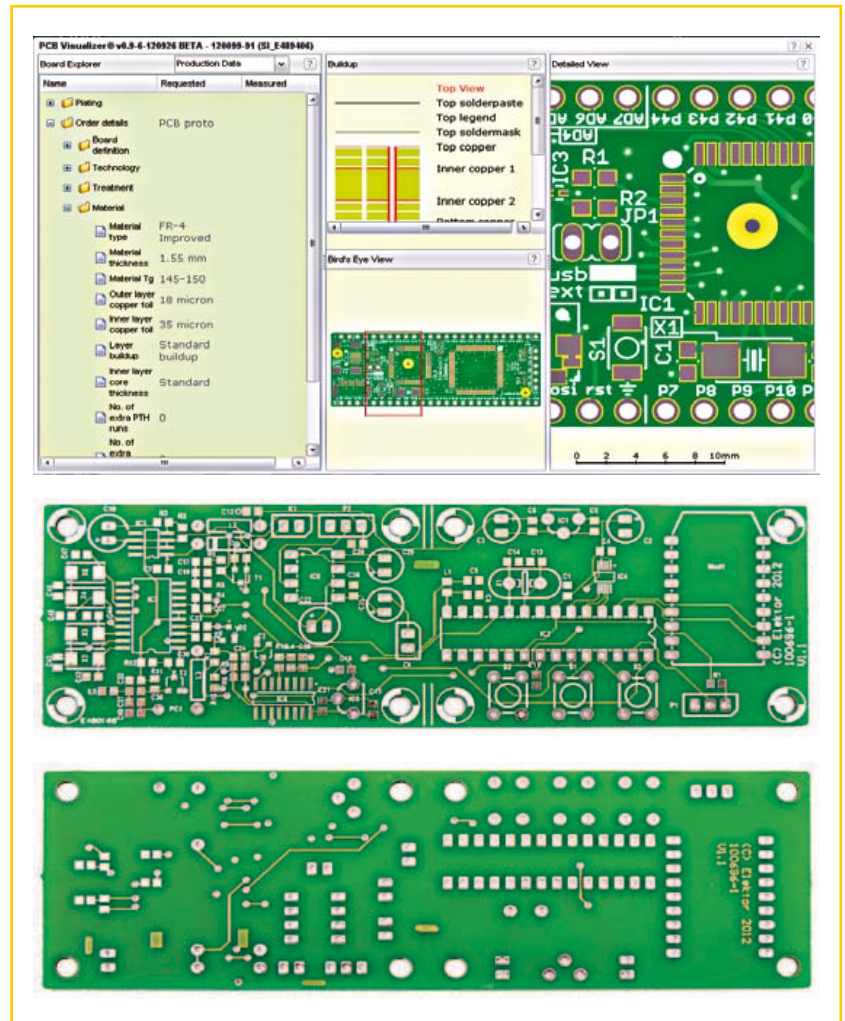
By Thijs Beckers (Elektor Editorial & Labs)

Not everything went smoothly as lab worker Luc was working on an aviation radio scanner project. Several components used by the author were no longer in production, albeit they would probably still be available from selected retailers and old stock vendors. So to ensure the project can be replicated by readers for some time in the future, a number of components had to be replaced with modern ones. Laying out the PCB (with the new components) and ordering all the components needed for the prototype build-up, however, went without a hitch. Until the proto PCB arrived in the post.

Check your board with PCB Visualizer!

Somewhere during the design process using our CAD software suite Altium Designer, Luc 'shelved' (hid) the top and bottom ground planes in order to have a better overview on the routed signals. When finished, he generated the Gerber files for the proto PCB production using our standard .out-job file and sent them off to our long term proto supplier Eurocircuits [1]. No errors were flagged during the process and it wasn't until Luc received the PCBs in the mail that he saw the shelved ground planes were missing on the physical board (see the pictures). This rendered the PCB useless, since a lot if not all of the ground connections were laid out using the ground planes.

Backtracking his actions, he remembered shelving those missing planes. As it turned out, everything that's shelved gets omitted from the Gerber files. Of course this could have been prevented by having a look at the files with a Gerber viewer program before they were submitted for production, but time pressure and distraction by other duties got into the way of this extra verification step. Having discovered the error, Eurocircuits' new service 'PCB Visualizer' (see screenshot) was mentioned as a possibly better checking option than the Gerber viewer software our labs had been using. This online utility accepts your uploaded



What's cooking?

Not Elektor's new SMPSU we hope!

Isolation transformer suitable for 115 and 230 VAC; this series uses an identical footprint for different power output ratings. Very convenient.

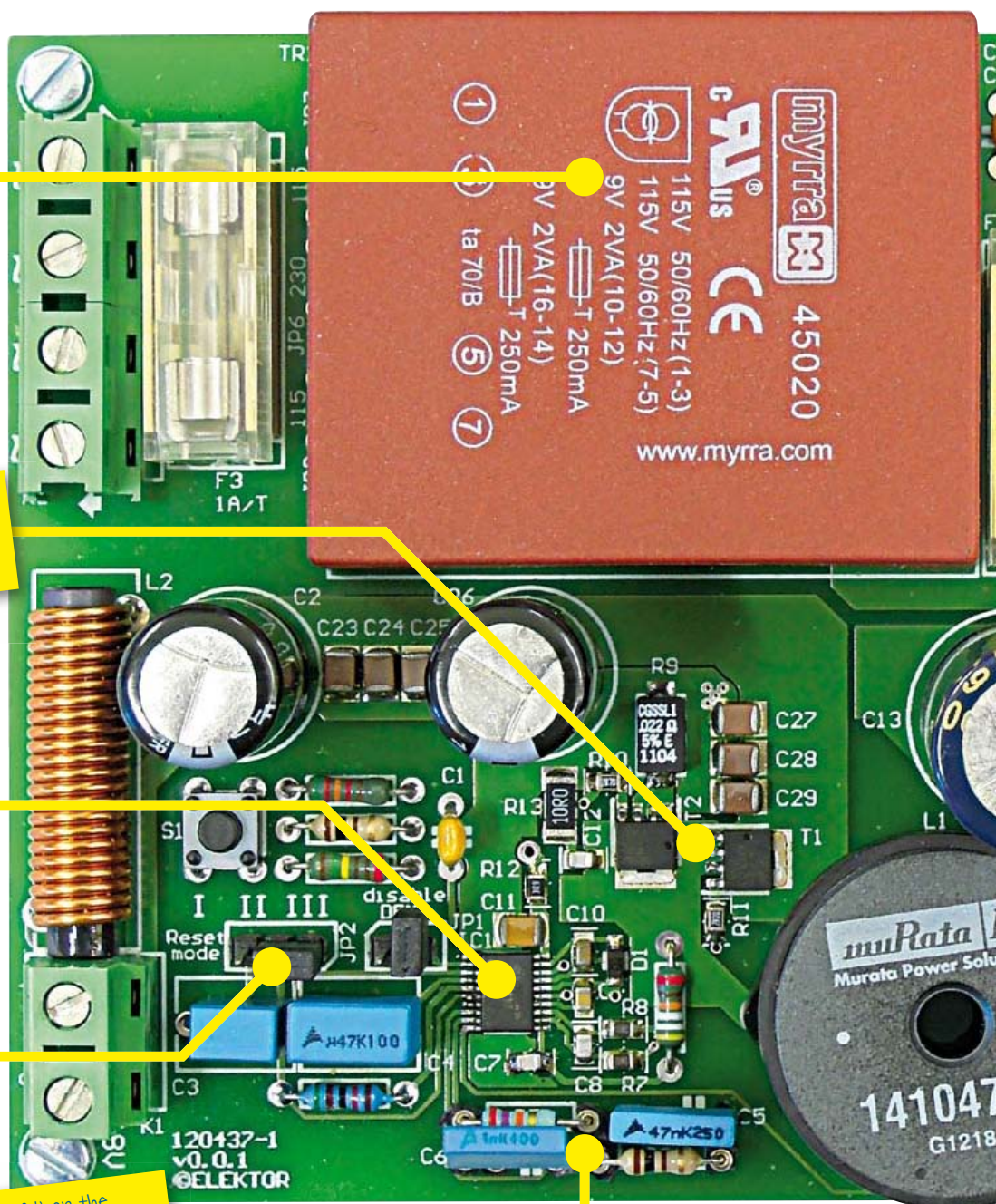
T1 and T2 are special high current FETs from NXP with extremely low $R_{DS(on)}$.

IC1 and IC3 barely solderable by hand. Still manageable, although the use of a hot air rework station is highly recommended. A reflow oven would be ideal for this job.

Reset mode is selectable with JP2 (hiccup, latch-off, cycle-by-cycle)

Note the 'v0.0.1' on the board.

Fine tuning of the feedback filter (C5, C6, R5, R28) is determined experimentally.



A new project is taking shape in our Elektor Labs. In the capable hands of designer Ton Giesberts a switch-mode power supply (SMPSU) is now being devised and has made it to its early proof of concept stages. Let's have a sneak peek...

By Thijs Beckers (Elektor Editorial & Labs)

Deep down the supply is a current mode Buck converter.

Dual layer PCB.

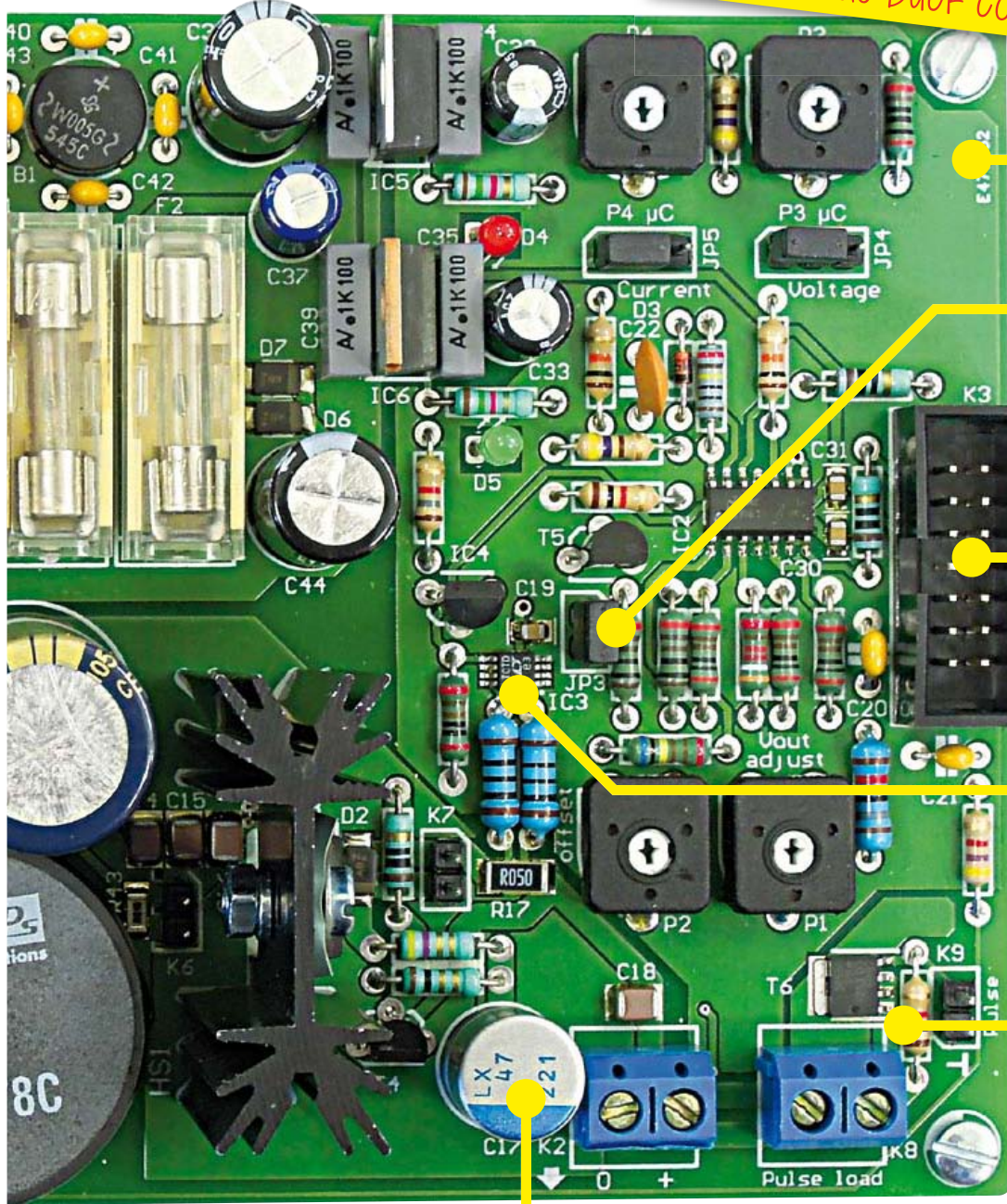
JP3 provides the option to choose between hardware and/or software current limiting.

Interface to microcontroller; software and design to be implemented.

IC3 has its NC pins short circuited by tracks. To be investigated.

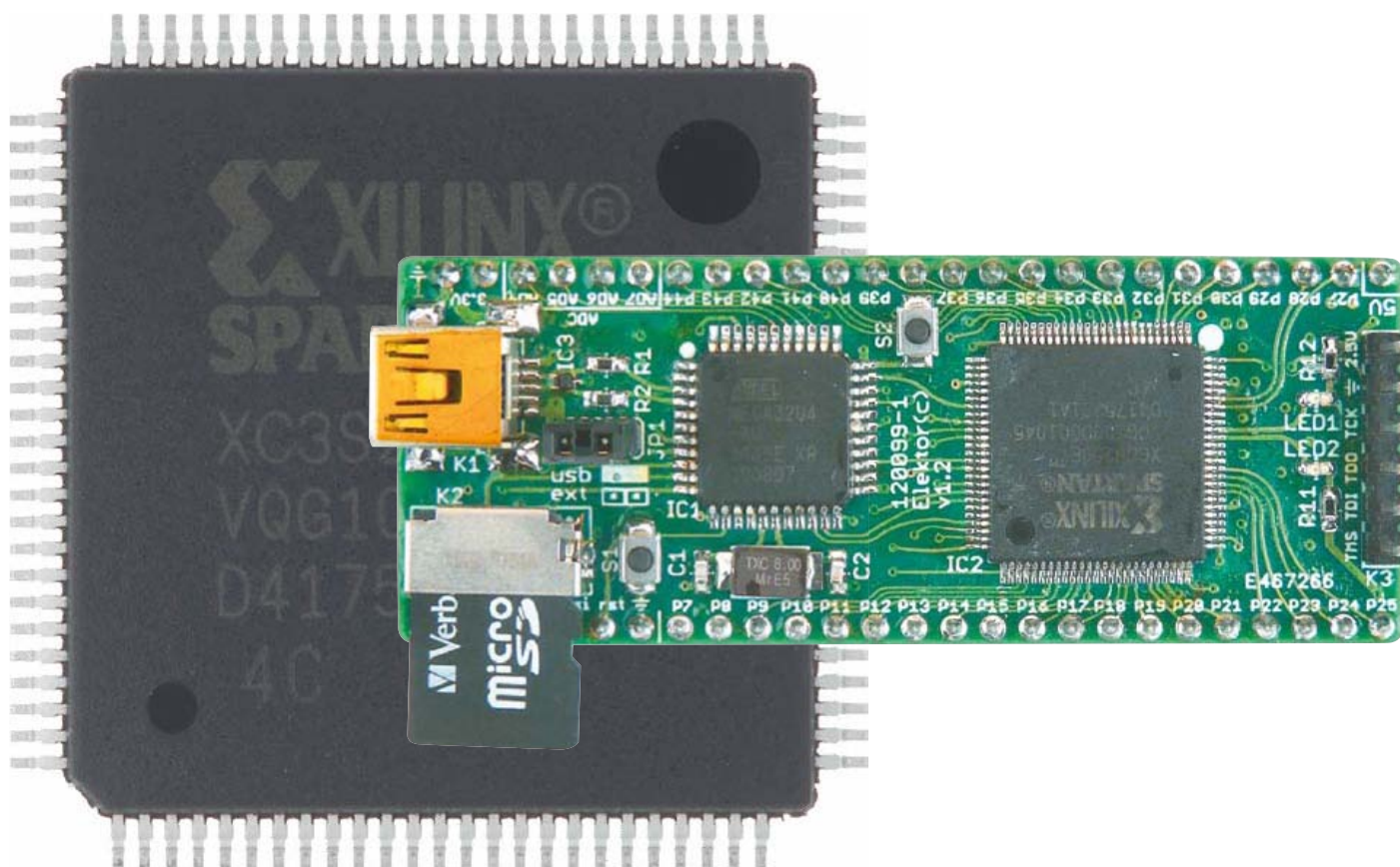
T6: section temporarily implemented for pulsed load testing purposes

C17 has specs by design, which are only met by a special Nichicon cap. Availability?! ESR measurement results on C17 were unsatisfactory (0.4xx ohms). Measurements conducted 'the old fashioned way' were inconclusive at first. It turned out to be a ground loop between the measurement devices messing up the measurements.



Taming the Beast

A simplified approach to working with FPGAs



Text:
Clemens Valens
(Elektor Labs)

Design:
Raymond Vermeulen
(Elektor Labs)

FPGAs are unquestionably among the most versatile but complex components in modern-day electronics. An FPGA contains a maze of gates and other circuit elements that can be used to put together your own digital circuit on a chip. In this series of articles we use an FPGA development board designed in the Elektor Labs to show how easy it is for any electronics enthusiast, whether professional or amateur, to work with these programmable logic devices. If you've been afraid to tackle FPGAs, you can relax now with the Elektor FPGA board.

The abbreviation 'FPGA' stands for 'field programmable gate array'. FPGA devices might better be described as user-configurable logic, since that's what it comes down to. An FPGA contains a large

number of logic gates, flip-flops and memory elements (as well as quite a few other things) that can be configured by the user to form a circuit. The main difference between FPGAs and micro-

processors or microcontrollers is that FPGAs do not execute programs (they are not processors), but instead should be regarded as circuitry.

It all started with a simple array

The first programmable logic devices consisted of arrays of a few dozen logic gates whose inputs and outputs could be linked together more or less as desired. This makes it relatively easy to implement complex logic functions in a single IC instead of using a large PCB with lots of individual logic ICs. Another benefit of this approach is that it allows functions to be modified quickly by reconfiguring the IC, with no need to cut PCB tracks and make new physical connections. This type of programmable logic closely resembles read-only memory (ROM) or erasable programmable read-only memory (EPROM), which offer similar capabilities. To take a simple example, consider an AND gate with two inputs (A and B) and one output (Q). The truth table of this gate is:

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

If you rename the A and B inputs to A1 and A0 and the Q output to D0, you have a data line and two address lines that can be used to access four memory locations. If you then store the values 0, 0, 0 and 1 in the memory locations, you have an AND function in memory. You can also store different values in the memory, and you can use a different number of addresses. This is a fairly simple way to implement a complex logic function. However, what you can't do in memory is to store the result of an operation somewhere else in memory. Even if the memory has many registers, it cannot access itself. Only the user can write data to the memory or read data from the memory.

Consequently, programmable logic was extended to include registers for storing results, so that they could be used somewhere else on the chip. What we mean by 'registers' here is ordinary flip-flops, which are one-bit memory cells. **Figure 1** shows a simplified version of the logic cells used in FPGAs.

You can do quite a bit with flip-flops, especially if you have a lot of them. For example, you can

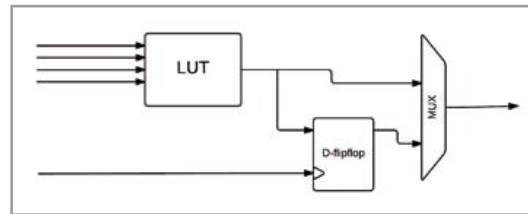


Figure 1.
A simple block diagram
representation of a logic cell
in an FPGA.

build counters and registers, such as 8-bit registers or registers of any desired width. You can also implement shift registers, which makes it possible to perform serial to parallel conversions (and the other way round) or delay digital signals. In short, an array of logic gates is nice, but it becomes truly attractive when you add memory elements.

Boosting the complexity with more options

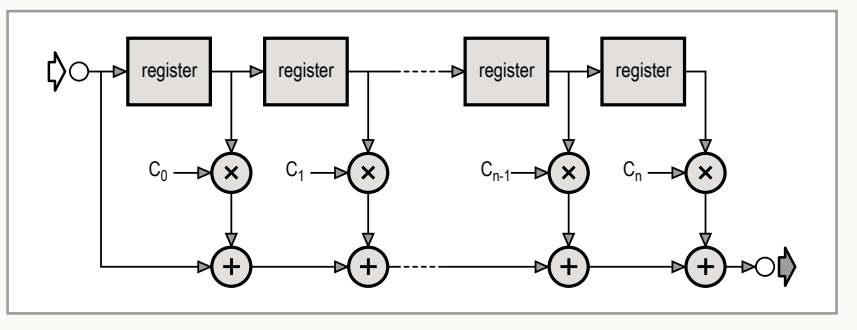
In the course of time programmable logic became increasingly complex, as can be seen from the designation 'complex programmable logic device' (CPLD). Technological progress also made it possible — or perhaps better put, necessary — to design the devices to more closely resemble random access memory (RAM) than read-only memory (ROM). RAM cannot retain data without power, and this property is shared by these programmable logic devices — every time the power is switched off, the device configuration is lost. It was therefore necessary to be able to configure the devices on start-up without using special programming hardware — 'in the field', as it were. This marked the birth of FPGAs. To simplify the configuration process, the configuration data is often stored in external nonvolatile memory and read out by the FPGA during start-up. This is highly reminiscent of a microprocessor system, which also reads its software from external memory. Incidentally, FPGAs that can retain their configuration are also available now.

Over the years FPGA manufacturers have managed to put an enormous number of gates and flip-flops on a single chip. This constantly increasing level of integration makes new applications possible, as well as extremely high data rates, but it also causes new problems in areas such as chip layout. The ability to arbitrarily link several million ports requires a lot of interconnects. Some of these interconnects can become relatively long, for example if the output of a gate in the upper left corner must be connected to an input in the

Figure 2.

A digital filter consists of a delay line, multipliers and adders. The synchronizing clock signal is not shown here. The delay line can be built using registers, the constants are stored in registers, and the multipliers and adders consist of logic functions and registers.

All of this can easily be implemented in an FPGA, which is why FPGAs are so suitable for digital signal processing applications.



bottom right corner. These distances can create problems at high data rates with correspondingly short pulse lengths. Propagation delays become significant, pulse edges are no longer aligned, and it is increasingly difficult to keep the gates synchronized with each other.

This has led to an overall FPGA architecture that is divided into function blocks. The positioning of the blocks on the chip is carefully chosen to allow them to be used optimally in most standard applications. Special functions are sometimes available for special applications, which means that FPGAs are not all created equal. The blocks present in every modern (but with various names) are:

- Logic blocks
- I/O blocks
- Computation blocks
- Memory blocks
- Clock blocks

Along with these general functional blocks, FPGAs may also contain processor blocks, flash memory, and peripheral devices such as Ethernet controllers, memory controllers, ADCs and serial communication controllers for SPI, I²C, etc. With all these extra functions, FPGAs are heading in the direction of complete systems, which is sometimes called 'system-on-chip (SoC) with FPGA'. The blocks can be interconnected using a flexible system of configurable links.

The blocks

Let's have a closer look at the previously mentioned functional blocks.

Logic blocks

Logic blocks occupy most of the area of the FPGA and are referred to as the 'fabric' of the FPGA. In a manner of speaking, the applications are woven on this fabric. The example described at the start of this article belongs to this block category. A logic block is a configurable arrangement of gates and registers based on a look-up table (LUT) architecture. The LUTs have the same operating principle as the previously described AND gate, but are usually larger and have more inputs. A LUT can be regarded as an n-bit register (e.g. 16 bits) that can be used to perform logic operations. It is also possible to use the LUTs as shift registers. In addition, the LUTs can be used as memory, in which case they are called 'distributed RAM' to distinguish them from the memory blocks. With the addition of a few multiplexers and some glue logic, the versatility is increased and the logic block can also do a bit of computation.

Since the FPGA has so many blocks of this sort, you can configure several of them in parallel to form a multiplier or an adder. A digital filter con-

Table 1. Functional blocks of the Xilinx Spartan 3E XC3S250E-4VQG100C.

Gates	250,000
Logic blocks	612
I/O blocks	66, of which 7 input-only
Computation blocks	12 multipliers
Memory blocks	12, with a total of 221,184 bits (216 Kbit)
Distributed memory	39,168 bits (38.25 Kbit)
Clock blocks	4

The number of I/O blocks depends on the device package. We chose the VQ100 option with 100 pins. A logic block, which Xilinx calls a 'configurable logic block' (CLB), consists of four slices. Each slice contains two 16-bit LUTs, half of which can be used as distributed RAM. The other half is for logic functions. Each slice also has two flip-flops, so each CLB has a total of eight flip-flops (see Figure 3).

sists of a number of memory elements, a few multipliers and an adder, and digital Fourier transform logic has the same basic recipe (see **Figure 2**). This means that logic blocks contain all the basic ingredients you need to put together digital filters or Fourier transform stages. Although this can't be done with a single logic block, it is readily possible with several blocks. This makes FPGAs especially suitable for digital signal processing. If you fit an FPGA between an ADC and a DAC, you can do really nice things with audio, video, radio or what have you — and with several channels at the same time.

FPGAs with over a million logic blocks are currently available, such as the Xilinx Virtex-7 XC7V2000T with 1.2 million LUTs and 2.4 million flip-flops. This IC contains nearly seven billion transistors. For comparison, Intel's ten-core Xeon Westmere-EX processor contains a paltry 2.6 billion transistors.

I/O blocks

The logic blocks on the chip, as well as the other functional blocks, need to be able to communi-

cate with the outside world via the pins of the IC (the previously mentioned XC7V2000T has 1200 pins). The communication paths are provided by the I/O blocks.

An I/O block is more than just a pair of transistors in push-pull configuration. I/O blocks support a variety of communication protocols (DDR, PCI, LVTTTL, LVCMOS, LVDS, etc.) as well as single-ended and differential signals, and they can achieve extremely high data rates.

For this purpose they are equipped with delay lines, synchronization flip-flops and parallel/serial converters (collectively known as SerDes, which is short for serialiser–deserialiser). Pull-up and pull-down resistors are also present, and termination resistors are often included.

The I/O blocks are arranged in banks, each of which has its own supply voltage. This allows different banks to support different communication protocols, which enhances the versatility of the FPGA. To reduce current consumption and keep electromagnetic radiation within reasonable lim-

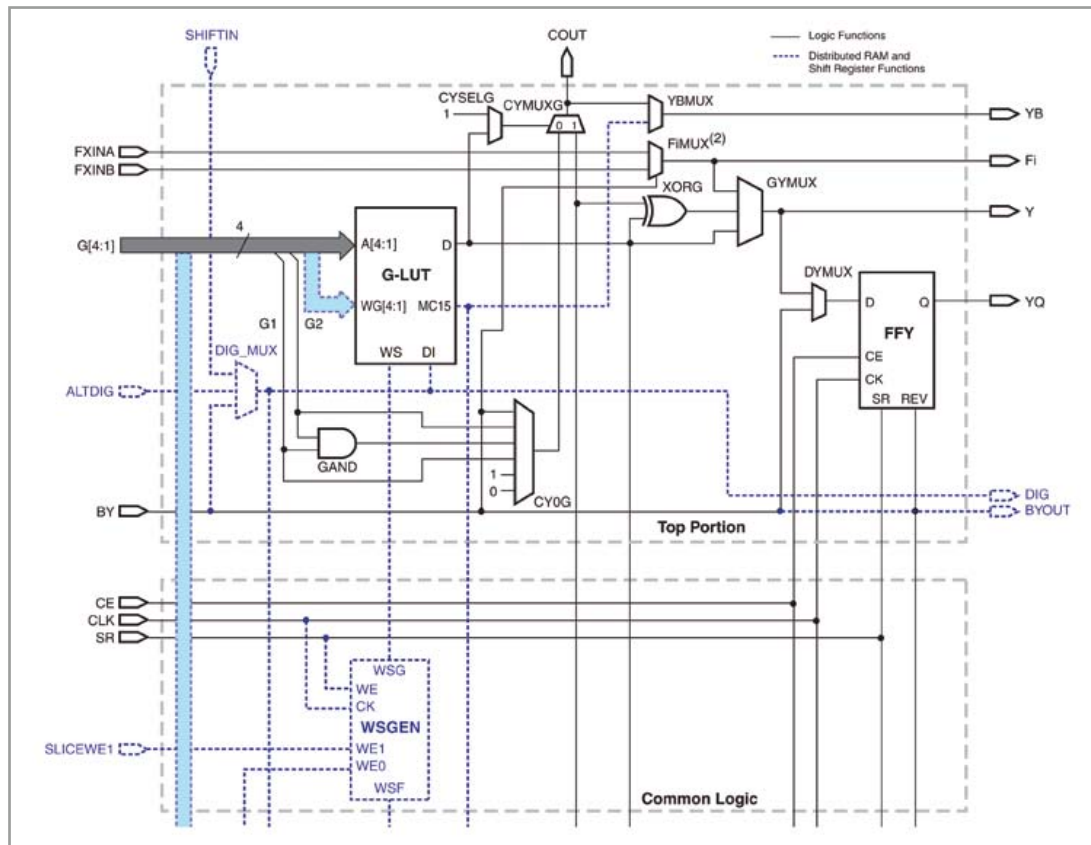


Figure 3. Block diagram of part of a slice. The missing part is the bottom portion. It is largely the same as the part inside the dashed line, which is called the top portion.

The circuit

As can be seen from the block diagram in **Figure 4**, the circuit primarily consists of an ATmega32U4 microcontroller, a Spartan3e FPGA, a power supply section and a MicroSD card socket.

The main component is the FPGA (see also the full schematic diagram in **Figure 5**). We decided to use the Xilinx XC3S250E-4VQG100C, a member of the Spartan3e family. It has 250,000 system gates, over 5,500 logic cells and 612 logic units, and the maximum clock frequency is 572 MHz (see also **Table 1**). We looked for an FPGA with a small package in order to keep the board dimensions compact. The device we selected is housed in an SMD package with 100 contacts, of which 66 are I/O contacts (enough for most applications). The amount of logic on board is sufficient for even fairly complex projects.

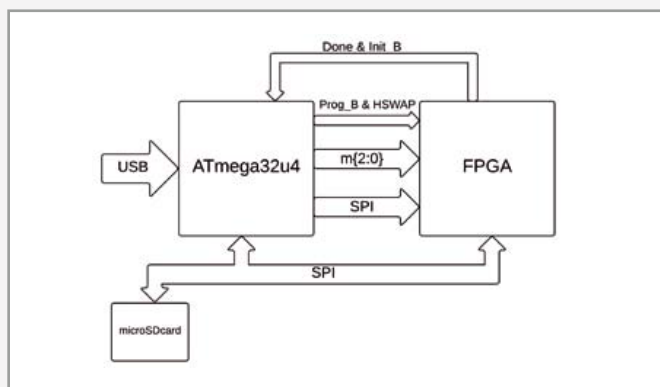


Figure 4.
Block diagram of the Elektor FPGA development board. The main components of the circuit are an FPGA, a microcontroller and a connector for a MicroSD card.

The ATmega32U4 microcontroller handles the control tasks on the board. It supports communication with a PC via USB, provides a virtual COM port, and allows the PC to treat the MicroSD card as a mass storage device. It also reads the data from the MicroSD card and programs the FPGA. We selected this microcontroller because it combines familiar ATmega technology with USB hardware. Although it's a relatively slow 8-bit device, it's perfectly adequate for this application. As usual, the firmware for the microcontroller is available on the Elektor website [1], and the microcontroller on the fully assembled board supplied by Elektor is pre-programmed. The MicroSD card socket was chosen specifically for its small footprint. It is wired for use in SPI mode, which is supported by most MicroSD cards. The two unused I/O lines are tied

to +3.3 V by pull-up resistors to prevent the occurrence of undefined states. There is a card-detect switch that indicates whether a card is inserted in the socket.

The power supply section is rather elaborate for a circuit of this size. This is primarily because the FPGA needs three different supply voltages: 3.3 V, 2.5 V and 1.2 V. The MicroSD card and the microcontroller are also powered from the 3.3 V rail. Extra capacitors are included to ensure stability and decouple the different supply voltages.

The board is powered from the USB port if a jumper is fitted on JP1. This +5 V supply voltage is also available on pin P26 of connector K5 on the card edge for external use. **Caution:** The I/O connections on the board cannot withstand +5 V. The board can also be powered locally if the jumper is not fitted on JP1. In this case an external +5.5 V power supply can be connected to pin P26 of connector K5, eliminating the need for a USB connection.

Due to the large number of signal lines and supply planes, it was necessary to design a four-layer board for this project (**Figure 6**). However, this doesn't create any problems for users, since a fully assembled board can be ordered from the Elektor Shop at a very reasonable price.

An unusual configuration method

FPGAs usually do not support nonvolatile programming, so they must be configured anew each time the circuit is powered up. With most FPGA development boards this is done with the aid of a PC and a JTAG programmer, which is also possible here via connector K3. Operational devices with FPGAs often use flash memory connected to the SPI bus to hold the configuration data. During the start-up process, the FPGA reads the configuration data from the flash memory.

Here we take a completely different approach, as illustrated in **Figure 4**. We use a microcontroller that utilizes the SPI protocol to load the configuration data into the FPGA. This is called Serial Slave mode. This mode is normally used in large systems with multiple FPGAs, but it can also be used with just one FPGA.

The microcontroller can communicate with the MicroSD card over the SPI bus. This card must be formatted in FAT16 or FAT32 format to allow it to be used with an operating system. Consequently, the microcontroller must be able to work with this file system in order to read a file from the MicroSD card.

The microcontroller must also interpret and execute the SCSI commands from the host system (PC) to allow the card to be used with an operating system.

This approach makes the work flow very simple. After generating a configuration file with the Xilinx ISE software, you can use Windows Explorer to copy the file to the MicroSD card, since the board appears in the Windows Explorer window as a USB drive with its own drive letter. After this you only have to rename the file 'config.bin', eject the drive from the PC, and reset the board. The new configuration data is loaded when the board starts up.

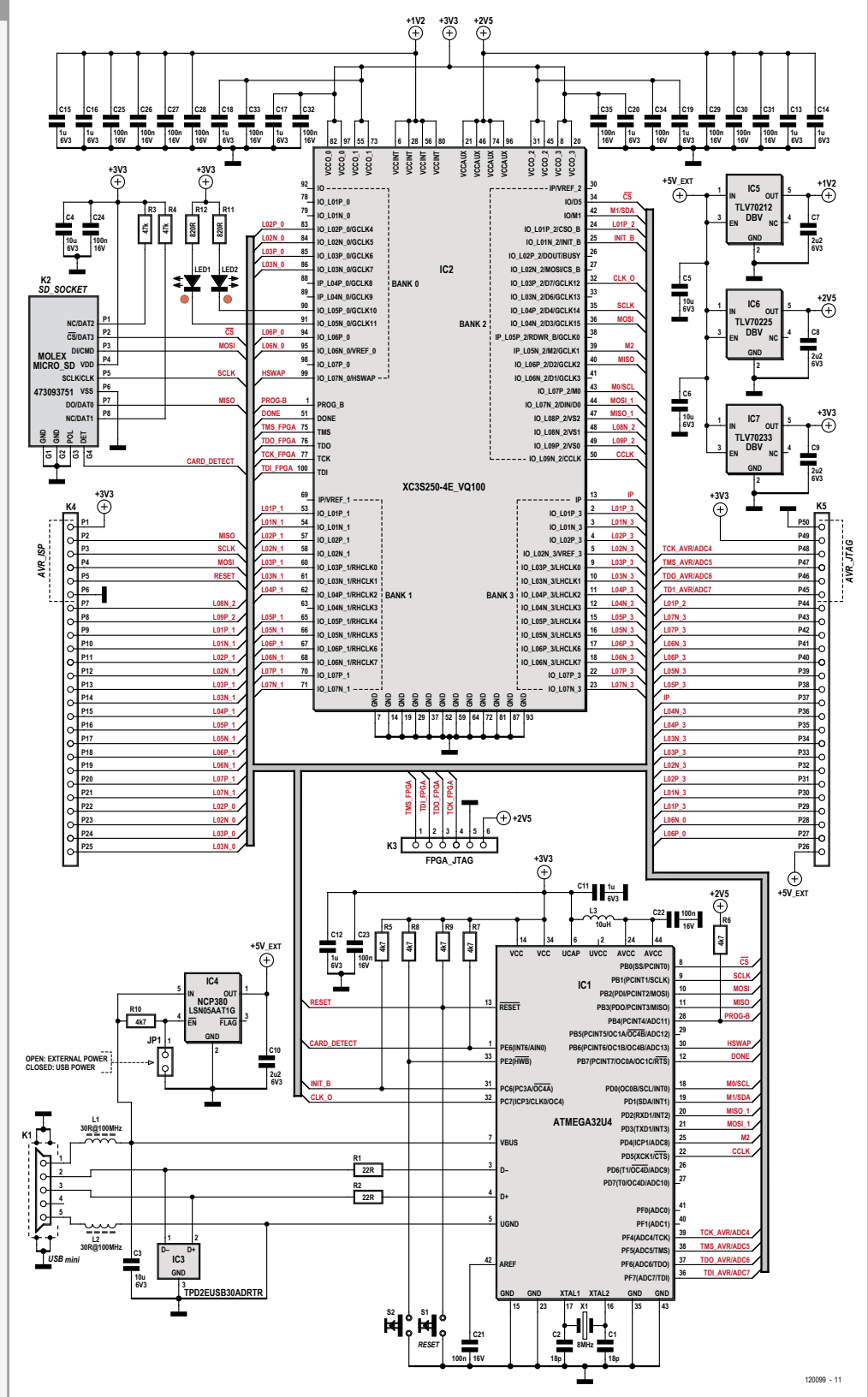
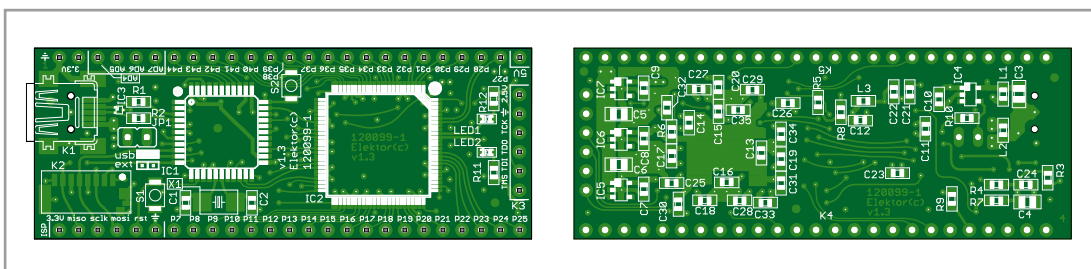


Figure 5. Schematic diagram of the FPGA development board. The complete schematic takes up nearly an entire page, but all of the components fit on a PCB measuring just 65 x 25 mm.

Figure 6.
The PCB is built as a four-layer board to allow all of the interconnects to be implemented in such a small PCB area. The board is supplied fully assembled.



its, the edge steepness and maximum source/sink current per output can be set by the user.

Computation blocks

Since FPGA users often use logic blocks to perform computations, manufacturers have integrated ready-made computation blocks in their FPGA products. This usually involves multipliers, but some manufacturers go a step further

and include DSP functions that can be used to perform frequently used computations, such as $a + (b \times c) \rightarrow a$ (multiply-accumulate operation).

Memory blocks

Along with distributed RAM in the form of LUTs, FPGAs have RAM blocks with a total capacity that can reach the megabit range. Integration of memory on the chip facilitates extremely high data processing rates. It also allows complete microcontroller systems to be implemented in a single FPGA. RAM blocks can also be used as first-in, first out (FIFO) data buffers by providing them with two independent ports. These buffers can be used to synchronise systems that operate at different speeds, among other applications. The word width of the memory is variable, so blocks of various sizes (like 1-bit and 32-bit) can be used intermingled.

Clock blocks

Although it is theoretically possible to implement functions in asynchronous logic (i.e. without a clock signal), this is usually not a good approach for relatively complex systems. Differences in signal path length in the circuit lead to time differences, with the result that signal edges that were aligned at the point of departure are no longer aligned at their destinations. This gives rise to unintentional states in the circuit, which lead to errors.

Figure 7.
Pinout of the FPGA used on the board. The orange and green pins are brought out to the connectors on the board, while the blue pins form the interface to the microcontroller. The two yellow pins are connected to LEDs, and the black pins are used for the JTAG interface, which can be accessed via a separate connector.

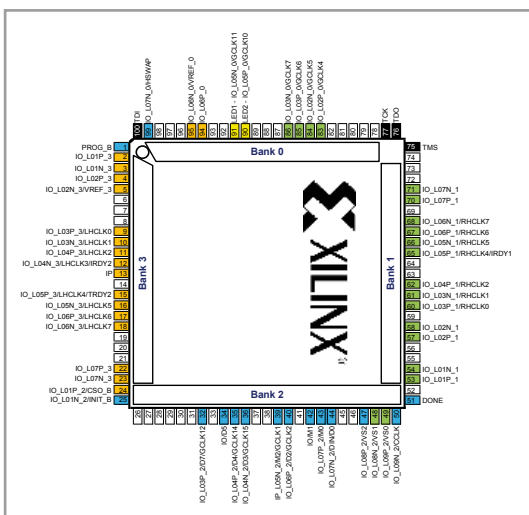
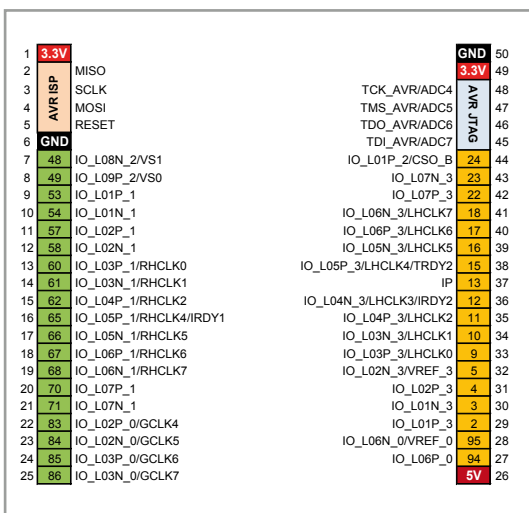


Figure 8.
The pins of the expansion connectors. The orange and green pins correspond to the pins shown in Figure 7. These connectors can be used to program the microcontroller. The pins of the AVR JTAG port can also be used as analog inputs.



and shift options in the clock blocks. The clock signals can also be distributed over special clock networks.

Another situation where using clock blocks is a good idea is when you have subsystems operating at different speeds (different clock domains). In this situation the clock blocks can be configured to generate different frequencies based on the same input clock. The clock frequencies can be very high, extending into the gigahertz range, which can be very beneficial for the speed of the circuit.

Practical aspects

After this introductory description, it's time to turn our attention to practical aspects and to start experimenting with a real FPGA. To this end, Elektor designer Raymond Vermeulen has developed a handy little FPGA development board that eliminates one of the main hurdles of most other FPGA development boards: the awkward method for programming the configuration in the FPGA. The software tools for configuring FPGAs are rather complex, and in many cases you also need an external JTAG programming device to download the configuration data into the IC. Raymond came up with the idea of making the FPGA board look like an intelligent USB stick to the host PC, together with a MicroSD memory card on the board. The configuration file can simply be copied to the memory card, just like any other file. When the board is restarted, the file is read from the 'virtual USB stick' and the data is loaded into the FPGA. This elegantly eliminates the thorny problem of developing specific drivers for specific programming hardware.

Coming up next

FPGA configuration is a complex subject, so our aim in the next few instalments of this series is to explain, as clearly as possible, how to create useful applications. If things go according to plan, in next month's issue of *Elektor* we will first explain how to install the free Xilinx ISE WebPACK software and how you should use it to create a simple project that causes the LEDs on the FPGA development board to blink. To fill the time until the next instalment appears, we recommend that you get a head start by downloading the Xilinx software [2]. The WebPACK version is free, but it weighs in at a hefty 6 MB. If you have a relatively low-speed Internet con-

The fully assembled FPGA development board is available from Elektor for approximately £ 50 plus shipping charges.



Visit www.elektor.com/120099

nection, it's a good idea to order the free software DVD from Xilinx (see **Figure 9**). It should usually arrive within two weeks.

(120099)

Internet Links

[1] www.elektor.com/120099

[2] www.xilinx.com/support/download/index.htm



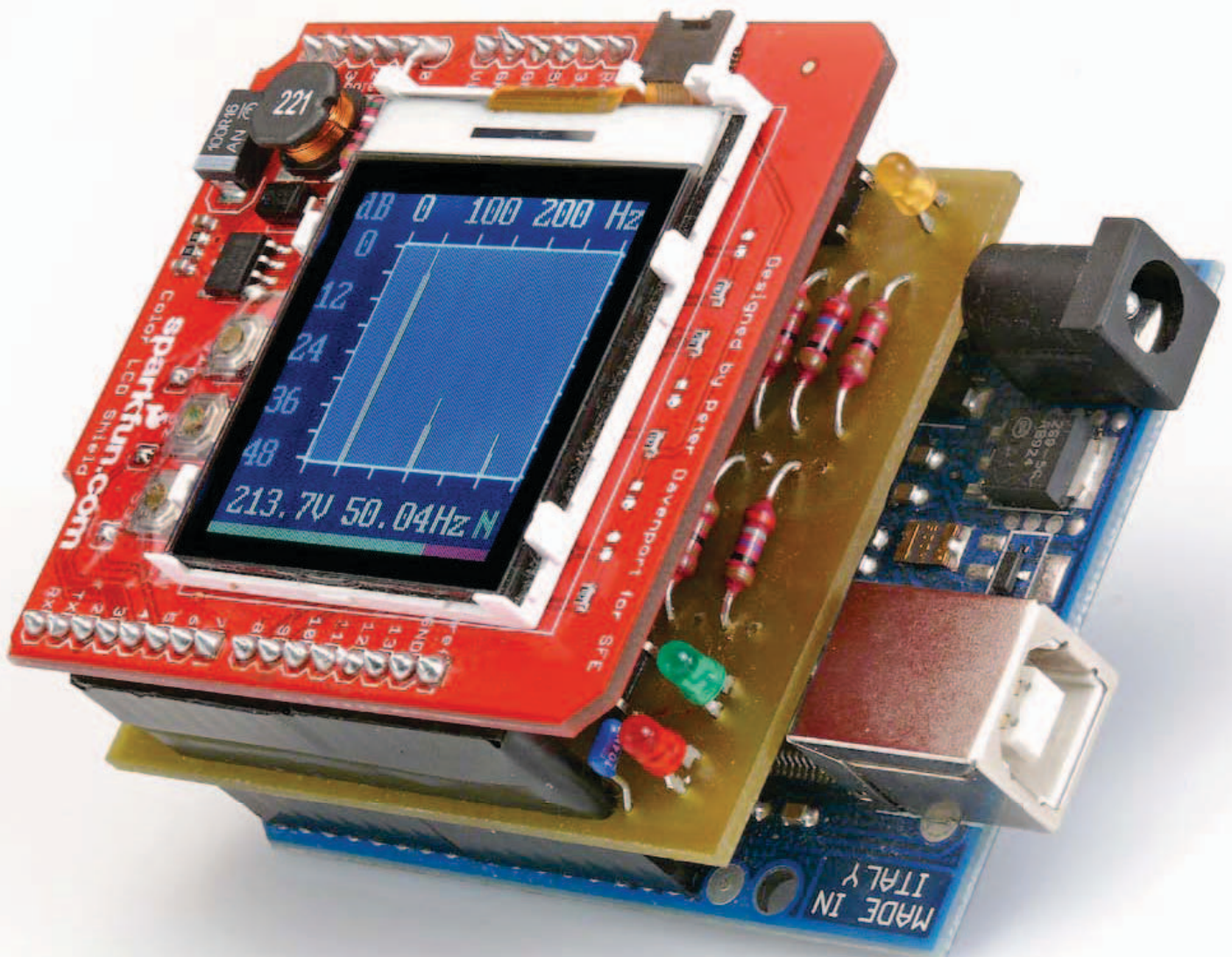
Figure 9.
The free DVD from Xilinx, which contains the ISE WebPACK software and other FPGA tools.

Arduino AC Grid Analyser

Frequency analysis on a mini colour display

By **Fidelis Theinert**
(The Netherlands)

A fully fledged spectrum analyser on a basic Arduino board? In this article we'll show you how this is possible when the software is tailored to the available processing power. A standard Arduino can be turned into an AC line ('mains') supply spectrum analyser with the addition of a filter shield and a display shield. It uses a Fast Fourier Transform to visualize the 'pollution' on the AC grid supply at home or at the office.



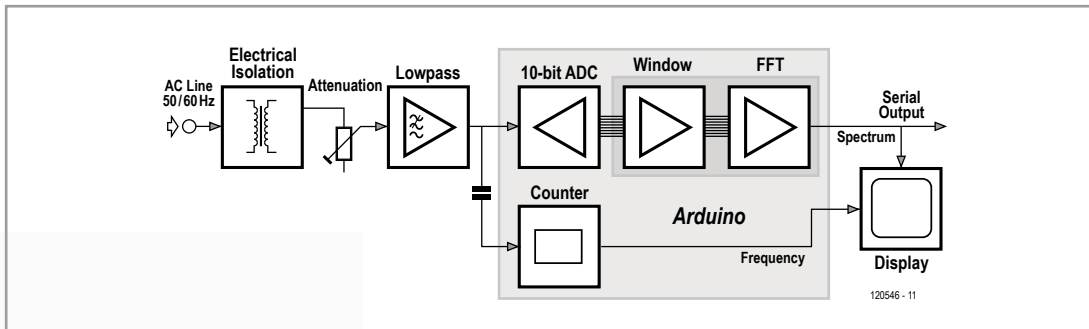


Figure 1.
Block diagram for the
spectrum analyser.

A simple Arduino board is much more powerful and versatile than many readers realize. In this article we'll use a practical example to show you what the capabilities are of this small 8-bit microcontroller, but also what its limitations are. We will also explain how the various components of the microcontroller can be used efficiently and which free development tools are required to set up projects such as these. The software and hardware are of a modular construction, which makes it possible to use the individual modules in other projects as well. The application was tested successfully on an Arduino Duemilleanova, UNO R1, R2 and R3.

The circuit analyses the AC grid voltage using a Fast Fourier Transform (FFT) and shows any harmonics present on the display. This can then be used to assess the quality of the supplied (energy) signal. These higher harmonics cause losses in machines and motors that are converted into heat, but which you still have to pay for. Furthermore, the efficiency of switch mode power supplies in PCs and other devices is compromised by a supply signal that isn't an ideal sinewave.

Fourier analysis

The Fast Fourier Transform is a mathematical algorithm that converts a discrete signal from the time domain into the frequency domain. The improvement in execution time of the FFT compared with the 'Discrete Fourier Transform' (DFT) can be quite substantial for a large number of samples (N). With an FFT the Fourier transform on a signal with N samples is split into two transforms with lengths of $N/2$.

This results in a much reduced execution time since there are fewer calculations to be done, but it does mean that the number of samples (N) has to be a power of two (64, 128, 256, ..., 4096 ...). In practice this is not a problem since the

length of the array can usually be freely chosen. According to the Nyquist-Shannon sampling theorem the bandwidth of the signal may not be greater than half the sampling frequency. In other words, the sampling frequency has to be at least twice the value of the highest frequency we want to investigate. With a sampling frequency of 800 Hz you can therefore analyse a signal with a bandwidth of 400 Hz, which makes it possible to show the spectrum of the local AC grid supply up to its sixth harmonic ($60 \text{ Hz} \times 6 = 360 \text{ Hz}$), or seventh ($50 \text{ Hz} / 350 \text{ Hz}$).

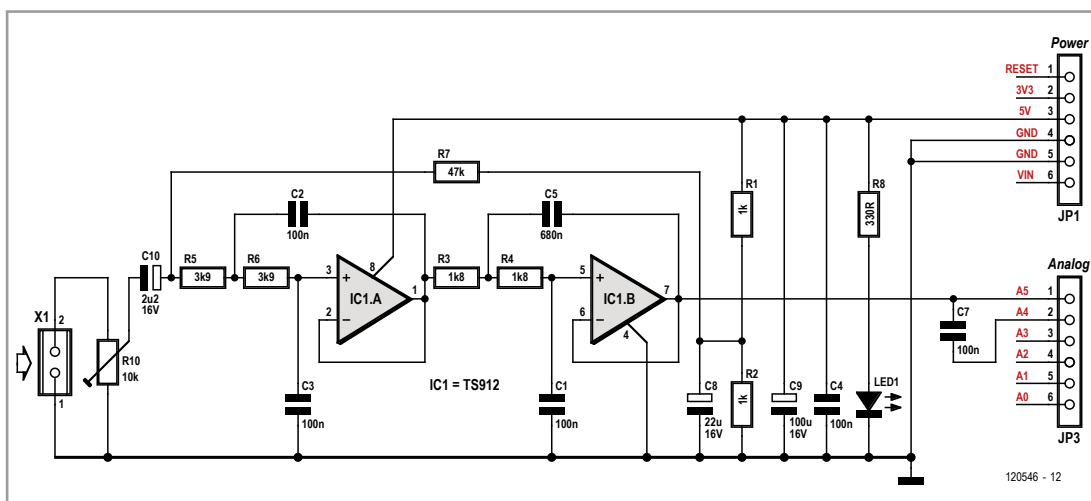
Hardware

The hardware consists of three boards: the Arduino board [1], a filter shield developed by the author and an LCD shield [2]. The two shields can be mounted on the Arduino board (with the LCD shield on top, of course), which results in a compact and sturdy module without any loose wires. **Figure 1** shows a block diagram of the complete setup.

When the Arduino has calculated the spectrum it is also possible to send the result to a PC via the serial port. The graph can then be plotted using a terminal program (Putty, for example). The transmission of the data requires a bit more time since the whole graph is sent to the UART of the PC at a rate of only 57,600 baud. This feature will be particularly useful to those of you who don't have an LCD shield.

To make sure that the input signal doesn't have any frequency components above 400 Hz, a fourth order Sallen-Key lowpass filter (or anti-aliasing filter) with a bandwidth of 350 Hz is put in front of the input (**Figure 2**). The two Butterworth sections are made around a TS912 dual opamp. This opamp has a rail-to-rail output and a wide operating voltage range of 2.7 V to 16 V, which

Figure 2.
Anti-aliasing filter (350 Hz
Butterworth lowpass filter).



means it can be powered directly from the 5 V supply voltage on the Arduino board.

The output signal from the filter goes directly from pin 7 of IC1 to channel 5 of the ADC on

the microcontroller (an ATmega328P). The output of the opamp is also capacitively coupled to connection A4 of the analog input connector of the Arduino board, for measuring its frequency. In **Figure 3** you can see the shield board designed

FFT tricks

The Cooley-Tukey 'butterfly' algorithm implemented here is a so-called 'in-place transformation'; this means that the result of the transformation is stored in the same buffer where the original samples of the input signal were stored. The input signal is therefore no longer available after the conversion. The advantage of this is that the memory required for the buffer is halved, which is extremely useful when the microcontroller used (ATmega328) only has 2048 bytes of memory available.

As was mentioned previously, the number of samples has to be a power of 2. This condition is satisfied by the buffer size of 256 (2^8) samples. The ADC has a resolution of 1024 levels, which means that two bytes are required to store each sample, or 512 bytes for all of the samples. During the calculations there is also a requirement for a buffer to hold the imaginary part of the numbers, since the Fourier transform is a conversion of complex numbers. This buffer obviously needs to be the same size as the buffer for the real part and therefore requires 512 bytes as well. In total we therefore need 1024 bytes to store the values for the FFT, which is half of the available memory.

In order to use the maths library of the GCC compiler it

would be useful if we could store the samples as 'single precision floating point' variables instead of integers. (In an FFT we don't just have simple mathematical operations such as addition and multiplication, but also sines, cosines, square roots and exponentiation.) However, these require 4 bytes each for storage, making the buffer 2048 bytes long, which is as much as the available memory in the microcontroller, and therefore too big for the chosen platform. One way round this dilemma would be to halve the number of samples for the FFT to 128, but this will result in a lower spectral resolution and is therefore undesirable.

A different solution is therefore required if we want to keep working with 256 samples. The real and imaginary parts of the complex numbers are still stored as integer values (2 bytes/sample) and converted into a 'floating point' variable before each mathematical operation and converted back into integer format afterwards. The extra overhead for these two conversions obviously makes the system a bit slower, but it is acceptable. The rounding error introduced by the conversion ($1/65536$) is a factor of 64 smaller than the quantization error of the ADC ($1/1024$) and can therefore be ignored (see the functions 'FFT_Int2Float()' and 'FFT_Float2Int()' in the module `fft_lib.c`).

for this filter. Note that there are some extra components on this board that aren't used in this application. The board layout is available from the associated web page for this article [3].

The AC line voltage is electrically isolated using a AC power adapter (an 'old fashioned' type with a transformer). The output of the adapter is connected to terminal block X1. Since the filter has a gain of one, a signal with an amplitude of $2 V_{rms}$ ($5 V_{pp}$) is sufficient for the dynamic range of the ADC in the ATmega. The output signal from the transformer is not distorted in practice because the transformer is effectively unloaded, as can be seen in **Figure 5** (the supply for the analyser comes from a separate adapter or via the USB connection).

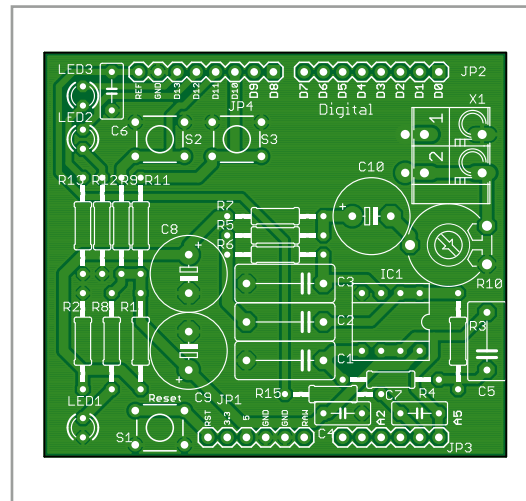
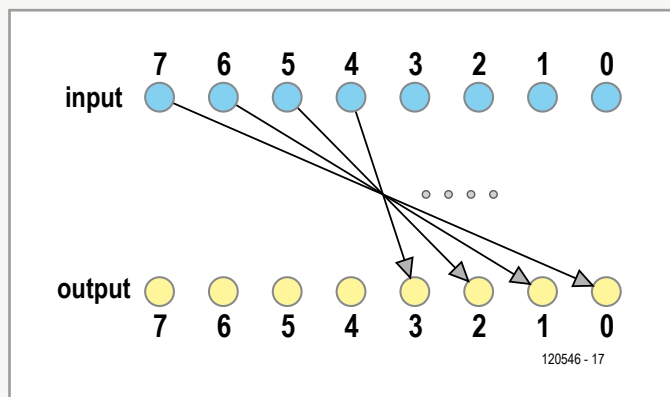


Figure 3.
The PCB layout for the filter shield.

This way it's possible to measure all harmonics in the domestic AC line supply fairly accurately. Potentiometer R10 should be used to reduce the output signal from the adapter to about $1.7 V_{rms}$ at C10 when the input voltage is at its nominal level, i.e. 115 V or 230 V. You can use a reference

multimeter to make a note of the line voltage and then adjust R10 so that the voltage shown on the display corresponds to that on the multimeter. We recommend that you use an AC power adapter with an output voltage of less than 20 V.

In order to use the memory as efficiently as possible, the samples from the ADC are initially stored in the buffer for imaginary numbers. At that point in time this buffer isn't used by the FFT routine.



Bit reversal algorithm

The samples are then normalized and scaled, with the results stored in the buffer for real numbers.

The imaginary buffer can then be cleared and the window

function can be applied to the real part of the samples (this window function is necessary in order to satisfy the boundary conditions for the FFT).

An important aspect of the FFT implementation is the so-called 'bit reversal' algorithm, which is used for addressing the samples in the buffers. At first this may look like a simple operation (the bits of the value of a register have to be put in reverse order, as shown in the figure above), but it isn't very easy to implement. Fast implementations use extra hardware registers for this. Another alternative that is often used is a table, but this requires a lot of memory and that is in limited supply in a small microcontroller. The slowest method is to shift all bits out of one register and shift them into another register in reverse order. This requires a loop with a counter and a total of 16 shift operations. A combination of these methods is used here, which provides a good compromise between memory usage and processing time. The byte is split into two lots of 4 bits. Each of these 4-bit values is reversed using a table and the results are then combined in the right order into a byte again (see the function 'fft_BitRev()' in module fft_lib.c). The resulting time saved is $3 \mu s$ ($2 \mu s$ instead of $5 \mu s$) for each instance this function is called, but it does require an extra 75 bytes for the code.

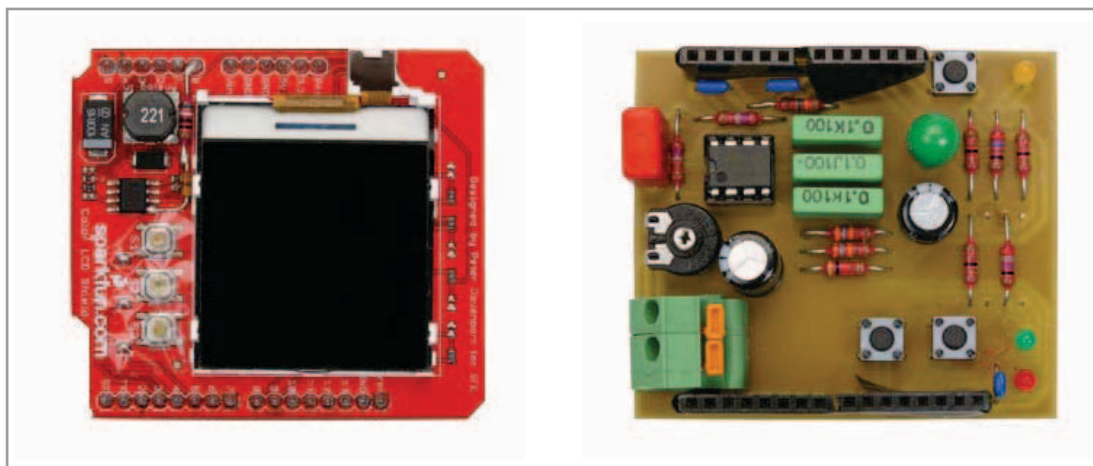


Figure 4.
The three boards next to each other.

Development Environment

We haven't used the standard Arduino programming environment for the software development. Instead, we've used the 'GCC cross-compiler' for the AVR microcontroller series. The 'cross-avr' compiler can be installed under Linux from the repositories of the distribution, along with the associated 'bin-utils'. For Windows you can install the 'WinAVR' [4] package. For both operating systems we've used version 4.3.3 of the GCC compiler [5]. For uploading the compiled hex file we've used the program 'avrdude'. This program is part of the WinAVR package for Windows, but

there is also a Linux version available. With both versions you'll get scripts that upload the executable to the Arduino. You will have to edit these files so that you have the correct serial port for communications with the Arduino and the path to the executable on the PC.

We also need a program called 'make', which compiles and links the source code of all modules with the help of a make-file. This program is included with WinAVR as well, and is also available for Linux.

To edit the source code you can use your preferred editor. Eclipse with the CDT plugin is rec-

Table 1. Software modules

Name of source code file	Name of associated header file	Description of functionality
avrfft.c	(none)	Main source code file
drw_fft.c	drw_fft.h	High-level draw functions for graph display on LCD
fft_lib.c	fft_lib.h	Definitions and functions for the FFT calculations
freq_cnt.c	freq_cnt.h	Definitions and functions for data acquisition and frequency measurement
lcd_driver.c	lcd_driver.h	Low-level functions and definitions for driving the LCD
menu.c	menu.h	Functions for reading the switches on the LCD shield
plt_fft.c	plt_fft.h	High-level functions for sending plots over the serial port
ser_avr.c	ser_avr.h	Low-level functions for sending and receiving data
std_avr.c	std_avr.h	Commonly used standard definitions and functions
timer_0.c	timer.h	Functions for timers and initialization of system timing

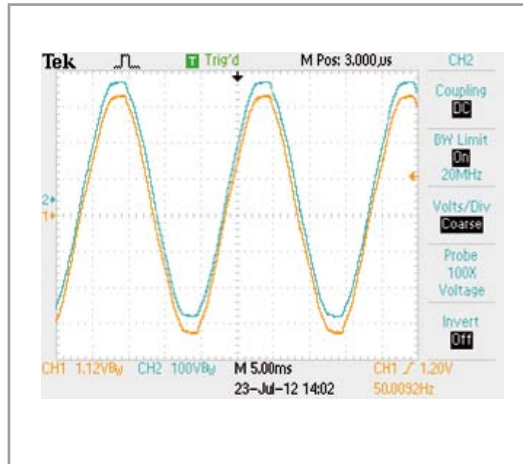
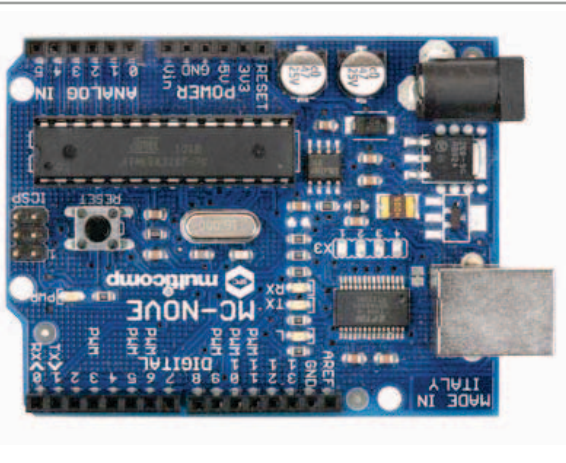


Figure 5.
The AC grid voltage at the primary side of the transformer (blue) and the voltage at the secondary (yellow).

ommended, but Notepad++, Scite or just about any other editor should be fine.

Software

The software has a modular construction and consists of a number of source code files (see **Table 1**).

Each C program begins with the function 'main()', which can be found in the file avrfft.c. When the program starts, it first initializes a number of global system variables and the hardware with the help of the associated init functions. Next, the display on the LCD shield is cleared and the system timer is prepared for later use. The program then comes to an endless loop, where the system timer is set for a period of 2.5 s. This means that a new cycle will start every 2.5 seconds.

All of the steps will always take place in the same order during each cycle. A simplified flow chart is shown in **Figure 6**.

Once the timer (timID) has indicated the start of the next cycle, the 256 samples for the FFT will be read in via the ADC with a sampling frequency of 800 Hz. Next, the true effective value (TRMS) of the signal voltage is calculated. In the next step the frequency of the signal at pin A4 on connector 'Analog In' is measured using the built-in comparator and Timer1 of the microcontroller. The next steps (scaling the measured values and applying the selected window function) prepare the measured signal for the actual FFT calculations. At the end the result of the calculations is sent to the display of the LCD shield to show the spectrum of the signal.

Usage and display

The display on the LCD shield used has a resolution of 131 x 131 pixels, of which a window of 80 vertical and 96 horizontal pixels is used to plot the spectrum. This results in a range of 0 Hz to 300 Hz (96 pixels x 3.125 Hz = 300 Hz) and 10 pixels per 6 dB vertical. The FFT routine calculates the spectrum for 0 to 400 Hz, but frequencies above 300 Hz aren't shown, although they are sent to the PC via the serial port. The measured AC line voltage and frequency are shown below the window, whereas the voltage is also shown as an analog bar.

To the left of the display on the LCD shield are three buttons, which are used to configure the spectrum analyser. Their functions have been programmed as follows. When the analyser is switched on it will immediately start its measurements and will show a logarithmic plot of the measured spectrum on its display. With a press of the top button the display changes to a frequency plot with a linear scale along the vertical axis. Another press of this button changes the display to the time domain (scope function) and the first 96 scaled samples in the input buffer are shown (see **Figures 7 and 8** for the different displays). Another press of the first button shows the complete contents of the input buffer after the window function has been applied. A fourth press switches back to the frequency domain with a logarithmic scale, as it was when the circuit was first switched on. The bottom button is used to select the type of FFT window function. When the analyser is switched on the default window type is Hanning. Pressing the bottom

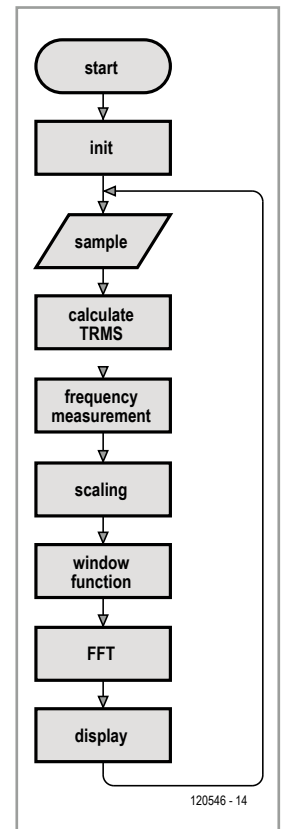
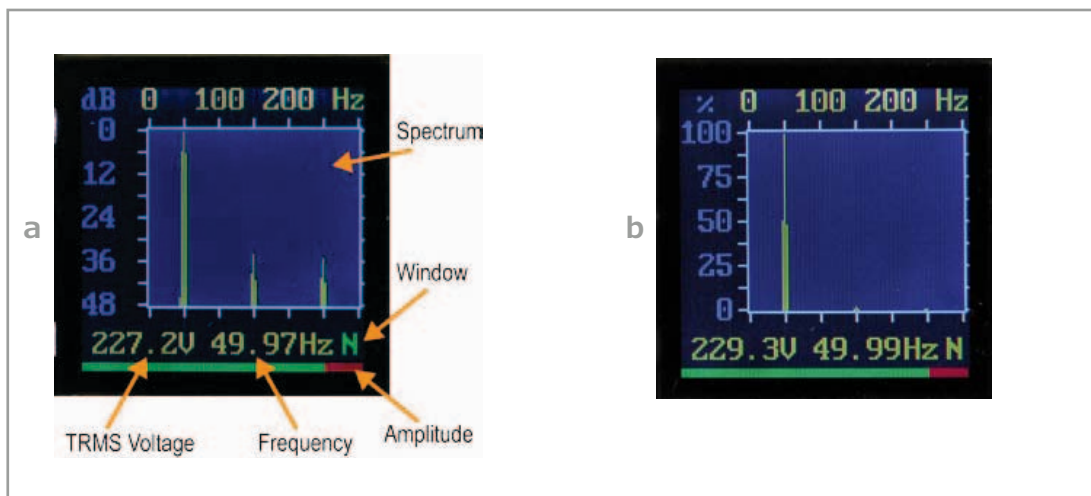


Figure 6.
Flow chart of the program.

Figure 7.
Logarithmic (a) and linear (b) plots of the frequency domain shown on the display.



button switches to a Blackman window and the next press changes to a Hamming window. The selected window function is always shown at the bottom-right of the display: 'N' for Hanning, 'B' for Blackman and 'M' for Hamming.

The middle button turns the serial communication with a terminal on and off. When no data is sent to the computer, the colour of the indicator for the window function turns from yellow to green and back again after each spectrum has been calculated. When the serial communication is turned on the indicator turns from yellow to red.

Performance

Depending on the compiler settings, a calculation of the FFT can take up to 1 second in the most unfavourable condition without optimization (-O0). The size of the compiled program is then about 26 KB. With optimization turned on (-O2) the length of the executable shrinks to about 18 KB and the calculation of the FFT takes only 700 ms. The time taken to run through a complete cycle of the program (measuring the frequency, acquisition, calculating the TRMS, scaling, window function, calculating the FFT, scaling, plotting on the display) takes about 2100 ms with optimization (-O2).

A cycle time of 2500 ms was chosen so that the system has a predictable and constant response. This is obviously when the transmission of data to the serial port is turned off. A detailed description of the '-O' flags can be found in the documentation for the GCC compiler.

1372 bytes have been used of the available memory (RAM 2048 bytes), of which 1024 bytes are for the FFT buffers. This is about 2/3 of the available memory. Only about 18 KB have been used of the available program memory (ROM 32 KB). Bearing in mind that the Arduino bootloader needs about 2 KB, this leaves more than 10 KB of ROM for possible enhancements.

Conclusions

The system described here is a fully functional real-time FFT analyser for the AC grid supply, which uses a 256-point floating-point FFT to calculate a new spectrum every 2.5 s and shows it graphically on a display. All this is implemented on an Arduino platform with only one third of its program memory used.

This application shows how efficient and powerful an 8-bit microcontroller can be when it is used to perform advanced mathematical calculations and display the results graphically, as well as including a user-interface and communications with a PC. But there are also limitations, such as the available memory (which restricts the FFT to 256 points) and the time taken to perform the calculations, although there is further room for improvement.

For the development of this project only open-source tools were used and the Arduino platform itself is completely open and freely available. The software is modular and it is very easy to use any of the modules in another project. The software has been published under the GPL V3

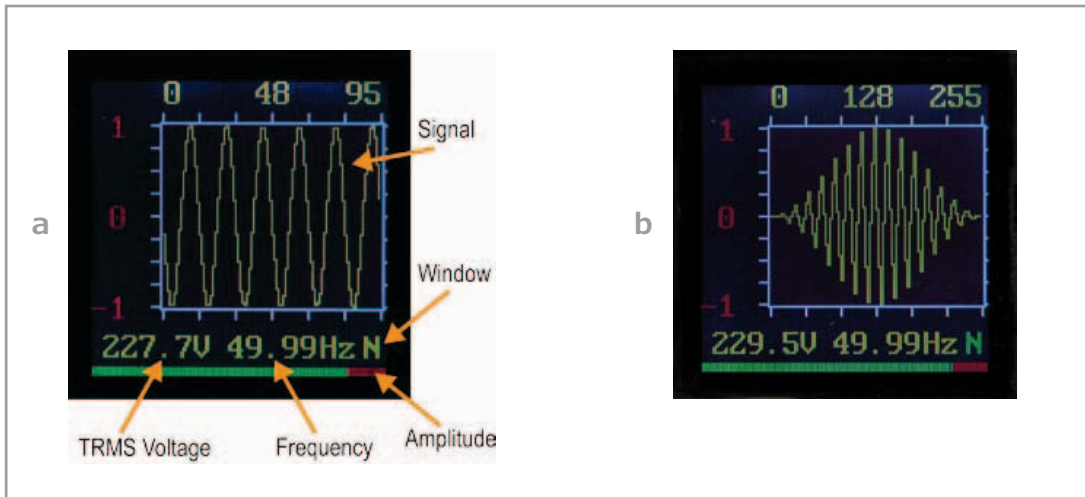


Figure 8.
Plots of the time domain shown on the display (96 samples without a window function (a) and 256 samples with a Hanning window (b)).

license, which means you can use it in your own projects if you want to. The conditions of the GPL are published on the website for the 'GNU Operating System' [6]. The full software package and the board layout can be freely downloaded from the associated web page on the Elektor website [3].

Possible enhancements

Despite the extensive functionality that is already implemented, it would be possible to add a calculation of the THD (total harmonic distortion) to the analyser.

Another enhancement would be a speech analyser that calculates the spectrum of a speech signal with the help of an FFT and shows the result on the display. This does require that the bandwidth of the lowpass filter is adapted. The ADC in the ATmega328 can still operate at its full resolution of 10 bits at a sample frequency of 15 kHz, which is more than enough for a speech signal with a bandwidth of about 5 kHz.

The speed of the system can be improved if the acquisition of new samples happens in the background via interrupts whilst the information is being sent to the display. This results in a 0.3 s (256 samples/800 Hz sample frequency) improvement. It is also possible to drive the display more quickly using the SPI interface of the ATmega328. Another enhancement is to have an elaborate program on the PC to display the graphs, but then we would no longer have a self-contained FFT analyser on an Arduino board.

(120546)

Internet Links

- [1] <http://arduino.cc/en>
- [2] www.sparkfun.com/products/9363
- [3] www.elektor.com/120546
- [4] <http://sourceforge.net/projects/winavr>
- [5] <http://gcc.gnu.org/onlinedocs/gccint>
- [6] www.gnu.org/copyleft/gpl.html

Also read: Grid Frequency Meter, Elektor January 2012.

The author

After completing his studies in electric engineering at Berlin Technical University, Fidelis Theinert worked for over 20 years at various companies in Germany and The Netherlands as a hardware and software engineer and has also taught at colleges. He currently teaches electric engineering at Haagse Hogeschool, Delft, The Netherlands. His expertise is in the field of embedded systems and in particular in signal processing of video as well as audio signals.



Hyperactive LED Xmas Tree

With elaborate light show and daisy-chain capability

By Willem Tak
(Netherlands)



These days it has become impossible to imagine even Christmas decorations without any electronics. While in the past we usually had to make do with a string of tiny, fragile light bulbs, which would flash at the most, now

we can choose from many different types of LED lights with built-in light effects. But as a real electronics enthusiast you can, of course, make one that is much better and more elaborate, such as this Xmas tree which, with 56 LEDs can display an extensive light show lasting more than three-quarters of an hour.

It's that time again to retrieve the Christmas decorations from the attic and to start looking for a (real or artificial) Christmas tree. Fortunately it is now rare for such a tree to be decorated with real candles. These were replaced long ago by miniature light bulbs or even LEDs. This is also true for other Christmas decorations, where LEDs are becoming increasingly more common and are controlled by an electronic circuit.

The author of the electronic Christmas tree described here wanted to make a mini Christmas tree which was a little bit more exclusive than the standard products in the DIY stores or garden

centres. That resulted in a circuit which comprises not much more than a microcontroller and 56 light emitting diodes. The LEDs are arranged on a circuit board in the classic Christmas tree pattern, as you can see from the headline photo. The intelligence is – as is usual with this type of circuit — in the software! This ensures that the microcontroller will light up the Christmas tree in the most interesting patterns (take a look at the video at [1]). It is even possible to join two of these trees together, after which the patterns that are produced will run back and forth between the two trees. The result is an amazingly effec-

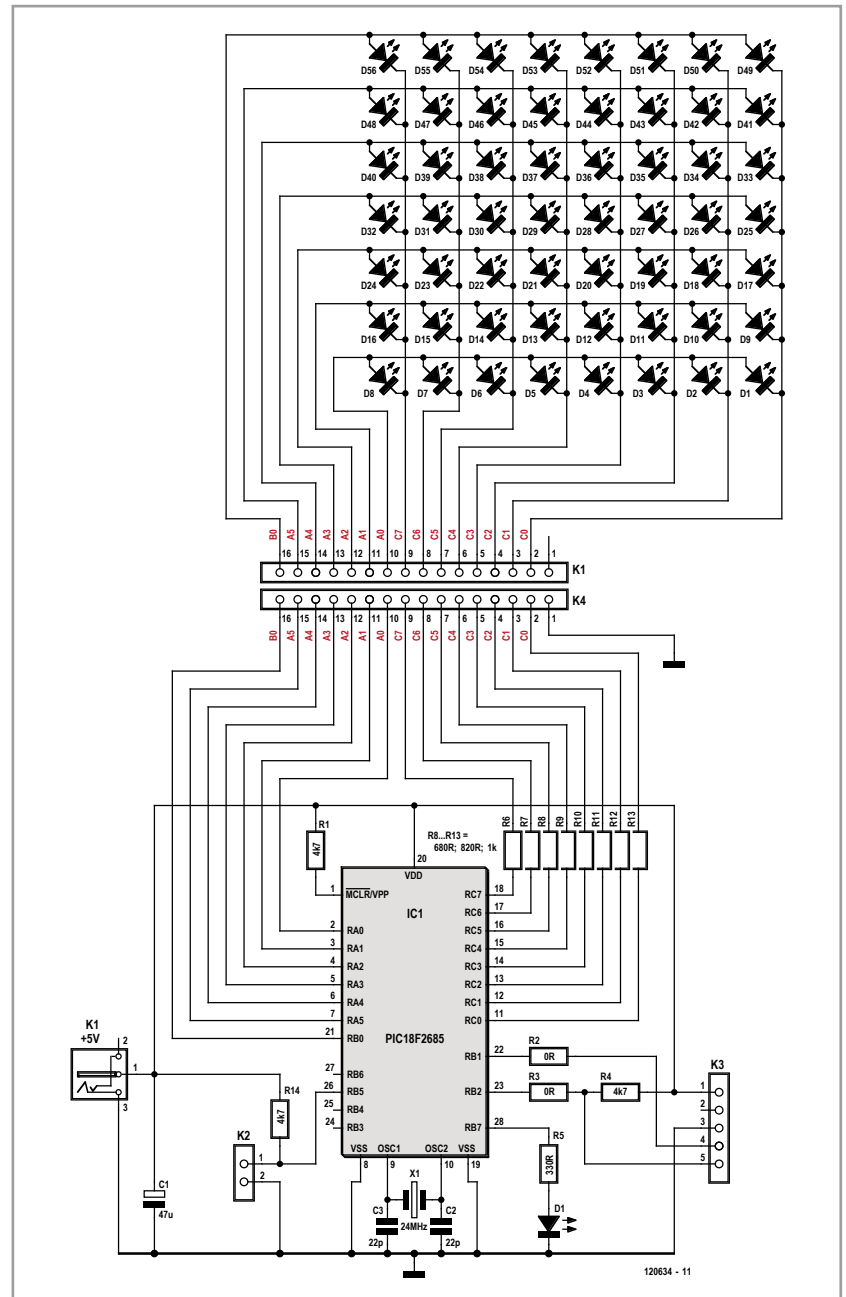
tive eye-catcher when it is placed on the window sill. And it contains more than just a few simple effects – the cycle time after which the entire performance repeats, amounts to no less than 40 minutes!

Little electronics, lots of memory

Seeing that the optical effects play an important role here and it is not the intention that you will see the same pattern again after only a short time, the author chose a microcontroller with a substantial amount of memory. This became the type 18F2685-E/SP made by Microchip, a microcontroller which is provided with 96 KB of flash memory (that is, 96 KB of program space). This memory is, incidentally, nearly entirely filled with the present program, hence the very long run time of over 50 minutes. The microcontroller is clocked using a standard 24 MHz crystal (the internal clock of 8 MHz turned out not to be fast enough for some patterns).

The circuit itself (**Figure 1**) is very simple: the LEDs are driven in an 8x7-matrix directly by the microcontroller. The seven columns of eight LEDs each are connected, one at a time, to the positive power supply voltage by pins A0 through A5 and B0. In addition, the rows of seven LEDs each are pulled to ground via resistors by the outputs of port C. The levels at the C outputs therefore determine which LEDs from each column will light up, while the A and B0 outputs take care of the multiplexing of the columns.

Because of the matrix configuration (each LED is on for only 1/7th of the time) and the limited current that can be supplied by the controller, the LEDs do not light up extremely bright. But then the circuit is not intended for use in environments that are too bright (like a softly lit living room). The author used wide viewing-angle LEDs purchased at SBL [2] and achieved a useful result with those. Good results are therefore obtained especially when you don't use ordinary LEDs, but bright LEDs with a high optical efficiency. The outputs of the controller can be loaded with a maximum of 25 mA. Due to the multiplexing that works out at about 3 mA per LED; in the dark even that can be too much. The eight resistors in the connections to port C determine the current through each LED. Depending in the desired brightness you can use resistors with a value of 680 Ω , 820 Ω or 1 k Ω .

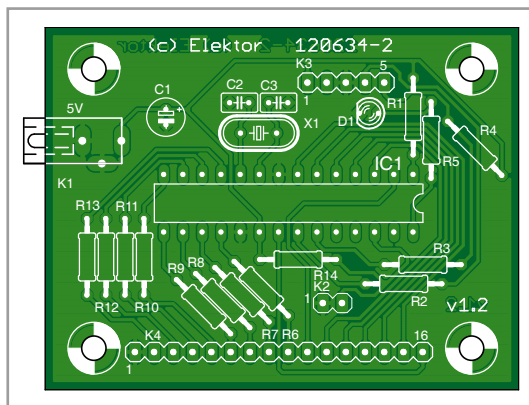


Via the sync-connection (pins 22 and 23, brought out to K3) it is possible to connect two electronic Christmas trees together. Certain patterns will then jump from one tree to the other. In order to obtain this effect, pins 22 and 23 of the two Christmas trees are connected to each other with a crossover cable. The software determines whether a tree is the left or the right version. The l-version indicates (as can be expected) the left tree and r indicates the right. The effects have been written in such a way that the left tree

Figure 1. It's almost unworkable to make it any simpler: a microcontroller drives 56 matrix-connected LEDs directly. All of the real work sits in an extensive program that takes care for the control.

Figure 2.

The circuit board for the microcontroller with a few passive components. A skilled electronics enthusiast can also easily build this on a piece of prototyping board.



should be positioned on the left, when viewed from the front.

The status LED, connected to RB7 is not essential. When the power supply voltage is switched on, or after a reset, the LED will rapidly flash a few times. In this way the correct operation of the PIC and its contents can be verified.

It is best to use a 5-VDC power adapter for the power supply; any standard adapter is certainly sufficient for two trees. When two trees are connected together, the power supply wires can run together with the synchronization lines (all are present at connector K3).

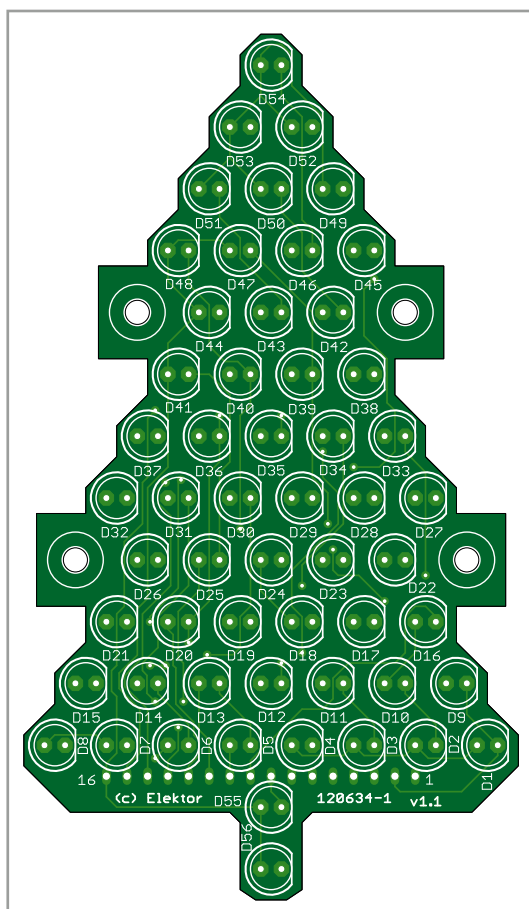
Figures 2 and 3 show the circuit boards designed for this Christmas tree. These are available from Elektor [1]. The construction is very easy, only a few leaded components are used. For the connection between the microcontroller and the display board it is best to use a short length of 16-way ribbon cable.

The software for this circuit can be downloaded free from the Elektor website [1]. There are separate versions for the left and right tree, keep that in mind when programming. If you are going to build only one tree then it doesn't matter which version you program into the PIC. The website also contains a demo video made by the author and the files for the PCB layout, for those who would like to etch their own boards. If you are not satisfied with the patterns offered by default, then you can get stuck into the available source code yourself. But with a program length of over 50 minutes, the pre-programmed light patterns are not likely to bore for a while...

(120634)

Figure 3.

This is what the (double sided) PCB with LEDs in Christmas tree shape looks like



COMPONENT LIST

Controller board

Resistors

R1,R4,R14 = 4.7k Ω
R2,R3 = wire link
R5 = 330 Ω (optional)
R6-R13 = 680 Ω (820 Ω , 1k Ω , see text)

Capacitors

C1 = 47 μ F 16V

C2,C3 = 27pF ceramic

Semiconductors

D1 = LED, red, 3mm (optional)
IC1 = PIC18F2685-E/SP, programmed

Miscellaneous

X1 = 24MHz quartz crystal
K1 = DC adapter socket (e.g. Lumberg type CUI-PJ-031DH)
K2 = 2-pin SIL pinheader (optional)

K3 = 5-pin SIL pinheader

K4 = 16 pin-SIL pinheader (or short length of 16-way flat cable)

Optional: pushbutton for mode and reset
PCB # 120634-1 [1]

LED board

D1-D56 = high-efficiency LED, 5mm
K1 = 16-pin SIL pinheader (see K4 control-board)
PCB # 120634-2 [1]



FRONT PANELS & HOUSINGS

Cost-effective single units and small production runs

Customized front panels can be designed effortlessly with the Front Panel Designer. The Front Panel Designer is available free on the Internet or on CD.

- automatic price calculation
- delivery in 5–8 days
- 24-Hour-Service if required



Sample price: 34,93 € plus VAT/shipping

Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 805 86 95-0
Fax +49 (0)30 805 86 95-33 · Web info@schaeffer-ag.de · www.schaeffer-ag.de



THE ORIGINAL SINCE 1991
PCB-POOL
Beta LAYOUT

FREE Stencil

with every PCB Prototype order

EAGLE order button

pcb-pool.com/download-button
20% off on your first PCB order

Free Phone UK: 0800 389 8560
sales@pcb-pool.com



www.pcb-pool.com

Beta
LAYOUT
create:electronics

PCB-POOL is a registered trademark of Beta LAYOUT GmbH. All registered brands remain the registered trademarks of the respective manufacturer.

SPECIAL CHRISTMAS OFFER

3 ELEKTOR PRODUCTS FOR JUST £44.00



20 products to choose from!

Discounts up to 57%

Limited to stocks

Valid until 1 January 2013

MAKE YOUR CHOICE TODAY AT WWW.ELEKTOR.COM/CHRISTMASOFFER

Embedded Linux Made Easy (6) Networks and servers

By
Benedikt Sauter [1]

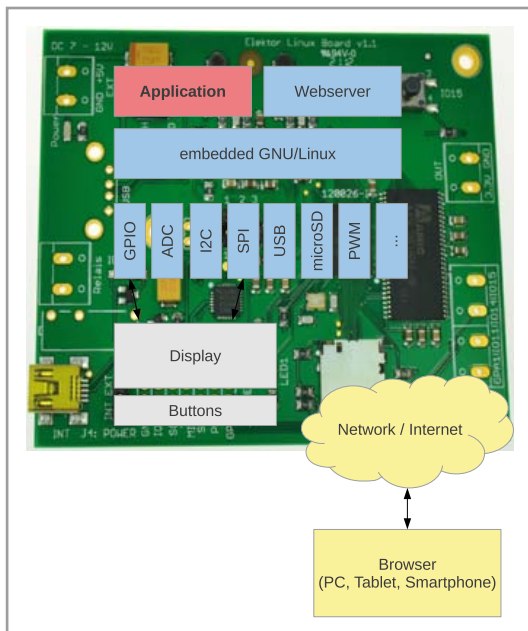


Figure 1.
Overview of the system.

This series has already covered the most important aspects of Linux from the bootloader and kernel to the installation of device drivers. However, we have not yet seen how everything can be pulled together to construct a full-scale application, including for example the ‘control logic’ running in the background, a web interface for configuration and operation, and a simple control panel using a display and buttons.

Messaging between applications

Figure 1 shows a typical scenario. In the middle we have the Linux system running the custom application and a web server to allow control from a wide range of devices using a browser. We can also add a display and a couple of buttons, the display being controlled using a separate program. The main application program is just an ordinary process, running in the background among many others. Such a program is known in the Linux world as a ‘daemon’ or ‘service’. Once a daemon has been started it continues to oper-

ate without further action on the part of the user. There are various ways to implement the interface between the daemon and other programs: such messaging between threads or processes is called ‘inter-process communication’, or IPC. For example, we can use pipes, FIFOs, shared memory, sockets, stdin and stdout (‘standard input’ and ‘standard output’), files, databases or any number of auxiliary programs. Which approach we choose depends on several factors. Are the programs written in the same language? Does the software have to be portable to other operating systems? What data rate is required? One widely-used approach is the classical network ‘socket’. Sockets have the significant advantage that data and commands can be sent over a network if desired. It is very easy to send and receive messages using sockets and suitable functions and classes are available in a wide variety of different programming languages. We have therefore chosen to use sockets as the basis for communications between the pro-

grams described in this article. Since the web is also based on standard sockets, we will also see along the way how to access data on the internet directly. So, as a first step, we will equip the Linux board with an internet connection.

Connecting to the Internet

We assume that you already have a router providing a connection to the internet and acting as a gateway for the other devices on your network. You will also need to connect a USB-to-LAN adaptor to the USB interface on the board and make sure that the driver for it is loaded [2].

To integrate a new device into a network it needs to know its IP address and other parameters such as the gateway and DNS server addresses. This can be done manually or by using DHCP ('Dynamic Host Configuration Protocol'). When DHCP is used the router on the local network will configure the Elektor Linux board automatically (see **Figure 2**). To test whether an IP address has been successfully assigned using DHCP, you can use the `ping` command to see whether you can establish contact with (for example) the web server at `www.elektor.com` (see **Figure 3**).

If no DHCP server is available the network parameters have to be entered manually.

IP address

To set the board's IP address, enter the command

```
ifconfig eth0 <ipaddress>
```

with the placeholder `<ipaddress>` replaced by an unused IP address on your network. To check that this has been successful, you can try pinging your router using the command

```
ping <routeraddress>
```

with the placeholder `<routeraddress>` replaced by the IP address of your router.

Gateway address

In order to send packets over the Internet, the board has to know the address of the router which provides the gateway for network traffic. This is configured using the command

```
route add default gw <routeraddress>
```

with the placeholder `<routeraddress>` again replaced by the IP address of the router. To test

```
root@gnublin:~# udhcpc
udhcpc (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.190...
Lease of 192.168.0.190 obtained, lease time 864000
Setting IP address 192.168.0.190 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.0.1
root@gnublin:~#
```

Figure 2.
Configuration of IP
addresses using DHCP.

```
root@gnublin:~# ping www.elektor.de
PING www.elektor.de (94.236.12.177): 56 data bytes
64 bytes from 94.236.12.177: seq=0 ttl=108 time=89.798 ms
64 bytes from 94.236.12.177: seq=1 ttl=108 time=76.505 ms
```

Figure 3.
Testing access to a server
using the 'ping' command.

this step, you can try pinging any other computer on the internet, for example by typing

```
ping 94.236.12.177
```

which will ping the Elektor web server. Assuming this is successful (that is, that no packets are lost), we can proceed to the final step.

DNS server address

In order to navigate the internet using domain names rather than IP addresses, each device on the network needs access to a DNS server. A DNS server can convert a string like 'www.elektor.com' into its corresponding IP address. To configure a DNS server for the Elektor Linux board to use, it is necessary to edit the file '/etc/resolv.conf' (see **Figure 4**). Run the editor by typing

```
nano /etc/resolv.conf
```

and look for the 'nameserver' entry in the file. One option is to enter your router's address here, as most routers these days provide DNS services. Alternatively, if you do not have a local DNS server, it is possible to use one on the internet. One that is easy to remember and ideal for testing is run by Google on the IP address 8.8.8.8. Save the file using control-O and then type control-X to leave the editor.

You should now find that you can reach any server on the internet using the 'ping' command followed by the name of the machine.

Writing your own server

Now that we have established a connection



Figure 4.
Configuration of the DNS
server.

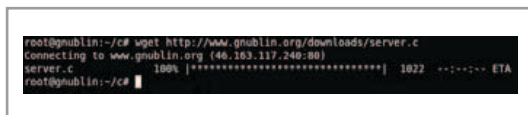


Figure 5.
Downloading the file
'server.c'.

between the board and the internet, we can directly download some example source code for a server. The command 'wget', which will be familiar to users of Linux on PCs, lets us transfer any file we like from the internet onto our board.

Figure 5 shows the command being used to download the server example code.

The server is written in the C programming language. For initial testing we will use the compiler on the board; if you plan to develop the code to any significant extent it is worth using the tool-chain on the development PC.

Listing 1 shows the source code for the program. The basic function of the server is to listen on port 5000. If it receives a connection on this port from another device, it will reply with the current time. The program could be extended to include a command interpreter so that different replies can be given or different actions carried out in response to different commands sent by the client.

To test the server we first need to compile it:

```
gcc server.c
```

When the compiler has finished, we start the server by typing:

```
./a.out
```

The program will now apparently become unresponsive to the console (apart from control-C, which exits the program). However, this is not quite true: you can type control-Z, which suspends the server program and returns you to the

command prompt so that you can execute other programs or commands.

The console will show

```
[1]+  Stopped                  ./a.out
```

and if you now type

```
fg
```

(where 'fg' means 'foreground') the console will become blocked again. Suspend the program again using control-Z and then type

```
bg
```

which will set the server program running again, but this time in the background.

As we mentioned above, the example server code sends the current time to any client that connects to it. We can try this out from the console using the 'telnet' program, which takes as its parameters the IP address of the server and its port:

```
root@gnublin:~/c# telnet 127.0.0.1 5000
```

```
Tue Sep 27 21:34:49 2011
```

```
Connection closed by foreign host
```

To bring the server program back to the foreground (for example in order to terminate it), simply type

```
fg
```

at the console.

Of course, you can equally well access the server from a web browser. The server could also be modified so that instead of simply returning the time it carries out some other action, such as turning an LED on and off. We showed how this can be done in the previous instalment in the series [2].

Remote access over the network

So far we have used 'picocom' or similar terminal program to provide a console over the board's serial port. However, now that we have a network interface, we can run as many consoles as we like over the network. This makes life much easier, especially when developing applications.

Listing 1: A simple server program.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

int main(int argc, char *argv[])
{
    int listenfd = 0, connfd = 0;
    struct sockaddr_in serv_addr;

    char sendBuff[1025];
    time_t ticks;

    listenfd = socket(AF_INET, SOCK_STREAM, 0);
    memset(&serv_addr, '0', sizeof(serv_addr));
    memset(sendBuff, '0', sizeof(sendBuff));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(5000);

    bind(listenfd, (struct sockaddr*)&serv_addr,
        sizeof(serv_addr));

    listen(listenfd, 10);

    while(1)
    {
        connfd = accept(listenfd, (struct sockaddr*)
            NULL, NULL);

        ticks = time(NULL);
        snprintf(sendBuff, sizeof(sendBuff),
            "%.24s\r\n", ctime(&ticks));
        write(connfd, sendBuff, strlen(sendBuff));

        close(connfd);
        sleep(1);
    }
}
```

For example, we can be editing the source code in one console while simultaneously running and testing the program in another.

We use SSH (the 'secure shell') to make the connection to the Elektor Linux board over the network. This console automatically provides an encrypted communication channel to the board, which has the advantage that other devices on the network cannot intercept the information being sent to and from the board.

SSH from Linux

Under Linux 'ssh' is one of the standard tools. Type the following into the console on the host computer:

```
ssh root@192.168.0.190
```

with the IP address replaced by that of your Linux board. When the connection has been established with the board, type the word 'yes' (see **Figure 6**). Shortly the standard console will appear, and you should feel at home!

SSH from Windows

Under Windows we can use the PuTTY program [3]. **Figure 7** shows the program's main window, and again we have to enter the IP

address of the board. Next click on 'Open' and you will be asked to verify that the remote device's 'fingerprint' (part of the SSH key for checking the identity of the remote device). Click on 'Yes'. In response to the prompt 'login as' enter 'root'.

Listing 2: A start-up script.

```
#!/bin/sh

if [ ! -d /var/log/lighttpd ]
then
    mkdir /var/log/lighttpd
    chown -R lighttpd:lighttpd /var/log/lighttpd
    chmod 777 /var/log/lighttpd/
    touch /var/log/lighttpd/error.log
fi

modprobe asix
udhcpc

echo 3 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio3/direction
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value

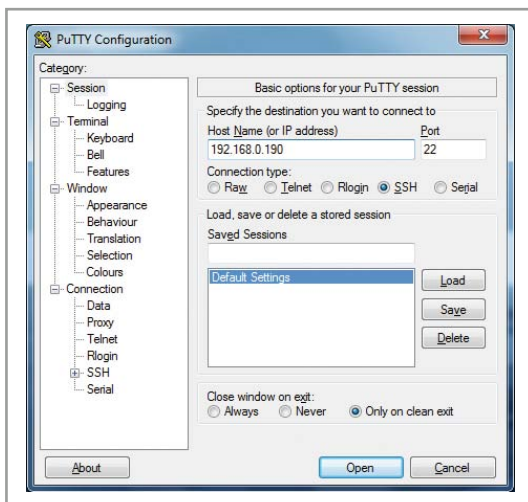
chown lighttpd:lighttpd /dev/lpc313x_adc
echo 1 > /dev/lpc313x_adc

chmod 666 /var/log/lighttpd/error.log
/etc/init.d/lighttpd start
```

Figure 6.
Accessing the board using
SSH under Linux.

```
leuter@hauteur-Inspiron-5380:~$ ssh root@192.168.0.190
The authenticity of host '192.168.0.190 (192.168.0.190)' can't be established.
RSA key fingerprint is bf:7f:52:57:86:59:f8:cb:a8:ae:c3:cb:97:79:36:01.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.190' (RSA) to the list of known hosts.
root@gnublin:~#
```

Figure 7.
Accessing the board using
SSH under Windows.



When you press enter you will be logged in to the system.

Using the Lua scripting language for simple applications

The previous example used the C programming language. C is very commonly used, fast and popular. However, on a Linux system there are many other languages available besides C: each has its advantages and disadvantages and is aimed at different applications.

One very flexible, fast and easy-to-use language is Lua. The language is interpreted, and has become best known for its use in script-

ing games programs. The Lua interpreter is just 120 Kbyte in size but nevertheless implements many of the most up-to-date constructs in programming. Accessing files (which in the Linux world also means accessing devices and hardware functions) is done using the commands `io.open`, `io.read`, `io.write` and `io.close`. The author has written a Lua library for the Elektor Linux board providing easy access to the main hardware functions, including the GPIO and A/D converter.

The advantage of using an interpreted language over a compiled language like C to build a simple control application is huge: you can easily edit your programs directly on the board using its built-in editor (nano or vi) and execute them immediately.

Using our newly-configured internet connection we can download the Lua example programs onto the SD card as follows.

```
mkdir lua
cd lua
wget http://www.gnublin.org/downloads/elektor.lua
wget http://www.gnublin.org/downloads/blink.lua
wget http://www.gnublin.org/downloads/button.lua
wget http://www.gnublin.org/downloads/adc_relay.lua
```

Figure 8 shows the listing of a simple Lua program that flashes the LED. First the Elektor library has to be loaded. The LED is initialised with a call to `initLED()` and then switched on and off in an infinite loop.

The program can be run from the console as follows:

```
lua blink.lua
```

The program 'button.lua' is a simple demonstration of how to control a relay using the IO15 button on the board. And **Figure 9** shows the code of the example program 'adc_relay.lua', which reads voltage values from the A/D converter in an infinite loop. If the reading is above the threshold of 500 the relay contacts are closed; otherwise they are opened. The program can be tested by connecting a potentiometer to the analog input as described in the previous instalment in this series [2].

There is plenty of information about Lua on the internet [4] as well as some good books [5].

```
require "elektor"

-- initialize hardware
initLED()

-- start main loop
print("blinking ...")

while 1 do
    setLED()
    wait(1)
    clearLED()
    wait(1)
end
```

Figure 8.
The "blink.lua" example
program.

```
require "elektor"

-- initialize hardware
initButton()
initLED()
initRelay()
initADC(1)

-- start main loop
print("Please tune the resistor")

while 1 do
    value = getADC()
    print(value)
    if value > 500 then setRelay() else clearRelay() end
    wait(1)
end
```

Figure 9.
The "adc_relay.lua" example
program.

Starting the web server and other programs automatically

It is rather inconvenient to have to restart a server application (and perhaps also reconfigure the hardware) manually each time the board is rebooted. The solution to this is to include the relevant commands in a start-up script. **Listing 2** shows a suitable script to start up the web server application from the previous instalments in this series [2]. The file 'start.sh' can be downloaded from the internet (see **Figure 10**).

In order to have this script run automatically when the Linux system is booted, we can for example add an entry in the file '/etc/rcS.d/S55bootmisc.sh' (the directory '/etc/rcS.d' is where the system start-up scripts are kept). First open the file using an editor.

```
nano /etc/rcS.d/S55bootmisc.sh
```

At the end of the file you will see the following entries.

```
/bin/mkdir /var/run/sshd  
/usr/sbin/sshd
```

After these add the line

```
/home/root/start.sh
```

and then save the file by pressing control-O and exit the editor with control-X. To make sure that the file 'start.sh' can be found it must be placed in the directory '/root'. The file must also be made executable by setting its 'x' flag as follows.

```
chmod +x /home/root/start.sh
```

From now on whenever the board is booted it will automatically establish a network connection and start up the web server.

Access from a smartphone or tablet

In the previous instalment in the series we showed how to access a demonstration web application from a browser running on a PC. Of course, such an application can equally well be accessed from a smartphone or tablet, as long as the network on which the *Elektor* Linux board sits includes a WLAN access point to which the mobile device can connect. **Figure 11** shows the HTML control interface running on an iPhone. The



Figure 10.
Downloading the web server start-up script.



Figure 11.
Accessing the board from an iPhone.

LED on the board can be turned on and off by touching the HTML button.

At this point it is useful to note that the Gnublin project community wiki [6] covers a very wide range of hardware and software topics. It includes guides to transmitting and receiving data over the I²C, SPI and UART interfaces (including moving data to and from user space) and much more besides.

We have invited readers to propose a topic for the next instalment in this series: at the time of going to press our poll was still open. We have been eagerly awaiting your suggestions! We can also announce that we are working on an expansion board that increases the range of interfaces available: more on that also in the next issue.

(120578)

Internet Links

- [1] sauter@embedded-projects.net
- [2] www.elektor.com/120182
- [3] www.putty.org
- [4] www.lua.org
- [5] www.lua.org/pil
- [6] <http://wiki.gnublin.org>

Arduino on Course (3b)

Part 3b: Sensors talk to Android phones

By **David Cuartielles**
(Spain)
and
Andreas Goransson
(Sweden)

Last month we introduced a series of tools for working with Android and Arduino Mega ADK. This month we go one step further in that process, showing you how to read sensors connected to Arduino, into the phone. With all development tools in place, running code in your phone and making it talk to your Arduino will be very easy, you can start by following the examples coming with the ADK libraries!



Android devices are about everywhere. There are many ways how to approach the creation of applications for those. In our case, we will continue exploring the use of Processing [1] as a way to write, compile and upload code for/to Android phones.

Materials

Continuing with the experiments performed last month, the materials for making the upcoming experiment are going to be:

- Arduino Mega ADK board or Arduino Uno board with USB Host Shield
- Android phone with Android OS 2.3.4 or newer (the newer OS the better)
- USB cable and microUSB cable
- TinkerKit Shield (this is optional, you could use a breadboard and jumper wires)
- TinkerKit Modules: Potentiometer and Push-button (also optional)

Note: in this article you really need to have an

Android phone to test how things work. We have made an extra effort in making sure the code is compatible with the latest version of Android OS, at the time of writing it was version 4.1, code Jelly Bean.

Tip: download the code shown in this article from Elektor's website, as we are only showing excerpts in the article.

Where we are now

At this point you have a certain understanding of Google's Accessory mode and how accessories tell Android devices to boot a certain app when connecting. You are familiar with Processing, the Java learning tool, and have created and uploaded small apps to your phone or tablet.

The next step is moving into creating new apps that will take advantage of the connectivity between Arduino — as an accessory — and your Android.

The USB Host Library

The Android Open Accessory Protocol (AOAP) is what determines how your Android device and your Arduino board can talk to each other. The Arduino board is technically a USB Host, while the phone becomes a USB Client. Android devices, like most of the state-of-the-art devices out there, carry a peripheral that performs the so-called USB On The Go functionality. According to it, the same USB connector (currently micro USB connector) can become a USB client (like a mouse or a keyboard) or a USB Host (like the one in your Personal Computer).

The AOAP establishes that accessories act as hosts while the Android devices act as clients. This allows the host feed the client with electricity (your Arduino board will be re-charging your phone once plugged), among other features.

To establish the communication between the two, you saw in the previous article that you need to run the USB Host library as part of your Arduino code. You could check one of the examples to the library just to confirm you have it installed. *Note:* use the menu to navigate through: “File / Examples / USB Host”, if you didn’t find those examples there, it would mean that you hadn’t installed the USB Host library in Arduino. Check the downloads for this article [3] and install it; you are going to need it. **Figure 1** shows the hardware involved.

Experimental setup: the processing side

Let’s first look into the software you need to program your Android applications:

- Android SDK with all the updates up to the latest version of the APIs
- Processing 2.0b3 or newer (if you read the first article, you will see we have updated to the latest version, you should do the same at this point)
- The Arduino ADK Tool for Processing
- The Arduino ADK USB Library for Processing

The last item in that list is what you will be missing if you experimented with the previous article in the series. That one was omitted intentionally back then.

You need the complementary library to USB Host on the Processing side to make sure you can get your board to talk to your phone. We call that library “Arduino ADK USB”. The following section explains how to install it; **Figure 2** shows the hardware side of things.

Installing the Arduino ADK USB Library for Android

Inside the download archive for this article you will find a folder named “Processing” which will have a subfolder called ‘libraries’. You should copy the contents of that folder inside your Processing libraries folder. The latter is located inside your Processing sketchbook folder, usually stored at ‘My Documents/Processing’ in Win-



Figure 1.
Arduino Mega ADK with
Tinker Kit Shield and phone.

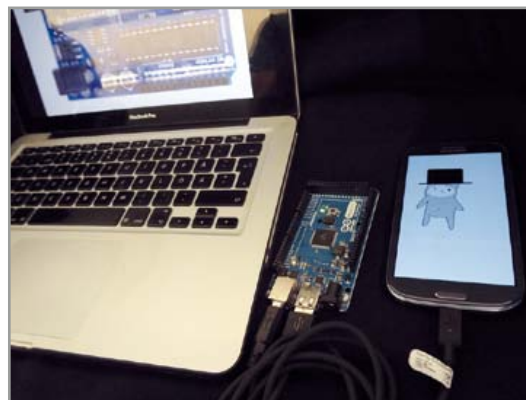


Figure 2.
Arduino Mega ADK
connected to a PC and to a
phone.

dows computers or “Documents/Processing” in Mac and Linux.

This library comes with four examples. We recommend you start your accessories from one of those examples, as it will make your life much easier. Let’s start by trying out the example called ‘digitalWrite’. In this case you will be connecting your Arduino board to the phone and when touching the screen, an LED connected to your board will go on. If you stop touching the screen, the LED will go off.

DigitalWrite

Each example has two parts. The first one runs on the Arduino board, it handles the communication to and from the phone, while performing actions on the other pins on the board. The second part is code to be compiled in Processing and that will run inside the Android device. This app will be reading and writing data to the phone’s cable connection as well as performing some actions on the screen or reading any of the phone’s sensors like camera, accelerometer, etc. Let’s first analyse the Arduino code, open your Arduino IDE and type in the code at **Listing 1**.

Listing 1. DigitalWrite example from the Arduino's USB Host library.

```
#include <AndroidAccessory.h>

// accessory descriptor. It's how Arduino identifies itself to Android
char accessoryName[] = "DigitalWrite"; // your Arduino board
char companyName[] = "Arduino SA";

// led variables
int ledPin = 10;

// counters
long timer = millis();

// initialize the accessory:
AndroidAccessory usb(companyName, accessoryName);

void setup() {
    // start the connection to the device over the USB host:
    usb.begin();
    // configure the LED pin as output
```

Remember to upload it to the board once you are done!

This code example will be listening to the USB port. Once the phone connects to the Arduino board, the `usb.isConnected()` function will

return an affirmative answer. When the phone starts sending data, `usb.available()` will return a value different from zero that Arduino will then read and store in the variable `val`. Something very similar to what's pictured in **Figure 3** should happen at your place too.

Listing 2. Processing sketch that will detect screen touches.

```
import cc.arduino.*;

// create the ADK object
ArduinoAdkUsb arduino;

void setup() {
    // Lock PORTRAIT view
    orientation( PORTRAIT );
    // initialize the ADK object
    arduino = new ArduinoAdkUsb( this );
    if ( arduino.list() != null )
        arduino.connect( arduino.list()[0] );
}

void draw() {
    // Draws a filled rect based on arduino connection state
    connected( arduino.isConnected() );
}
```



```
pinMode(ledPin, OUTPUT);
}

void loop() {
  // print to USB 10 times per second
  if(millis()-timer>100) {
    if (usb.isConnected()) { // is the USB connection open?
      if (usb.available() > 0) { // is there data?
        char val = usb.read();
        // 'a' turns the LED on, 'b' off
        if( val == 'a' )
          digitalWrite( ledPin, HIGH );
        else if( val == 'b' )
          digitalWrite( ledPin, LOW );
      }
      timer = millis();
    }
  }
}
```

On the Processing side you will need to write a program that will check for the event of someone touching the screen. If there is a connection over the cable, the phone will then send 'a' to indicate someone just started touching the screen. It will send 'b' to indicate the finger was

just lifted. We are detecting the events 'start' and 'stop touching' as we don't want to flood the communication port.

Open your Processing IDE and search for the `adk_digitalWrite` example. You will find it at the menu 'Examples / Contributed Libraries /

```
public boolean surfaceTouchEvent(MotionEvent event) {
  if ( arduino.isConnected() ) {
    if ( event.getAction() == MotionEvent.ACTION_DOWN )
      arduino.write('a');
    else if ( event.getAction() == MotionEvent.ACTION_UP )
      arduino.write('b');
  }

  // if you want the variables for motionX/motionY,
  // mouseX/mouseY etc.
  // to work properly, you'll need to call
  // super.surfaceTouchEvent().
  return super.surfaceTouchEvent(event);
}

[...]
```

ArduinoADKUsb' if you installed the library properly, see **Listing 2**.

The code in **Listing 2** is mostly self-explanatory. One part you might find a little upsetting is the method definition `surfaceTouchEvent (Motion-Event event)`. Java is a very powerful programming language and in this case it allows overriding some of the existing callback methods. Whenever you touch the Android's screen, the system is configured to call that specific function (`surfaceTouchEvent (MotionEvent event)`). If you override it in your program (basically, if you rewrite it) you can make it do whatever you want to.

In this case, we detect two of the many system events: `MotionEvent.ACTION_DOWN` (the equivalent to start pressing the screen) to send 'a' to Arduino and `MotionEvent.ACTION_UP` (the same for stop pressing the screen) to send 'b'.

A bit more fun

Once you have got the first example to run... why not building something a little more fun? The Internet is packed with cats, that's why we decided to create the `Elektor_Miau` application last month. We wanted to make a fun comment and, at the same time, show you that it is possible to start making things more appealing than the classic "Hello World" example when learning programming.

This month we want to build upon the story and encourage you to learn how to make small animations using Processing. You can look for any animated GIF on the internet and use it to replicate what you will see in this example. The only condition is that you need the animated gif exported as frames. You should name them in the following way: `s00.gif`, `s01.gif`, `s02.gif` ... You are of course welcome to use the same image we are using for our article. We got an illustrator to draw for us a small rabbit appearing out of a hat. The goal of this example is making a small interactive game where you will use sensors (a button and a potentiometer) to interact with the small rabbit on the phone's screen, like in **Figure 4**.

This time it will best to start looking at the code to be running inside the phone. The first step is building an animation inside Processing. If you are versed in Android, you can try porting the

GIFdecoder library we used last month to play animated GIFs inside your Android app. As we want to keep the programming simple to understand, the way we recommend you animate your Rabbit is very simple: you should make an array of images designed to contain the frames you want to be part of the animation. This concept is based in the Processing example `Animated Sprite` by James Paterson [2], see **Listing 3** at [3].

Note: Listing 3 is not showing the `Animation` class that will open all the images named `s00.gif`, `s01.gif`, etc. We recommend you check the full code listing coming with the example `Elektor_Simple_Animation`. You will also see a folder called 'data' inside the Processing sketch folder containing all the images that are part of the animation (54 in total).

The ultimate setup

Next up, we'll modify the example to include the library handling the USB communication. We recommend you merge the previous code with the `adk_analogRead` example. The result will look like the code in **Listing 4** you can find at [3]. There are a couple of things in the example that need to be highlighted. First there is an array called `delays[i]`. It's responsible for controlling the time each frame will be displayed in the animation. Every time you load a new frame, you also change the timer length in the condition `millis() - timer > delays[index]`. You are welcome to experiment with the numbers in that array to see how it affects the animation. Remember there have to be as many numbers as frames in the animation.

Also there is a `keyFrames[i]` array that holds a list of frames in the animation that will be expecting an action from the user for the movie to continue. In this case we are going to detect button presses. The animation will stop at those frames until the user presses the button attached to the TinkerKit Shield as seen in **Figure 5**.

If you look at the line checking how many bytes arrived from the Arduino board, we check `arduino.available() >= 3` to see whether we got 3 bytes. The information is encoded as follows:

- Byte 0: has to be 255, it's a marker we use to distinguish when the block starts;

- Byte 1: is the potentiometer's value; range 0 to 254;
- Byte 2: comes from the button; it will be either 0 or 1.

The potentiometer's value will be used to change the screen background colour, while the button will trigger the next sequence of the animation.

Note: use the Arduino ADK Usb tool (menu 'Tools / Arduino ADK' in your Processing IDE) to upload this program from to your phone. A dialog window will pop up. In the text field labelled "model", enter the name of the app you call inside Arduino's IDE. The last parameter is used to select the version of the SDK for your accessory type. If your phone or tablet runs Android 3.0 or later, you should check SDK v12; SDK v10 is only for those devices running Android 2.3.4. In the unfortunate case your Android device was running OS version 2.3.3 or earlier, you wouldn't be able of using the Accessory mode.

Finally, the Arduino code is practically a repeat of the `adk_analogRead` example adding the extra marker byte, limiting the analog reading to 254 and adding the button. The loop for that program is found in **Listing 5** at [3]. Please check the full program inside the documentation you can get from Elektor's website [1], in the folder "Arduino / code".

Closing words

With this article we finish our introduction to Android and how it connects to Arduino. You should be ready to create new projects, build robots where your phone can be the intelligence, or integrate sensors that aren't coming with your Android device. If you make any cool projects using any of our code don't hesitate to drop us a line at blog@arduino.cc, we will be happy to let the rest of the community know about them.

(120573)

Internet Links & References

- [1] Processing Project: <http://processing.org>
- [2] Animations using multiple images with Processing: <http://processing.org/learning/topics/sequential.html>
- [3] www.elektor.com/120573



Figure 3.
The LED is on when touching the screen.

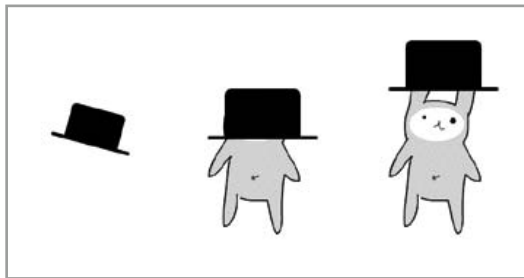


Figure 4.
A couple of frames of our rabbit movie.

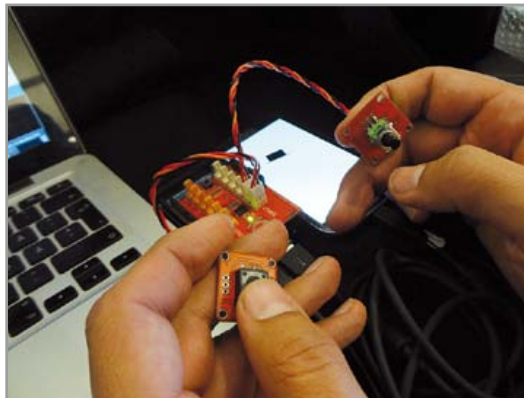


Figure 5.
Button and potentiometer on top of the TinkerKit Shield.

Acknowledgements

The Arduino design team expresses its thanks to the team at Circuits@Home for their first work on the USB Host Library for Arduino and Philip Lindsay for his work on debugging the initial collection of official Arduino Libraries for Android. Also to Rodrigo Calvo who helped unveiling how to improve the USB Host library to work with Android 4.1 devices. And Laura Balboa for her illustration job.

Electronics for Starters

(10)

Radio Frequency (RF)

By Burkhard Kainka
(Germany)

It all started with high frequencies: radio technology was the driving force of electronics. At first there were electron tubes and AM radio, later followed by VHF and FM. Now we work with semiconductor devices, and the frequencies just keep on rising. Nevertheless, RF projects for the medium- and short-wave bands are still extremely interesting, especially for starters.

Building your own radio is still a good way to get started in electronics. However, your first radio project should be very simple, with no special components, and as foolproof as possible. The wideband regenerative receiver shown in

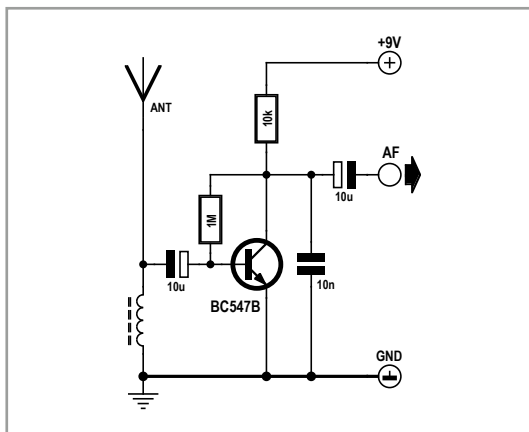


Figure 1.
A wideband regenerative receiver.

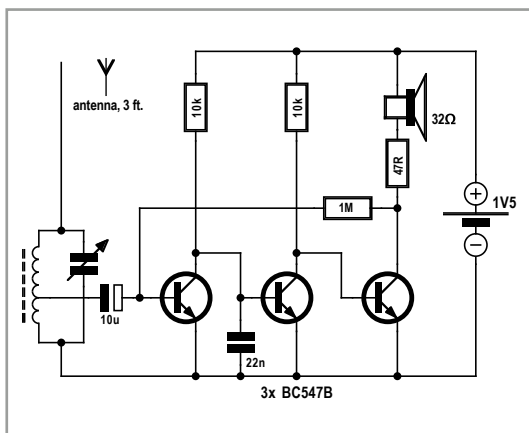


Figure 2.
A regenerative receiver with higher gain and better selectivity.

Figure 1 fits the bill. A regenerative receiver is a circuit that amplifies HF signals and demodulates them at the same time. With this extremely simple receiver you can listen to everything on the ether, whether it be a medium-wave broadcasting station or a very distant short-wave transmitter – and of course all possible types of HF noise in the near vicinity. The only prerequisite is a sufficiently long wire antenna. Around 30 feet of wire in the open air is ideal, but even shorter antennas can still yield good results. You also need an earth connection. A sensitive headset can be connected to the audio output, or better yet, a loudspeaker amplifier.

The wideband regenerative receiver picks up everything at the same time. You will usually hear several transmitter signals, although most of the time one of them will be distinctly louder. The frequency range of the receiver is partially determined by the coil you use. If you wish, you can also use a fixed inductor. For medium wave reception the coil should have an inductance of around 300 μH , and for short wave reception approximately 10 μH , or as little as 3 μH if you want to listen to the higher-frequency bands in the 15 to 17 MHz range. A DIY coil for medium wave reception could consist of 100 turns on a ferrite rod. For short wave reception a coil with 10 to 30 turns and a diameter of 10 mm, without a ferrite core, is enough. However, the results also depend on the antenna.

It's a good idea to try several types of coil or inductor. A wide variety of stations can be heard,

depending on the receiving site, the time of day and the coil that you use. For example, you may sometimes be able to listen to Radio China with this simple receiver, since the range is especially great in the higher-frequency short wave bands. If you take a good look at the circuit, you will see that it consists of a single amplifier stage. This means that the high frequency signal from the antenna is amplified. With a truly ideal amplifier you wouldn't be able to hear anything, since the signal frequencies are far above the audible range. The reason that you do hear something is that the transistor has a non-linear input characteristic. As a result, the positive half-waves of the HF signal increase the collector current more than the negative half-waves reduce the current. This means that the average current varies according to the amplitude of the HF input signal. And because the transmitter modulates the amplitude of the HF signal at the frequency of the audio signal, the demodulated audio signal appears at the output of the receiver.

The capacitors in the circuit are also important. The relatively large capacitor at the input shorts audio signals on the base of the transistor to ground via the input coil, so that only HF signals are present at the base. The current in the collector circuit consists of a combination of the amplified HF signal and the audio signal. The 10 nF capacitor at the output shorts the HF signal to ground, leaving the audible AC signal in the audio range. The end result is a simple, low-cost receiver with fairly good sensitivity.

Figure 2 shows a simple radio with the same sort of receiver stage, but with more gain and better selectivity. It has a tunable resonant circuit at the input. Depending on the coil characteristics, this radio can receive signals over a range extending from the long-wave band up to the upper short-wave region. The receiver coil is tapped at approximately 10% of its length. This keeps the damping factor of the resonant circuit relatively low, which means it has a good Q (quality) factor (see the 'Resonant Circuits' inset). This tuned circuit allows a specific broadcast signal to be selected. With a total of three transistors and a supply voltage of just 1.5 V, this circuit works well as a headphone receiver. Incidentally, this circuit has a certain similarity to the TAPIR HF Sniffer published in the July & August 2012 issue of *Elektor*, which also has three transistors but operates as a wideband regenerative receiver.

Resonant Circuits

If you combine an inductance (L) with a capacitance (C) you get a resonant circuit. In a manner similar to mechanical energy in a pendulum, electrical energy can flow back and forth between the inductor and the capacitor at a specific frequency (f), which is called the resonant frequency. A resonant circuit oscillates after being excited by any short pulse of electricity. An ideal resonant circuit would have no losses and would therefore oscillate forever after being excited.

However, in practice the oscillation is damped because energy is dissipated in the form of heat due to the resistance of coil winding, magnetic losses in the core of the inductor, and electromagnetic radiation. For the sake of simplicity all of these losses can be collectively represented by a resistance R in parallel with the circuit. The resonant frequency f is given by the formula:

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

A quality factor (Q factor) can be specified for every resonant circuit. Q is determined by the ratio of the parallel damping resistance R to the inductive impedance $Z_L = 2 \pi f L$ or the capacitive impedance $Z_C = 1 / (2 \pi f C)$ at the resonant frequency, at which the impedance is resistive:

$$Q = R / Z_L \text{ or } Q = R / Z_C$$

If a resonant circuit is excited by a variable-frequency signal from an AC signal generator with high internal impedance, the voltage across the resonant circuit is the highest at the resonant frequency. The amplitude of the voltage at the resonant frequency is directly proportional to the Q factor, or in other words, the lower the damping due to energy dissipation of any form, the higher the resonant voltage. At either side of the resonant frequency it is possible to find points on the resonance curve where the voltage is reduced by a factor of 0.707 ($1 / \sqrt{2}$), or 3 dB down from the maximum. The frequency difference between these two points is designated the bandwidth b of the resonant circuit. The relationship between the resonant frequency f , the bandwidth b and the Q factor of the circuit is given by the formula $b = f / Q$.

With careful coil construction it is usually possible to attain Q factors up to 100 or so. However, resonant circuits are also damped by the connected circuit or connected antenna. This damping can be countered by using a small coupling winding, a coil tap or a suitable capacitor to achieve loose coupling. If the resonant circuit is connected directly to a gain stage, this stage should have high internal impedance to keep the damping small.

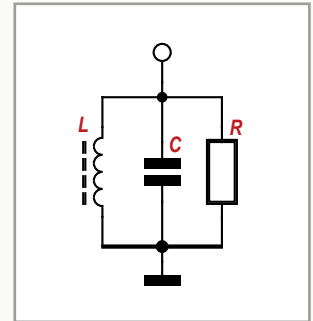


Figure 3.
A tuned 'spark gap' transmitter.

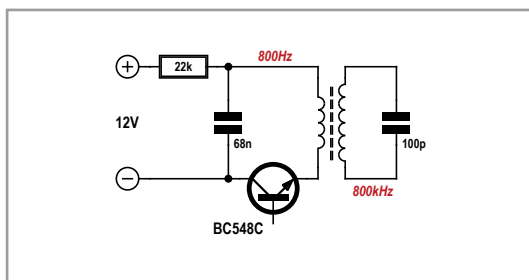


Figure 4.
A sine wave oscillator with inductive coupling.

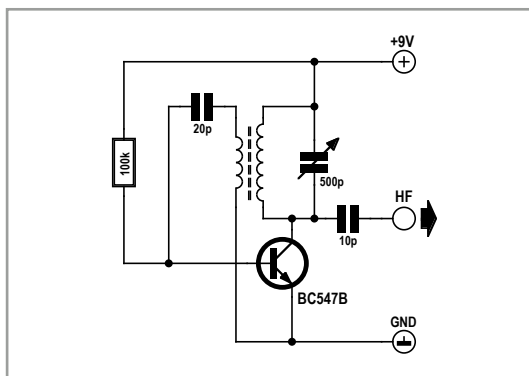


Figure 5.
An oscillator for use with low supply voltage.

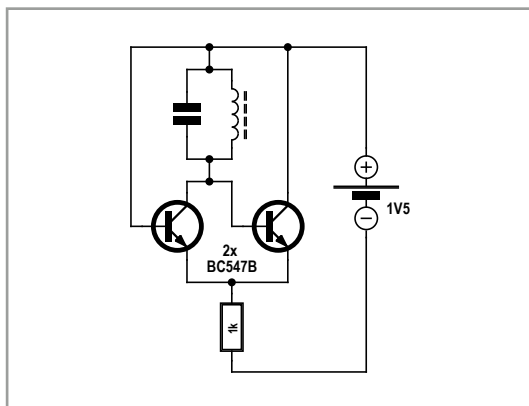
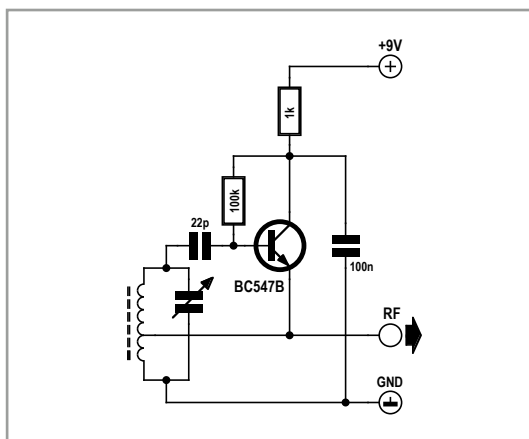


Figure 6.
An oscillator with a tapped coil.



Generating HF signals

The world's first transmitter was a spark-gap device. To see how it works, all you need is an ordinary light switch. When you flip the switch, you can hear a bit of static on the radio — at least if you're listening to the medium wave band and you aren't receiving a signal from a strong AM transmitter. A pulse waveform with steep edges always has signal components in the RF range. In fact, in many places switch-mode power supplies and other electrical devices are a major source of radio interference, especially in the medium wave band. This means that the difficulty is not generating RF signals, but instead preventing the radiation of undesirable RF signals.

You don't really need a high-power transmitter suitable for a pirate radio station, but a signal generator that produces just a wee bit of RF power can come in handy now and then, if only for testing your own receivers. **Figure 3** shows a tiny transmitter circuit that works on the same principle as the old-fashioned spark-gap transmitter. In this case the periodic pulses at a frequency of around 800 kHz result from switching an NPN transistor in an astable multivibrator circuit (see instalment #7 of this series in the September 2012 issue of *Elektor* [2]). Each steep pulse edge excites oscillations in the resonant circuit, which then quickly decay as damped oscillations. The transmit frequency is determined by the resonant circuit. For example, if you use a ferrite rod antenna from an old radio, the frequency will be somewhere in the medium wave band, such as 800 kHz. You will hear a humming sound if you hold a radio close to the ferrite rod. All you need now for a full-fledged transmitter is a Morse key. The shipboard transmitters of days gone by worked in a similar manner, but with considerably more power and at much lower frequencies.

LC oscillators

Spark-gap oscillators could only transmit damped oscillations. The first real progress came with the invention of electron tubes, which were able to generate undamped oscillations. In the previous instalment of this series, which dealt with sine wave oscillators [3], we described what you need for this: enough gain and suitable feedback.

The grandfather of all RF oscillators is the Armstrong or Meissner oscillator, in which a

coupling coil provides feedback with the right phase (**Figure 4**). The first radio transmitters were also built this way: take a tube and a resonant circuit and add a bit of feedback, and you've got an RF power oscillator. Modulation was handled very simply by fitting a carbon microphone in the antenna lead (!).

The circuit diagram in Figure 4 shows a single-transistor Armstrong oscillator. It's not difficult to get this circuit to oscillate. If you have problems, the phase may be wrong, in which case all you need to do is to reverse leads of the coupling coil. The frequency of this simple oscillator can be adjusted with the variable capacitor. The amplitude depends on a lot of factors, including the coil damping, the feedback winding, and the coupling capacitor. Sometimes it takes a bit of experimentation before you get stable oscillation, since this circuit has a peculiar problem: it is prone to intermittent oscillation if the amplitude is too great. This is caused by rectification of the HF signal by the base-emitter diode, which drives the base negative until the transistor is cut off and oscillation suddenly stops. After a few microseconds the base resistor restores the normal operating point and the cycle starts over again.

This problem does not occur with the oscillator shown in **Figure 5**, since it has automatic amplitude stabilization. Furthermore, this circuit only needs a simple coil without a coupling winding or a tap, and if the coil doesn't even need to have an especially high *Q* factor, thanks to the high gain. This circuit is therefore a good choice for DIY construction, and it always oscillates reliably. Here the feedback is provided by a pair of transistors. Both transistors also operate at a very low voltage (approximately 0.6 V). The circuit can operate with a supply voltage as low as 1 V and can also operate with very low current consumption. This can be advantageous for some applications.

The basic circuit can operate over a wide frequency range extending from audio to UHF, depending on the resonant circuit, making it ideal for building small test oscillators or the like with little effort or expense. However, it does have a drawback: the internal capacitances of the transistors affect the frequency, and they also depend on the operating voltage. This means that although the circuit is very simple, it's not exactly a shining example of frequency stability.

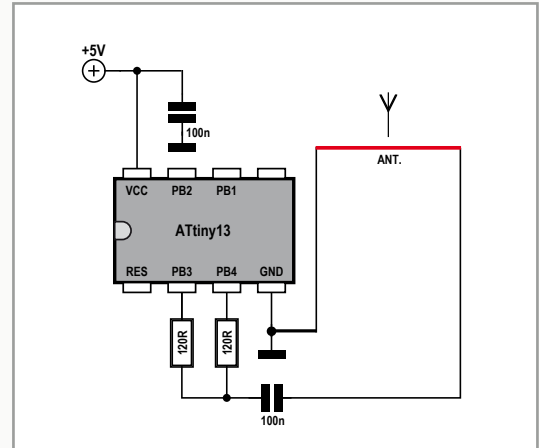
An AM signal generator

Anyone who designs or builds radio receivers can make good use of a small AM signal generator. With a suitable signal waveform, you don't even need to be able to adjust the frequency. This works if the generator operates at an adequately low frequency and generates enough harmonics. The

ATTiny13 AM signal generator described here produces short pulses at a frequency of 70 kHz. This results in strong harmonics covering the entire long-wave and medium-wave spectrum. The pulse train is also briefly interrupted at regular intervals to produce amplitude modulation at a frequency of 750 Hz. A radio can therefore receive an AM test signal at 70 kHz, 140 kHz, 210 kHz, etc. and demodulate it to produce a humming tone.

A wire loop with a diameter of about 4 inches (10 cm) or so makes a suitable antenna. It generates an AC magnetic field that couples directly to the ferrite rod of the receiver. The signal generator can also be used for comparative sensitive measurements by testing how far away from the generator the signal can still be heard. A good receiver can receive a clear signal at distances up to 3 feet.

The source code can be downloaded from the web page for this article [4].



```
'ATTiny13 AM Generator
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4

Config Portb = Output
Dim N As Byte

Do
  For N = 1 To 50    '70 kHz
    Portb = 255
    Portb = 0
  Next N

  For N = 1 To 50    'AM 750 Hz
    nop
    nop
  Next N
Loop

End
```

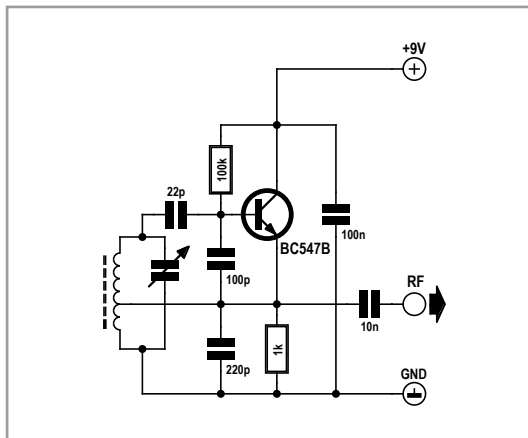


Figure 7.
An oscillator with a
capacitive voltage divider.

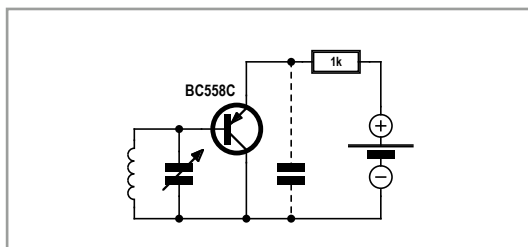


Figure 8.
A DC coupled oscillator
in common-collector
configuration.

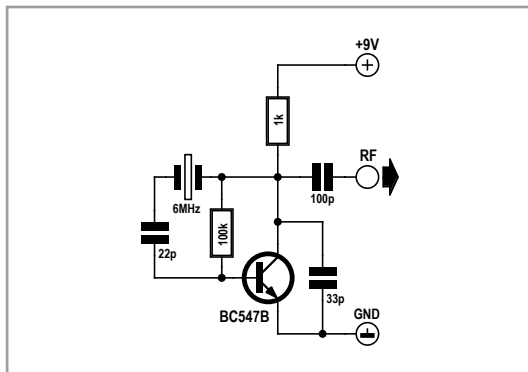


Figure 9.
A crystal oscillator.

Maybe you can think of some way to remedy this — such as wireless battery voltage monitoring?

Another way to obtain the right phase relationship in an oscillator is to use a tapped coil, as illustrated in **Figure 6**. Alternatively you can use a capacitive voltage divider (**Figure 7**). The circuit can be simplified considerably by using a PNP transistor as an emitter follower (common-collector circuit) with DC coupling, as shown in **Figure 8**. With a sufficiently damped resonant circuit, the capacitive voltage divider shown in Figure 7 can actually be omitted because the internal capacitances of the transistor perform the same role. In addition, this circuit can also operate with a low supply voltage, such as 1.5 V.

However, direct coupling to the resonant circuit affects the frequency in this circuit as well, since the voltage divider capacitances vary with the operating voltage. This simple oscillator also highlights a stability problem that may be encountered with amplification stages using a common-collector configuration. Although your intention may simply be to use an emitter follower as an impedance converter, it can inadvertently turn out to be an oscillator. Whether the circuit acts as an amplifier or as an oscillator depends on several factors, including the operating point and the damping of the resonant circuit.

Crystal oscillators

If you need a really stable frequency, you should use a crystal oscillator. Quartz crystals have essentially the same characteristics as a resonant circuit with an extremely high Q factor. They can therefore be used to build relatively simple oscillators with good frequency stability. A typical circuit is shown in **Figure 9**.

If you like to experiment with DIY medium-wave radios, you may wish to have a modulated oscillator for test purposes. A test transmitter of this sort is not connected to an antenna and therefore does not disturb your neighbours. Instead, the RF energy is transferred directly from one coil to the other.

However, the oscillator needs to be stable, as otherwise there's little point to such a modulated signal generator for medium-wave frequencies, but crystals for these relatively low frequencies are very expensive. For this reason, ceramic resonators are often used in such cases. Resonators with rated frequencies of 500 kHz or 2 MHz are commonly available, but these frequencies are outside the medium wave band. However, resonators are also available with frequencies that fit better. For example, we found a ceramic resonator with a frequency of 986 kHz in a discarded television remote control.

Figure 10 shows the circuit of a complete AM test oscillator. The trimmer can be used for fine tuning. If a broadcast station (most likely weak) can still be heard in the background, you can simply tune the oscillator to a beat frequency that neutralizes the station, such as 981 kHz. The small ferrite coil in the transmitter couples directly to the ferrite rod in the receiver. Incidentally, with

relatively minor alterations you can also use this circuit in the short wave region, for example at a frequency of 13.56 MHz, which can be used for experimental purposes without a license.

The modulator stage is designed as an emitter follower that modulates the supply voltage of the output amplifier stage. As monaural transmission is still standard in the medium- and short-wave bands, the signals from the left and right stereo channels are combined at the audio input. The potentiometer should be adjusted for the lowest distortion and the best sound quality. The RF amplifier stage has been intentionally kept modest, since high power is not the objective here.

Short wave regenerative receiver with feedback

The three-stage short wave receiver shown in **Figure 11** has an additional control for the feedback. The first stage is essentially an oscillator corresponding to the circuit in Figure 8. However, its operating point can be adjusted as desired to vary the gain. Here the trick is to adjust the gain to just balance out the losses of the resonant circuit, so that the circuit is just on the edge of oscillation. At this point the receiver has the highest gain and the best selectivity. The PNP oscillator stage in the common-collector configuration is also the regenerative stage that demodulates the HF signal. The two following audio stages generate enough power to allow a small loudspeaker to be used.

Whether the regenerative stage can be adjusted to the edge of oscillation depends in part on how much the input circuit is damped by the connected antenna. For this reason there are two options for connecting the antenna. Connection through a small coupling capacitor results in loose coupling and low damping. By contrast, direct connection is suitable for very short antennas, since a relatively long antenna also radiates HF energy, thereby damping the resonant circuit. With the right setting, extremely high sensitivity can be achieved with a regenerative receiver. Such receivers were standard equipment in the early days of radio engineering. Even with weak transmitters, they achieved ranges of several thousand miles. You can still experience the fascination of this circuit today, since it allows outstanding receivers to be built with very low cost and effort.

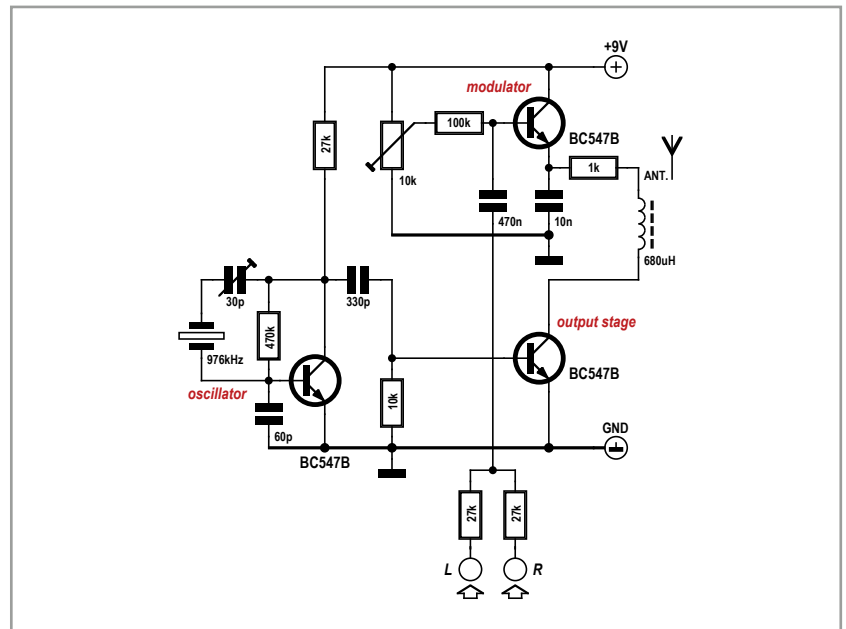


Figure 10.
A modulated medium-wave
AM signal generator.

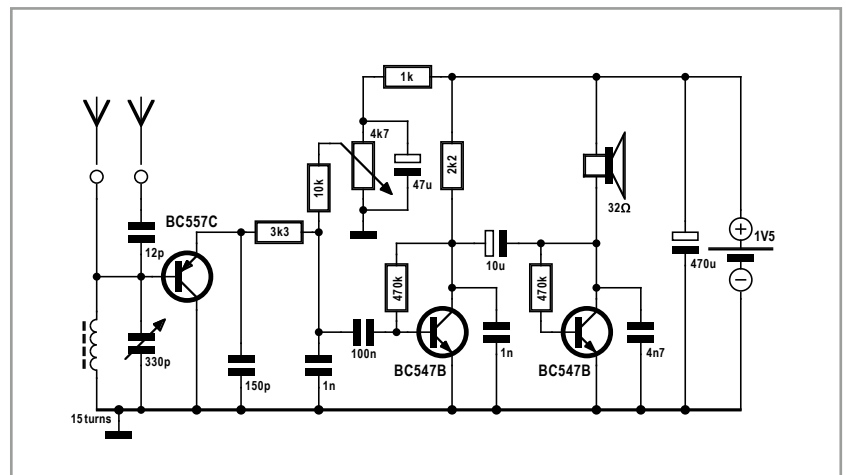
This brings us to the end of the *Electronics for Starters* series. However, we intend to continue presenting articles on basic aspects of electronics at irregular intervals in upcoming issues of the magazine. Some of the topics on the agenda are operational amplifiers and commonly used digital ICs. Stay tuned for more!

(120010)

Internet Links

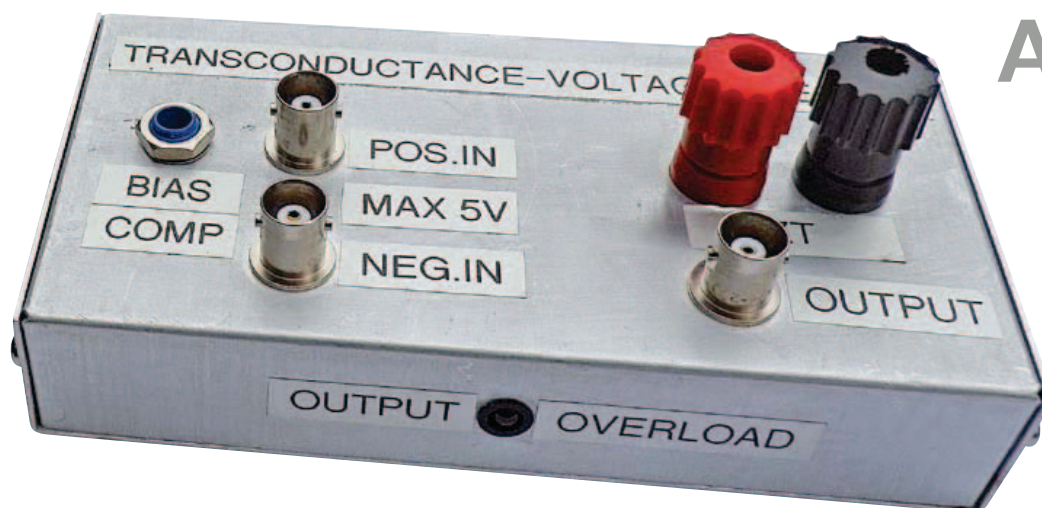
- [1] www.elektor.com/120354
- [2] www.elektor.com/120007
- [3] www.elektor.com/120009
- [4] www.elektor.com/120010

Figure 11.
A regenerative receiver with
feedback.



Transconductance Amplifier

**Voltage in,
current out**



A transconductance amplifier is a special differential amplifier where the output current is a function of the difference in the input voltages. With the help of an AD844, made by Analog Devices, it becomes a simple matter to create such a circuit. A transconductance amplifier like this can be used as a versatile amplifier stage for a wide range of measurement functions, as you'll see in this article.

By Wim de Jager
(The Netherlands)

The AD844 opamp is primarily designed for use in current feedback applications, but due to its particular structure it can also be used in an open loop configuration as a dependent source. Because of this, it's fairly easy to use this opamp as a voltage controlled current source (transconductor), which can be used in various aspects of measurement functions. This article describes the inner workings of the AD844, the design and construction of a pre-amp using this IC and several examples of its use.

Operation of the AD844

A functional diagram of the AD844 is shown in **Figure 1**. Buffer B1 is used to obtain a + input with a high impedance and a - input with a low impedance. The generated current I_{in} in the internal circuit is also available at the external pin TZ, the output of the V/I converter (also called a transconductor). I_{in} in the pre-amp is meant for driving external components or networks. The voltage at TZ is made available at the output via B2. Resistors R1 to R4 and capacitors C1 to C3 are there to compensate for the non-ideal behaviour of the opamp. The values given in the diagram come from the documentation of Analog Devices [1].

When a voltage differential V_{in} is connected between the + and - inputs, the transfer function is $I_{in}/V_{in} = 1/R2 = 20 \text{ mA/V}$. When no load is connected to TZ, $A_o = R3/R2 = 60,000$. The bandwidth depends on $R3||C3$, which is $f_{3dB} \approx 12 \text{ kHz}$ with the values in the circuit.

The design of the pre-amp

The design for the pre-amp (see **Figure 2**) includes a circuit that provides bias current compensa-

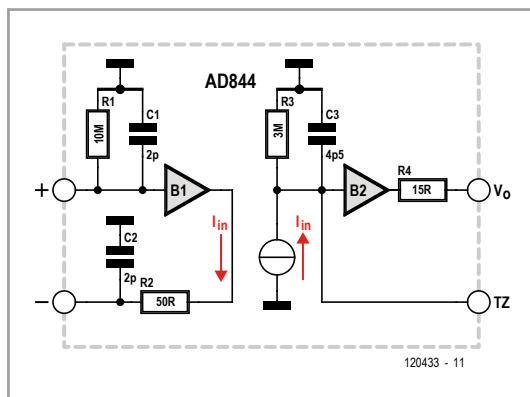


Figure 1.
Functional diagram of the
AD844.

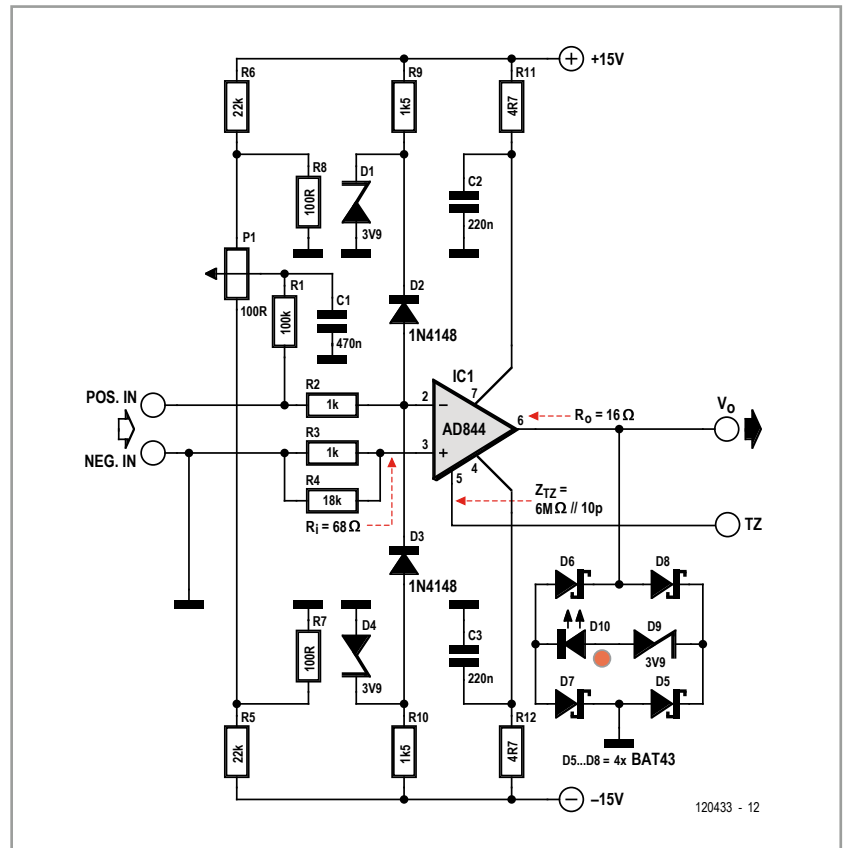
Transconductance Amplifier

tion, a clipping circuit to protect the positive input (POS.IN) and an overload detector for the output voltage (V_o). For all applications in this article the negative input (NEG.IN) is connected to ground. The input signal at POS.IN is connected via R2 to the + input of the AD844.

The maximum permissible voltage between the input terminals is $5 V_{DC}$ and the clipping circuit consisting of D1 to D4, R9 and R10 protects the input against an overload. We have chosen 1N4148 Silicon diodes for D2 and D3. Schottky diodes haven't been used in this case because they have a relatively large reverse-bias current, which would cause an unwanted DC offset. When a $\frac{1}{4}$ W type is used for R2, the maximum input voltage may be $20 V_{DC}$. It is also possible to use a metal film $1 k\Omega$, 2 W resistor, in which case $50 V_{DC}$ is permissible. Large values for R2 (as well as wirewound types) have a negative impact on the HF response. When a coupling capacitor is used, the clipping circuit plays an important part to prevent any damage from occurring to the AD844 from the current spike that occurs when the capacitor is charged up. For the transconductance a value was chosen of 1 mA/V , from which follows that $R3 = 1 k\Omega$. Resistor R4 compensates for the effect of R_i . With a value of $1 k\Omega$ the current at $5 V_{DC}$ is equal to 5 mA, which is the maximum value according to the specifications for the AD844. With these settings the AD844 functions optimally.

The input resistance is mainly determined by R1. The bias current of the + input of the opamp creates a voltage across R1 of 10 mV when the POS.IN connection is open circuit. Since the offset of the opamp is very small, it can be useful to compensate for the effect of the bias current in certain applications (AC coupling or high source impedances). Since the direction of the bias current can vary due to the effect of tolerances in complementary circuits, the adjustment range of the circuit around P1 was designed to be -23 mV to $+23 \text{ mV}$. Resistors R7 and R8 ensure a low resistance path for the wiper of P1 to ground. When POS.IN is driven by a source with a low output impedance there is no need for the compensating circuit. This circuit is not suitable to compensate for the internal offset of the AD844.

TZ is the output of the voltage controlled current source as well as the input of internal buffer B2



from Figure 1. On TZ you can connect any type of component (also semi-conductors) or a network. Since there is no feedback there aren't any stability issues. The impedance Z_{TZ} does introduce a measurement error, but the value of this is very small in most applications. The voltage at TZ (V_o) is made available via buffer B2. With the help of the diode circuit consisting of D5 to D10 an overload detector has been added. V_o is

Figure 2.
Complete circuit diagram of the amplifier, using an AD844.

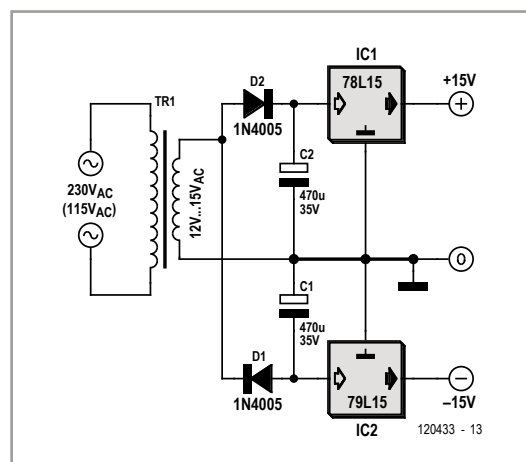


Figure 3.
The associated power supply uses an AC power adaptor. Through the use of half-wave rectification we obtain symmetric supply voltages from a single secondary winding.



Figure 4.
The insides of the prototype built by the author.

full-wave rectified via diodes D5 to D8 (Schottky diodes are used for these because of their low forward voltage drop and their good HF properties) and this signal is fed to the series circuit of zener diode D9 and LED D10. When the LED just lights up the distortion is about 0.5 % due to the maximum output limits being exceeded; a bright light indicates that severe clipping is taking place. The filters R11/C2 and R12/C3 are important for the stability of the AD844. Analog Devices recommends that they are included in any design.

The power supply

A mains adapter has been used for the power supply (see Figure 3). The secondary voltage is 12 V_{AC} at 500 mA. When it is only lightly loaded, as is the case here, the secondary voltage of the transformer rises to about 15 V_{AC}, which is sufficient for this application. In order to obtain a symmetrical power supply of ± 15 V from a single secondary winding it is necessary to use half-wave rectification. With the use of large electrolytic smoothing capacitors this won't be a problem though. Voltage regulators IC1 and IC2 provide a stable output voltage and also provide pro-

tection against overloads (max. 100 mA) and short circuits.

Implementation

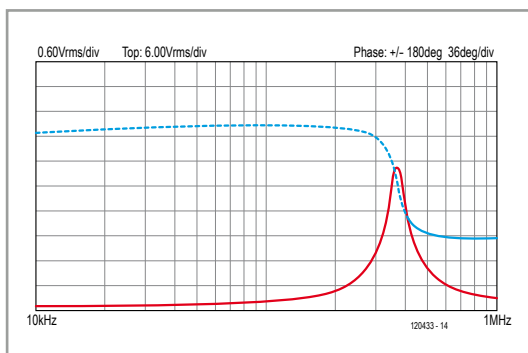
For the enclosure, see Figure 4, use is made of an enclosed aluminium box with a tinned copper-plated board inside. The tinned layer of this board is used as a ground plane for soldering the ground connections to. BNC connectors have been chosen for the inputs and the output, with banana plugs used for the TZ connections. The DIL socket for the AD844 is mounted on a small breakout board, which you could make yourself if necessary. Diodes D5 to D8 have been soldered onto PCB pins that have been inserted into the DIL socket. P1 is a 15-turn cermet potentiometer, which can be used to precisely set the bias compensation. The complementary bipolar transistors inside the AD844 have a high f_T , which makes a careful layout essential in order to prevent parasitic oscillations. The ground plane in conjunction with short connections are the best guarantee for good HF specifications and stability.

Measurements on the pre-amp

The effect of the internal offset of the AD844 was measured with a micro-ammeter, which was connected to TZ, when the input terminals were shorted together. The measured value was 0.1 μ A, which is so low because of the use of laser trimming to minimise the offset. If required, the offset can be reduced even further using an offset circuit described in the documentation for the AD844. With the POS.IN input open circuit the bias current of the + input creates a voltage across R1 of 10 mV, which results in a current of 10 μ A at TZ. This value is much larger than the offset so it makes sense to include the bias compensation circuit just described. With this circuit it is possible to reduce the current to under 0.1 μ A.

The AC response when no load is connected to TZ was measured with an input signal of 0.5 mV at 500 Hz. The gain was then found to be 6000. Since $1 \text{ mA/V} \times R_{TZ} = 6000$, it follows that $R_{TZ} = 6 \text{ M}\Omega$. The -3 dB frequency is at 2.5 kHz. $\omega \times R_{TZ} \times C_{TZ} = 1$, so $C_{TZ} = 10 \text{ pF}$. The measured value for R_i is 68 Ω . For R_o the value is 16 Ω .

Figure 5.
An impedance measurement of an inductor, made with a USB scope with a built-in function generator. 0.6 V/div corresponds to 6 k Ω /div, with the maximum of 6 V being the equivalent of 60 k Ω .



Measurements on capacitors and inductors

Connect a capacitor to TZ and a sinewave generator to POS.IN. Then set the amplitude to say

about 500 mV and adjust the frequency (preferably from high to low so as to avoid any overloads) to a value where the voltage at TZ is also 500 mV. At this frequency the value of the gain is 1 and it's possible to calculate C from the formula $R = 1/\omega C$, from which follows that $C = 1/\omega R$.

As an example, for a connected capacitor a gain of 1 occurred at a frequency of 16 kHz. Since $R = R_3 = 1 \text{ k}\Omega$, the formula gave a result for C of 10 nF. When a Voltcraft 9036 LCR meter was used to measure this capacitor it was found to have a value of 10.47 nF.

For the measurement of self-inductance we also recommend a generator voltage of about 500 mV. In this case the frequency should be adjusted from low to high in order to avoid any overloads. When an inductor with a toroidal core was connected we found that a gain of 1 occurred at 56 kHz. In this case $\omega L = R$ and $L = R/\omega = 2.84 \text{ mH}$. When the LCR meter was used we obtained a value of 2.86 mH. It is also possible to calculate the self-capacitance of an inductor winding (C_p) from the frequency at which self-resonance occurs. For this particular inductor this frequency was found to be 360 kHz. Since $C_p = 1/\omega_0^2 L$, the calculated value was 69 pF. However, to obtain the correct value you have to deduct $C_{TZ} = 10 \text{ pF}$, which means that the true value for C_p is 59 pF. At self-resonance the impedance is equal to the parallel equivalent resistance, which can be determined from the measured gain. The value of the gain was 38, from which follows that $R_p = 38 \times R_3 = 38 \times 1 \text{ k}\Omega = 38 \text{ k}\Omega$.

Impedance measurements with the Velleman PCSGU250

The PCSGU250 USB oscilloscope from Velleman contains a built-in function generator. It is possible to use it as a circuit analyser with the help of the associated software. With this combination it is possible to create Bode plots, where the modulus and argument of the transfer function of an electronic circuit is shown as a function of frequency. The V/I converter of this pre-amp makes it possible to measure the modulus and argument of an impedance. This type of meter is sometimes called a vector impedance meter because the phase is measured as well. In **Figure 5** you can see the impedance and phase characteristics of the inductor used previously. The phase is about 90° up to about 200 kHz, which is to be expected for an inductor. At 370 kHz the self-resonance can be seen, from which the parallel equivalent resistance can be determined. Since the amplitude of the generator is 100 mV is, the vertical scale for the modulus (Y) is 6 k Ω /div. The measured value for R_p is 34 k Ω in this case.

Measurements on semiconductors

A current output is also useful for measuring the characteristics of semi-conductors, such as bipolar transistors. A measure-

DEEP MEMORY MIXED SIGNAL OSCILLOSCOPES

NEW from Pico Technology



PORTABILITY & PERFORMANCE

pico[®]
Technology

PicoScope	3204 MSO	3205 MSO	3206 MSO
Channels	2 Analog 16 Digital		
Bandwidth	60 MHz	100 MHz	200 MHz
Buffer memory	8 MS	32 MS	128 MS
Resolution (enhanced)	8 bits (12 bits)		
Signal generator	Function generator + AWG		
Price	£649 \$1070 €785	£849 \$1400 €1028	£1049 \$1730 €1270

ALL MODELS INCLUDE ADVANCED DIGITAL TRIGGERS, SERIAL DECODING (I²C, SPI, RS232, CAN, LIN, FLEXRAY) MASK LIMIT TESTS, SEGMENTED MEMORY, DIGITAL FILTERING, FREE SOFTWARE UPDATES AND 5 YEAR WARRANTY

www.USBmso.com/PS170

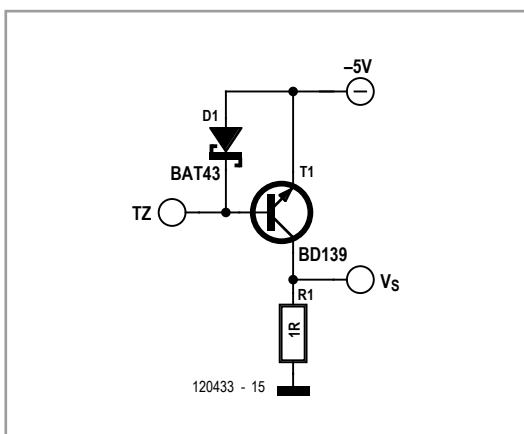


Figure 6.
The circuit for carrying out measurements on an NPN transistor.

ment circuit for an NPN transistor is shown in **Figure 6**. The -5 V power supply at the emitter makes it possible to connect the measurement resistor ($R1$) to ground. The voltage across this resistor ($1\ \Omega$) is proportional to the collector current. The TZ connection of the pre-amp is connected to the base of the transistor. The base current is equal to 1 mA/V . Schottky diode D1 protects T1

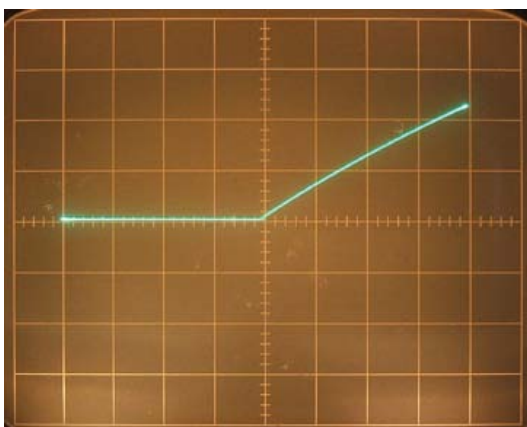


Figure 7.
This is the scope display when a measurement is carried out on the circuit from Figure 6. Horizontal 1 mA/div , vertical 0.2 A/div .

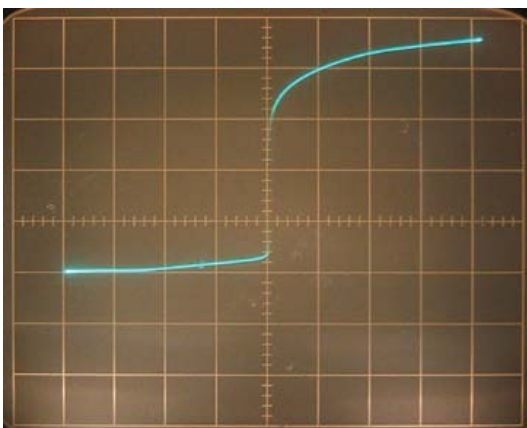


Figure 8.
The display of a measurement on a zener diode. Horizontal 0.5 mA/div , vertical 1 V/div .

against high reverse-biased voltages. The β graph was measured with an oscilloscope (see **Figure 7**), the vertical being the voltage across $R1$ and the horizontal the voltage at POS.IN. The vertical sensitivity is set to 0.2 V/div , which corresponds to 0.2 A/div for the collector current. The horizontal sensitivity is 1 V/div and that corresponds to 1 mA/div base current. Since the signal at V_s is negative the 'invert' mode (which may not be found on every oscilloscope) was selected to obtain a positive display. From the graph you can work out that at $I_c = 400\text{ mA}$ the current gain $\beta = 118$. The second example is for the measurement on a zener diode, see **Figure 8**. The horizontal scale is 0.5 V/div , which corresponds to 0.5 mA/div for the current through the zener diode. The vertical scale is 1 V/div , which is for the voltage across the zener diode. At a current of 2 mA the zener voltage is equal to 3.6 V , which was exactly what was expected from the specifications.

Pre-amplifier

A DC-coupled preamplifier can be created very easily by connecting a load resistor (R_L) to TZ. For example, with $R_L = 10\text{ k}\Omega$ the gain is $R_L/R_3 = 10$. The bandwidth can be calculated from $R_L||C_{TZ}$ and is equal to 1.6 MHz . Measurements have confirmed these values. A possible use for the pre-amplifier is to improve the sensitivity of any measurement equipment.

Buffer

The output buffer can also be used as an independent voltage follower, with TZ as the input and POS.IN grounded. The bandwidth is about 10 MHz . The input impedance is equal to $Z_{TZ} = 6\text{ M}\Omega//10\text{ pF}$ and the output resistance is $R_o = 16\ \Omega$. The maximum output voltage is 5 V_{rms} with a THD of 0.7% (1 kHz). The maximum output current is 80 mA_{DC} and the circuit is protected against short circuits.

(120433)

Internet Link

[1] www.analog.com



DESIGNSPARK

NEW. FREE. **DESIGNSPARK PCB** **VERSION 4**

COMPREHENSIVE NEW LIBRARIES.
EASIER COMPONENT SELECTION & QUOTATION.
NEW PCB PROTOTYPING SUPPORT.

Discover today at
www.designspark.com

UNIQUE
RESOURCES BY 

PWM to phase shifter

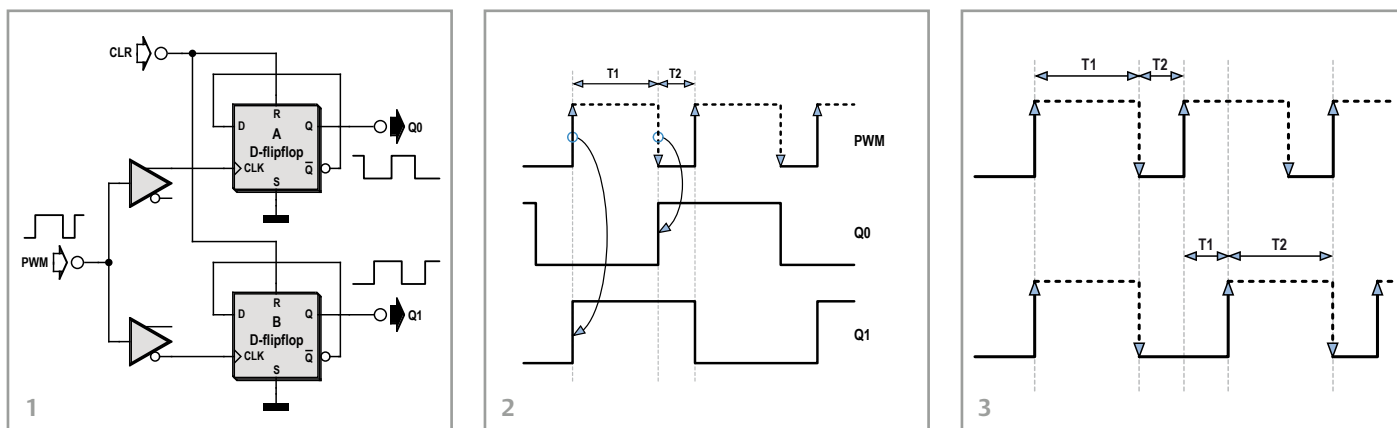
By
Stephen Bernhoeft
(United Kingdom)

In some situations it can be useful to vary the phase between two square waves. In the schematic shown in **Figure 1**, the two D-flipflops 'A' and 'B' convert the PWM signal into a square wave in the standard manner. However, D-flipflop 'A' is clocked by the rising edge of a pulsewidth modulated (PWM) signal, whereas D-flip flop 'B' is clocked by the falling edge. This way the duty cycle of the PWM signal determines the phase difference between the outputs of the two flipflops, as illustrated in timing diagram **Figure 2**. A limitation is that phases 0 and 2 pi are not possible. A possible variation would be to replace both A

and B by quadrature output blocks. If then Q0 is taken from the 'I' (in-phase) output of block A, and Q2 is taken from the 'Q' (quadrature) output of block B, a phase difference of pi in the original scheme maps to a phase change of 2 pi. Of course, then 'exactly pi' (i.e. quadrature) phase difference would be impossible.

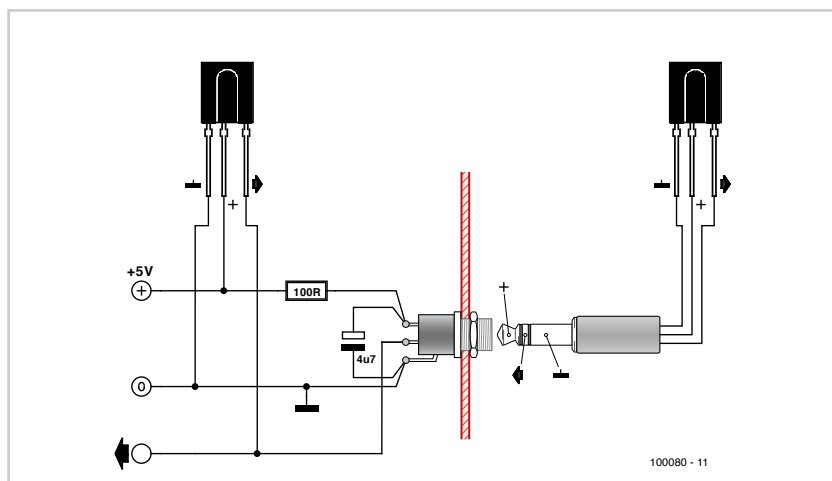
If the PWM is microprocessor controlled, then the relative phases can be made to cross because (for example) a 'trailing Low' may be 'stretched into a 'leading Low', as pictured in **Figure 3**.

(120193)



Improving the pick-up angle of an infra-red satellite receiver remote

By **Géry Szczepanski**



When the TV is on a rotating base and the satellite receiver remains fixed, the infrared remote sometimes has trouble getting through to the satellite receiver. This is easy to put right if you're a bit of a DIYer, as all you have to do is add a second infra-red detector in parallel with the receiver's existing one. This can also be applied to a DTT decoder, TV, etc. Here's how to go about it for an XSAT CD TV360 satellite receiver.

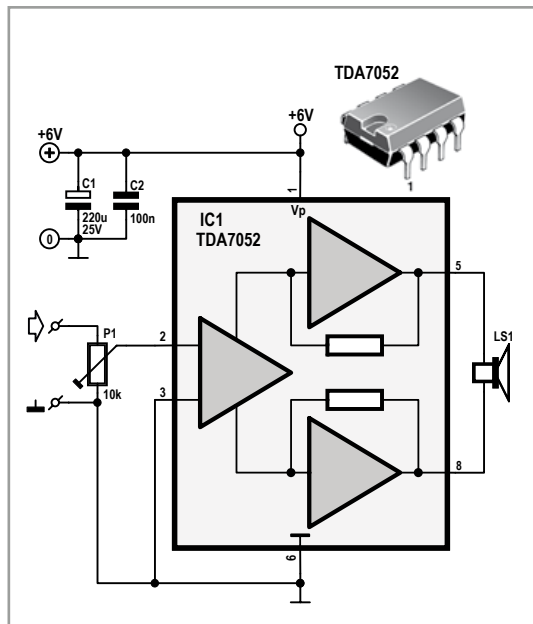
Open the satellite receiver, unclip the front panel and hinge it downwards with its PCB. Identify the ground, +5 V rail and the 'signal' terminal (using a voltmeter, it is at +4.5 V). Fit the circuit

show here in parallel with the existing detector. Close the satellite receiver back up again. The second IR detector is attached to the TV using a spot of glue. Under these circumstances, the satellite receiver could even be located inside a

unit, a drawer, behind a piece of furniture, etc. (but don't forget to think about ventilation). Watch out, plugging the jack in or out must not short out the 5 V rail, even very briefly.

(100080)

Audio amplifier in dinner mint format



There are countless situations and systems devisable in which a sound signal needs to be amplified in order to drive a (small) loudspeaker, but where space constraints rule out the use of a regular sized amplifier. For these situations, this sub miniature amplifier is perfect. With some skills it can be assembled to a size smaller than an after dinner mint!

The TDA7052, which was released by Philips a many moons ago, is a typical example of a fully integrated circuit. The only external components required are two decoupling capacitors. That's all. In the circuit diagram of the amplifier you can find an internal schematic of the integrated circuit. It's easy to see two amplifiers connected in a bridge arrangement. This is done in order to squeeze a 'decent' power from the IC at relatively low supply voltages. A preset has been added to the input to prevent the circuit from being overdriven, this will

be appreciated in view of the high sensitivity of the TDA7052. If required, this can of course be a real potentiometer with knob and all to adjust things on the fly.

Maximum power output is just over 1 watt, which is more than enough for most applications. The power supply voltage can be up to 18 V, but be aware that voltages above 6 V or so can cause the IC to become hot. When using higher voltages, use a loudspeaker with an impedance greater than 8 ohms or limit the input signal of the amplifier. Don't worry about FUBR-ing the amplifier though. Even though the IC heats up, an internal thermal security will stop anything from blowing up.

A few measurements were carried out on the prototype and the results in the table below show what happened at a supply voltage of 6 volts and a loudspeaker impedance of 8 ohms:

By
Ton Giesberts
(Elektor Labs)

THD+N*	0.09% (1 kHz, 100 mW in 8 Ω), 0.3% (20 kHz, 100 mW in 8 Ω)
P _{max}	750 mW (THD+N = 1%) 1 W (THD+N = 10%, heavily clipped output signal**)
Gain	38 dB (P1 at maximum)
Supply voltage	3–18 V
Current consumption	5 mA (quiescent), 340 mA (1 W continuous power output)
* THD+N = Total Harmonic Distortion plus Noise	
** clipping only noticeable at 2% distortion.	

Pretty good results for an amplifier this simple!

The construction of this mini amp is unlikely to cause problems. If you work neatly, the circuit will most definitely work. Some people have taken it as a challenge to make the circuit as small as possible.

Whatever you choose to do, we wish you a lot of fun tinkering!

(i-00909)

Creating a Low-Leakage Rectifier using a self-powered opamp

By **Martin Tomasz**, Senior Scientist, Touchstone Semiconductor, Inc.

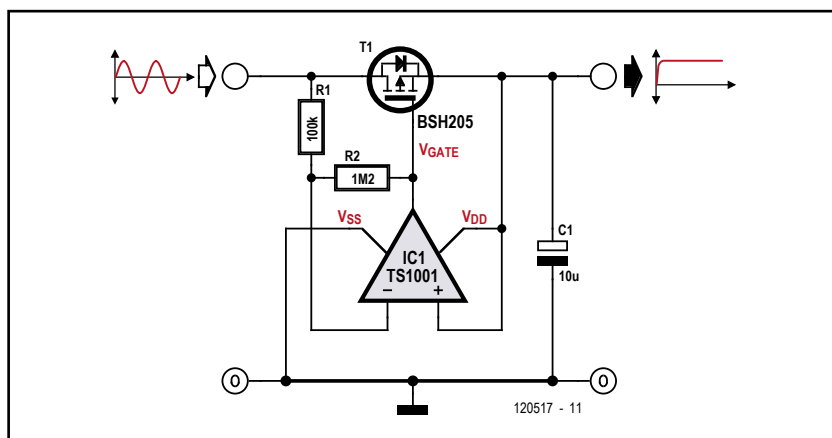


Figure 1. This circuit emulates a rectifier, but it has forward-voltage drop of 40 mV or less. The circuit has less reverse leakage than a Schottky diode.

A nanopower opamp, a low-threshold P-channel MOSFET, and two feedback resistors are combined to make a rectifier circuit with less forward drop than a diode (**Figure 1**). Since the rectified output voltage powers the active circuitry, no additional power supply is necessary, and the circuit's quiescent current is lower than most Schottky diodes' reverse-leakage current.

This circuit provides active rectification at voltage drops as low as 0.8 V. At lower voltages, the MOSFET's body diode takes over as an ordinary diode.

When a forward voltage develops between the input and output voltages, the opamp circuit turns on the MOSFET, according to the following equation:

$$V_{\text{GATE}} = V_{\text{OUT}} - (R_2 / R_1) \times (V_{\text{IN}} - V_{\text{OUT}})$$

where V_{GATE} is the MOSFET's gate drive, V_{IN} is the input voltage, and V_{OUT} is the output voltage. The input and the output voltages to the MOSFET's drain-to-source and gate-to-source voltages are described by the following equations:

$$V_{\text{DS}} = V_{\text{IN}} - V_{\text{OUT}} \quad \text{and} \quad V_{\text{GS}} = V_{\text{GATE}} - V_{\text{OUT}}$$

where V_{DS} is the drain-to-source voltage and V_{GS} is the gate-to-source voltage. Algebraically, these equations relate to the MOSFET's gate drive to a function of the drain-to-source voltage:

$$V_{\text{GS}} = -(R_2 / R_1) \times V_{\text{DS}}$$

A good design choice is to make R_2 12 times larger in value than R_1 , resulting in a 40-mV voltage drop across the MOSFET's drain-to-source voltage, sufficient to turn on the MOSFET at low drain currents (**Figure 2**). A higher ratio further reduces the voltage drop within the limits of the opamp's worst-case input-offset voltage of 6 mV. Power for the opamp (the Touchstone Semiconductor TS1001) comes from the circuit's output-the output-reservoir capacitor C_1 . Since the amplifier has rail-to-rail inputs and outputs and no phase inversion when operating near the rails, connecting its supply rails to the output works well. Additionally, since the amplifier operates at power-supply voltages as low as 0.8 V, the circuit's rectifying action works to this low voltage. The opamp's non-inverting input is connected to the V_{DD} rail and the amp's output to the gate of the MOSFET. The circuit consumes slightly more than 1 μA when actively rectifying a 100-Hz sine wave, less current leakage than that of most Schottky diodes. The BSH205's threshold is low enough to support milliamp-level currents at a gate-to-source voltage of 0.8 V.

About the author

Martin Tomasz is a seasoned analogue, radio, and mixed-signal engineer with 22 years' experience in circuit and systems design. As senior scientist at Touchstone Semiconductor, Martin has 17 patents to his credit. Prior to joining Touchstone Semiconductor, Martin spent two seasons in Antarctica working for the US Antarctic Program, working with scientific monitoring devices designed to work in extreme environments.

Low-Leakage Rectifier

The opamp's bandwidth does limit the circuit to lower-frequency signals; at bandwidths higher than 500 Hz, the amplifier's gain begins to decline. However, as the signal frequency increases, the body diode of the MOSFET takes over the rectification function while the MOSFET remains off. Fast fall time input signals could potentially drag the output with reverse current through the MOSFET, before the opamp has a chance to turn the MOSFET off. However, for small currents, the MOSFET operates in its sub-threshold range. On the other hand, the amplifier quickly turns off due to the exponential relationship of the gate-to-source voltage to the drain-to-source current in the sub-threshold range. Ultimately, the amplifier's slew rate of 1.5 V/ms is the limiting factor. As long as the circuit's load is light enough to keep the MOSFET from operating in its linear range, reverse currents won't exceed forward currents.

A micropower solar-harvesting application is shown in **Figure 3**. The BPW34 cells generate 10 to 30 μA at 0.8 to 1.5 V, depending on the light. This circuit rectifies the peak harvested voltage in conditions of rapidly changing light and minimizes reverse leakage to the cells at low light conditions.

(120517)

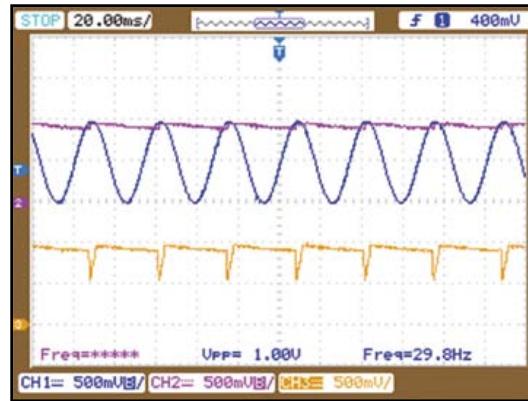


Figure 2.
The output of the circuit (green) with a sinewave input (yellow) shows that the FET's gate voltage (blue) drops out only when the input-to-output differential is less than 40 mV.

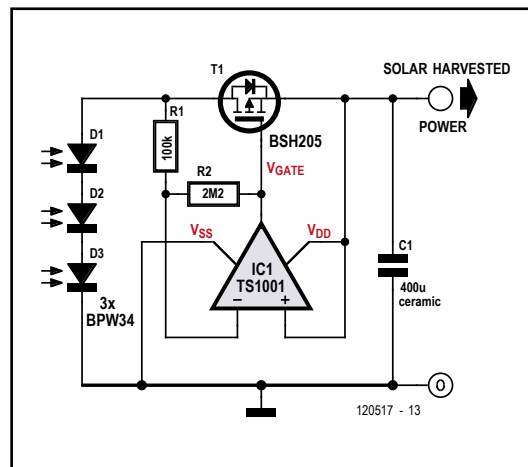


Figure 3.
You can use the active-rectifier circuit to charge a capacitor from solar cells. The rectifier has a low voltage drop and protects the cells from reverse current when there is no light.

World's smallest humidity and temperature sensor

Sensirion's tiny SHTC1 humidity and temperature sensor is specifically designed for mobile devices where size is a critical factor. Sensirion has rigorously followed the maxim "smaller is better" and developed the world's smallest sensor in its class, measuring a mere 2 x 2 x 0.8 mm. The sensor is based on the proven CMOSens® Technology, which allows the sensor and the signal processing electronics to be combined on a single silicon chip to achieve small device size. Production is targeted at high volume to allow low unit prices to be achieved by economy of scale. The supply voltage of 1.8 V is also fully in line with the needs of the target industry.

The specifications of the new sensor show that the size, power consumption and price reductions do not have any impact on performance. It measures relative humidity over a range of 0 to 100 %RH with a typical accuracy of ± 3 %RH. The temperature measuring range is -30 to +100 °C with a typical accuracy of ± 0.3 °C. The sensor is fully calibrated, has a digital I²C interface, and is suitable for reflow soldering. This makes it compatible with standard industrial mass production processes for electronic modules. To ensure effective integration of the sensor in complex applications, Sensirion draws on its many years of experience with sensor products deployed in the millions and supports customers with corresponding application expertise. The new SHTC1 humidity and temperature sensor combines extreme miniaturization with top performance, and with Sensirion's implementation support it is doubtless a decisive step ahead of the competition.

www.sensirion.com/shtc1 (120663-II)



State-of-the-art audio to truck cabs and buses



STMicroelectronics has introduced the market's most powerful and most simple-to-use audio power amplifier for commercial vehicles such as trucks, buses and agricultural vehicles, whose electrical systems run at 24 V rather than the 12 V used in cars. It is the world's first audio amplifier that can drive the cabin loudspeakers directly, without any external components, in 24-V vehicles, con-

firming ST's dedication to the heavy-vehicle segment. Audio amplifiers used in automotive systems must be designed to withstand the demanding automotive environment, including a high level of electrical noise and the need to provide comprehensive protection against potential hazards such as short circuits, load dumps, and excessive chip temperature. Audio amplifiers for use in 12-V automotive systems have reached a very high level of sophistication, but up till now audio equipment manufacturers have been unable to replicate this performance in 24-V systems without using external components that not only drive up the Bill of Materials cost, but can even compromise safety under certain conditions.

ST's TDA7576B 24-V audio amplifier uniquely solves this problem. It is built in a proven high-volume semiconductor manufacturing technology called BCD5 that allows a single chip to include both the power blocks and the signal management blocks — so there is no

need to use a more expensive multi-chip approach — and it can sustain a peak supply voltage of 60 V without any damage. The most powerful audio power amplifier for 24-V systems available on the market, the TDA7576B has been designed with the same guidelines and care that ST uses for car radio audio power amplifiers, so the final user can enjoy the same high-quality audio as car drivers.

The TDA7576B audio amplifier expands ST's dedicated portfolio of products for 24 V vehicles that includes recently announced chips for exterior and interior lighting. Key features of the TDA7576B include:

- 24V battery operation;
- High output power (2 x 20 W);
- Minimized external components (no decoupling capacitors, no bootstrap capacitor and no external compensation components);
- Integrated power-saving standby function;
- Diagnostic output pin that alerts the host controller in the event of clipping, short circuits or excessive temperature;
- Output DC offset detection;
- Protection against 60V load dump, excessive chip temperature, ESD (Electrostatic Discharge) and output short circuit to GND, Vcc, or across the load.

The TDA7576B is available now in a compact Multi-watt15 package.

www.st.com/internet/automotive/product/254152.jsp
(120605-V)

Magic PCB and more from Beta LAYOUT

Beta LAYOUT's newest innovation 'Magic PCB' was unveiled at this year's electronica exhibition in Munich, Germany. Beta LAYOUT has developed a groundbreaking application machine to embed RFID chips into a printed circuit board. The chips are embedded during the initial production steps allowing for identification and traceability from the onset. RFID chips have a high reading accuracy with the added advantage of an invisible copy protection. The RFID chips function within the UHF frequency bands (860 MHz to 960 MHz) and can be operated globally. The second highlight at the Beta LAYOUT booth was their new online IDF-to-3D tool (IDF=Intermediate Data Format). Using any printed circuit board software that outputs IDF data this tool allows design engineers to create a 3D data model of their assembled printed circuit board. This powerful 3D conversion tool enables the design engineer to check and verify the dimensions and position of components, such as keyboard sections, plugs, sockets etc. on the soft version before freezing the design. Beta LAYOUT continues its unique relationship with EAGLE users by developing a specific ULP (User Language Program) to interface with the IDF-to-3D tool specifically for them. Beta LAYOUT offers this free of charge to its loyal EAGLE user customer base.



www.betalayout.com (120663-III)

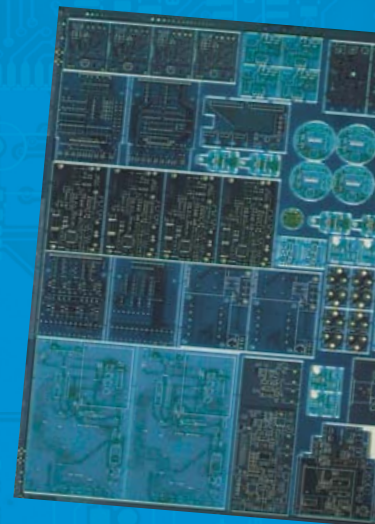
The European reference for PCB prototypes and small series

Easy-access online pooling services cut your board costs

- No tooling charges
- No minimum order charge - order from 1 PCB
- Instant online ordering without pre-payment

New added-value PCB services get your designs into production faster

- PCB Visualizer immediate online data check
- Online advice and tips on design for manufacture



PCB proto – *dedicated engineering prototype service, fast and low-cost*

- 1 or 2 PCBs in 2, 3, 5 or 7 working days
- DRC-checked and fully finished with 2 soldermasks and 1 legend, 150µm technology
- 1 x 100 x 80mm in 7WD - 2 layer 47,03 € - 4 layer 95,52 €
- 2 x 100 x 80mm in 7WD - 2 layer 36,89 € each - 4 layer 74,76 € each

Prices quoted include 21% Belgian VAT but exclude transport costs

STANDARD pool – *the widest range of pooling options in Europe*

- 1-8 layers 150µm technology PCBs
- from 2 working days

TECH pool – *all the benefits of pooling for high-density PCBs*

- 2-8 layers 100µm technology PCBs
- from 4 working days

IMS pool – *aluminium-backed PCBs for high heat-dissipation (LED) applications*

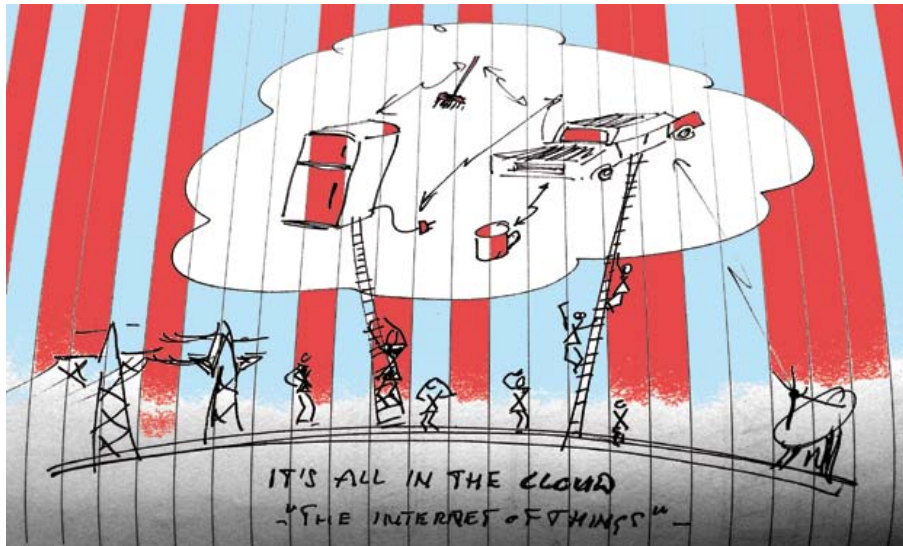
- Single layer Insulated Metal Substrate PCBs
- from 3 working days

On demand – *multi-option non-pooling service for specialist needs*

- 1-16 layers down to 90µm technology
- from 2 working days

The Internet of Things

By Tessel Renzenbrink (Elektor TTF Editor)



The Internet of Things (IoT) is the merger of the physical and the digital world. Ordinary objects featuring tiny computers will be assigned an IPv6 address and have a digital representation on the Internet. We will be able to access information about the objects in our daily lives over the internet and even remotely control them.

Many instances of the IoT won't even have humans in the loop. The IoT as a machine-to-machine network where sensors and actuators communicate with each other to move the environment to a desired state.

In September the workshop ROBIOTS: Robots Meet Internet of Things took place as part of the two-day PICNIC festival for media and innovation, hosted in Amsterdam. Afterwards I discussed the future of IoT, its technological challenges and its potential dark side with three of the speakers: Florian Michahelles, Heico Sandee and Stefan Gessler.

The IoT experts

Dr. Florian Michahelles is the associate director of the Auto-ID Labs Zurich/St. Gallen. The Auto-ID Labs are the leading global network of seven academic research laboratories in the field of networked RFID. His research interests centre around the Internet of Things.

Dr. Heico Sandee, program manager at the Department of Mechanical Engineering at Eindhoven Technical University, is specialized in control systems. He participates in the RoboEarth project, a multi-disciplinary partnership of robotics researchers from academia and industry aiming to create a world wide web for robots.

Stefan Gessler is project manager at the Software and Service Research Division of NEC Laborato-

ries Europe. He participates in the Florence project, an activity to make low-cost robots interact with a smart home environment to support elderly people in their daily life at home. He is also involved in the formation of the IoT-Forum.

The purpose of the IoT

Tessel: Why does the world need the IoT?

Florian: One answer is: it just happens. It's just the natural evolution of technology.

Heico: It's about span of control. I am so used to having the power at my fingertips to communicate with everyone, I find it almost stupid I can't communicate with the things in my life. I want to be able to check on my phone whether I forgot to lock my door. I want to have access to this information and even be able to close it remotely.

Stefan: The IoT is all about cooperation and communication. Communication is a prerequisite for cooperation. You can think of thousands of scenario's in which it is an advantage when things work together. For instance, a sensor detecting water on the bathroom floor can signal the water tap to close itself. Or a car detecting a safety hazard ahead telling the cars behind it to slow down. Heico: And it opens up entirely new possibilities. In the RoboEarth project we're creating a system to exchange information between objects, robots and humans. Say, you connect your coffee machine to your alarm clock. As soon as the

alarm goes off the machine starts making coffee. This is already possible if you have a smart coffee maker. But what if the machine runs out of coffee? This is where the robot comes in. Because it has the capability to move it can refill the coffee bean container. Robots will give arms and legs to the Internet.

And when robots are part of a network they can learn from each other. RoboEarth is building a cloud-based database which can be accessed by different kinds of robots. When a robot has solved a problem it can add this information to the cloud so that the next robot encountering the same problem can download the instructions. This way you don't have to plan for every possible problem in the design phase of the robot, or equip them all with a large array of sensors to give them learning abilities.

Technological challenges

Tessel: What are the technological challenges of realising the IoT?

Stefan: The biggest challenge is the modelling of information. To enable every device to communicate and to cooperate with all other devices. What happens a lot at the moment is the so called silo-approach. You have a certain application field and devices designed to execute one particular task. This system is a closed box consisting of sensors, actuators and software. If you want to add a second application you have to buy it all again, the sensors, the actuators but this time with different software and protocols. The next step we have to take is reusing the data, sharing it among the devices. But that means you need to have an interoperable interface. That is what we are working on at the moment.

Florian: At the Auto-ID labs we focus on the product side: how to make the flow of items and the supply chain more efficient. The idea is to have a barcode or uniquely identifiable RFID number that is associated with data describing the type of product. And now we recognize that the barcode –a forty year old technology- can also be useful for consumers with smartphones. But the consumer is interested in different data. He isn't interested in the size or the weight of the product but in the ingredients or its application. And next we have to derive semantics from the data which can be interpreted by robots. In the IoT different systems that emerged for different reasons have to work together.

But standardization isn't so much a technical

challenge but rather an organisational and motivational one. Because it isn't one company providing all the infrastructure but many disparate parties. How do you get all the parties to play together? Indeed, you have the data generated by one company but what is the incentive to share? So the question really is: what is the ecosystem? What is the business case you can set up that makes the different parties willing to share because they mutually benefit from it?

Heico: I agree that interoperability is the biggest challenge but is it really such a big problem? Isn't it a matter of making a choice and sticking to it?

Stefan: It is about making a choice but in this choice you have to consider all the cases which could possibly appear. Florian: Exactly, that's the point. You have to understand what you need it for. Technologically it is difficult to build a solution that fits every case. So it is easier to agree on, say, five cases and get a standard on it.

Stefan: The IoT is going to develop bottom up. We will not build an IoT system and switch it on like we once did with TV technology.

Internet infrastructure

Tessel: The Internet is the backbone of the IoT. Is the current state of the Internet infrastructure sufficient?

Stefan: We will need increasingly more bandwidth. Everything is working so far but we have to be prepared for Big Data. The growth of data is exponential. As one of the speakers pointed out in the workshop: 5 billion gigabytes of data was created since the beginning of recorded time. In 2011 that same volume of data was created in two days. By 2013 we will create it in 10 minutes. The capacity of the radio spectrum is inherently limited. You can't add an unlimited amount of devices to the wireless network.

Florian: You have to manage the scarcity. Today if the Internet is choked it may take a little longer for you to



Florian Michahelles.



Heico Sandee.



Stefan Gessler.



download that picture. Most of the data traffic isn't urgent. But with the IoT it might be different. There might be health applications involved which depend on permanent access. To manage the scarcity you may have to give priority to health applications or have users pay a little fee for priority access. I think we will always be at the limits of bandwidth but I don't think it is a showstopper for the IoT.

Stefan: On the contrary, the IoT could contribute to a more efficient organization of data traffic. It's about making decisions as early as possible. Data relevant to a local network doesn't need to be pushed to a remote datacenter. What we see now is two distinctions: local and the cloud. In the future we will have different layers, different levels of cloud. The data will be organized hierarchically: can it stay local, or does it have to be made available regionally or globally?

Dark side

Tessel: What are the possible dangers of this technology?

Heico: One of the worries is security. We already see what damage malicious hackers can do to the Internet. When all kinds of objects are connected to the Internet the impact of systems being compromised is even bigger.

Stefan: Whenever you introduce something new it can also be used for something negative. Another thing is that a new technology always influences the way we live our lives. Many people consider changes as a threat, so we also need to take care about the acceptance of IoT in future. Security, but also privacy are certainly key factors.

Florian: A possible danger is transparency. As humans we are used to understanding why things happen. With this digital component added to our world, we may not perceive the cause of occurrences with our senses anymore. That is why we continuously have to think about designing the technology in such a way that we always have a mental model to understand what is going on. It's scary if I can't grasp what's happening. It means I'm controlled by something else. That's a dark scenario. But in my opinion the worst case scenario for the Internet of Things is if it doesn't happen.

Elektor and the IoT

As Stefan Gessler rightly points out, the IoT isn't one technology with a single switch-on moment. Rather, it is a boundless sprawl of technologies and applications many of which are already in operation. Probably unknowingly, Elektor has already covered many of them. In the November edition alone I counted five IoT projects, proving that Elektor's new e-departments .LABs and .MAGAZINE cover IoT without expressly identifying it. We need to work on that!

One of the best known IoT applications is the smart meter. Serving as the nerve endings of the smart (electrical) grid, smart meters provide information about power consumption at the local level enabling both energy suppliers and consumers to make informed decisions about saving energy. Last month Søren Mikkelsen showed how to install an S0-compatible electricity meter that sends detailed power consumption data to a local or cloud-based database.

In the November edition Benedikt Sauter in his series "Embedded Linux Made Easy" described how to free your Elektor Linux board from a lonely existence by setting up a network connection. And Jens Nickel introduced the C library for the ElektorBus to make it easier for developers to add microcontroller nodes. Both the Elektor Linux board and the ElektorBus are crying out to be connected to the Internet.

Also in the November 2012 edition Dr. Joost Mertens went to the very basics of internet technology when explaining how to build a network interface card (NIC) using a 8-bit AVR microcontroller. When paired with an old parallel port camera, the microcontroller allows you to send video to multiple recipients on a network.

In part 3a of the "Arduino on Course" titled 'Connect the physical world to your Android phone' David Cuartielles and Andreas Goransson set out to teach you how to start building your own IoT. Cheerfully continued this month!

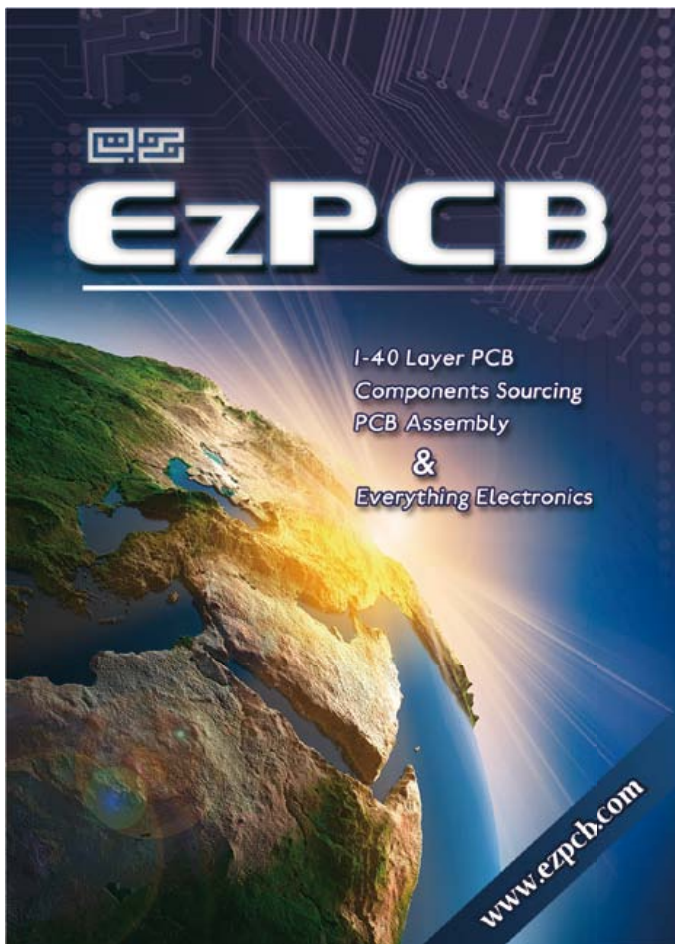
Isn't it great to discover a trend when you appear to already have everything at hand that defines it? Which project in this December 2012 edition do you think is IoT ready?

(120632)

Internet References

- [1] www.autoidlabs.org
- [2] www.roboearth.org
- [3] <http://iot-forum.eu>

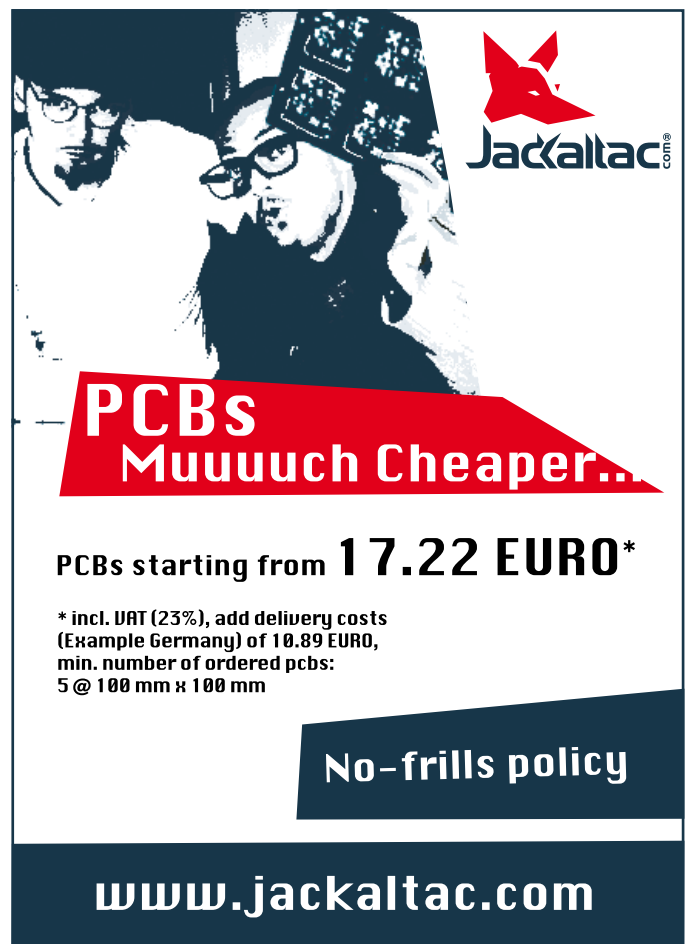




EzPCB

1-40 Layer PCB
Components Sourcing
PCB Assembly
&
Everything Electronics

www.ezpcb.com



Jackaltac com

PCBs Muuuuch Cheaper...

PCBs starting from **17.22 EURO***

* incl. VAT (23%), add delivery costs
(Example Germany) of 10.89 EURO,
min. number of ordered pcbs:
5 @ 100 mm x 100 mm

No-frills policy

www.jackaltac.com

elektor preferred suppliers

Now on www.elektor.com:

Elektor Partners showcasing hot and relevant information to all Elektor members.

Including:

- Coast Electronics • CS Technology
- Easysync • Elnec • FTDI Chip
- FTT First Technology Transfer
- Robot Electronics • Robotiq



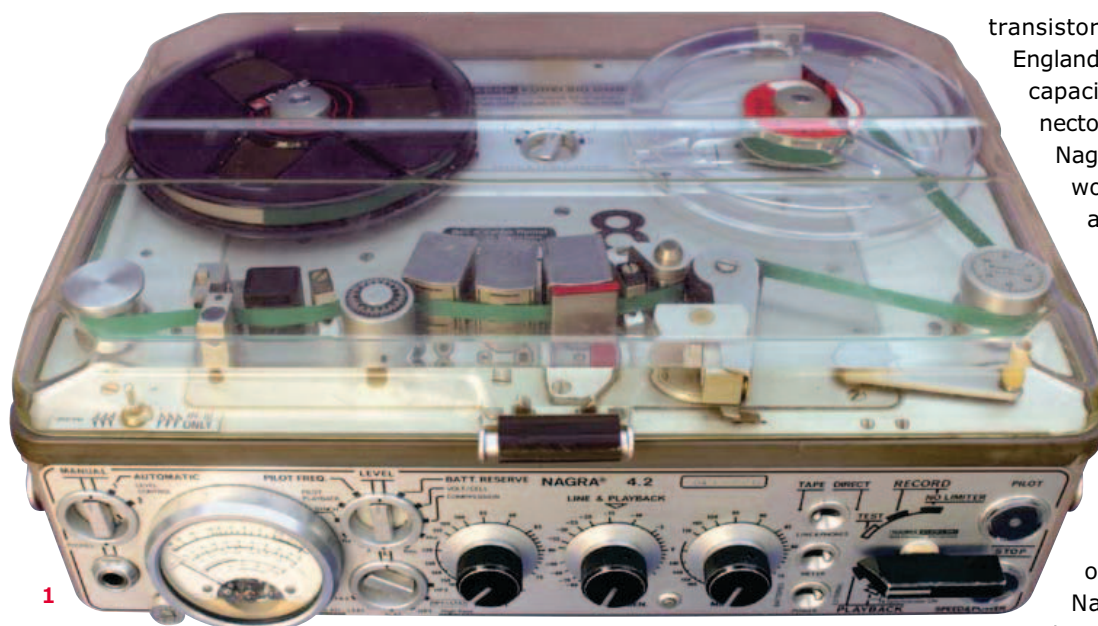
Get Elektorized

You are just one click away from reaching more than 500,000 visitors per month and 2 million monthly pageviews by Elektor members.

For more information of how to become an Elektor Preferred Supplier, contact Johan Dijk by j.dijk@elektor.com

The Nagra IV Tape Recorder

A reporter's delight, ca. 1968



1

By Peter Beil (Germany)

It was once the most celebrated portable tape recorder in the world – and yet few Elektor readers will have heard of it. The Nagra (**Figure 1**) was a professional device, developed for professional transportable applications, whether for broadcast use or picture-synchronized audio recording for film and television.

Stefan Kudelski, a Pole by birth and its spiritual father, specified for it the highest standards technically measurable at the time. Its innards contained

transistors from America, pots from England, switches from Switzerland, capacitors from Holland and connectors from Germany. The name Nagra is derived from the Polish word for recording. Its quality and reliability have given the Nagra legendary status and one encountered the device from ice-cold Alaska to the searing heat of the Sahara.

The portable tape recorder era had begun in the 1950s using models driven by clockwork mechanisms, with the launch of the electrically driven Nagra III taking place at the beginning of the 1960s. At the

end of that decade came the most widely used model, the Nagra IV, which is the subject of this article.

The Nagra draws its power from 12 'D' cell (IEC R20) batteries, which together account for a third of the machine's bulk (**Figure 2**). The unregulated voltage from these was available for tape spooling or the playback monitor amplifier. The latter delivered a crystal-clear 1.5 watts with remarkable sound intensity and quality. As long as you were sparing in your use of fast forward/rewind and loudspeaker playback, a set of batteries would last without difficulty. Later on you

2



3



Retronics is a monthly section covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com



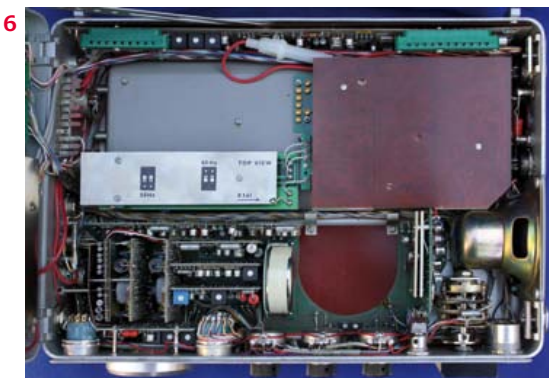
could get rechargeable batteries in this form factor, which meant you were dragging 20 pounds (10 kilos) around with you.

Internally a reference voltage of -10 volts governed all sections of the circuitry that handled wanted modulation. If you are wondering about the -10 volts (we'll come back to this later), let it be said that the whole circuitry is based on a distinctly unusual positive earth potential. This may be related in part to the germanium transistors used originally. For backward compatibility this could no longer be altered subsequently but it caused significant headaches for the burgeoning accessory industry. A simple contact with another device could lead to short circuits too, given that most of these were wired with negative ground. This would happen when an audio cable with its screen grounded at each end was plugged in... Electromechanically there are no major surprises. The fully encapsulated capstan motor had three electronically regulated speeds offering 3.75, 7.5 and 15 inch/s (9.5, 19 and 38 cm/s). A sophisticated drive and friction mechanism operated the feed-out and take-up reels using, as already mentioned, a single motor (**Figure 3**). At full voltage this could rewind fairly rapidly, also spool forward

at double speed. Special oils and greases, which could be applied on request by service dealers, made operation feasible at environmental temperatures between -20 and +50 degrees C (!) (-4 to +122 degrees F).

The device illustrated, the Nagra 4.2 from the 1970s, optically identical with the first machines, represents the pinnacle of technical development for the series. On the front panel we have, besides the self-explanatory control elements on the right-hand side two so-called rotary indicators, normally known as star indicators (**Figure 4**). These provided a confidence check of the operating parameters and consumed only a fraction of an indicator lamp's current.

The entire electronics were constructed using discrete components. Good audio ICs were still under development then and had a poor reputation for reliability, distortion and noise. Only years later was a model built using ICs. The inputs and outputs used balanced transformers, whilst the internal amplifier stages were asymmetric. A special feature was the floating balanced output of 4.4 volts amplitude, an order of magnitude required by the postal telecomms authorities in those days for transmission lines (**Figure 5**).





The device was constructed in modular fashion, which was pioneering for the time, with a motherboard on which all modules were arranged. Everything plugged in, using locking gold contacts (**Figure 6**).

Incidentally, if the white automobile fuse at the upper centre upsets the trained eye of the electronic engineer, this is actually not original. It's there because Mr. Kudelski for all his genius did forget one thing: the fuse...

The team paid particular attention to the indicating instrument. Here they had to take into consideration the varying norms or conventions for indication. The 'Modulometer' fulfilled these requirements perfectly, in which the criteria for accelerating the pointer needle and handling the inertia of the indication were combined brilliantly (**Figure 7**).

Metrologically all three recording speeds were adjustable individually for frequency response, equalized separately for standard and low-noise distortion correction. Even the HF symmetry and the distortion factors H3 and H4 could be optimised.

Depending on the device we have achieved frequency ranges from 20 Hz in part up to 18 kHz -2 dB. At an unweighted signal-to-noise ratio of -60 dB, this could rival any studio machine. The recorder provided two microphone inputs and additional user-modifiable inputs (**Figure 8**). The mike inputs were implemented either with input transformers or as electronically balanced current amplifiers. On the earliest models you had to plug in the preamplifier specific to the type of microphone (dynamic high and low impedance, condenser mike, etc.) (**Figure 9**). Later the preamp was made switchable.

The inputs were adjustable to an extremely high degree, meaning that the switchable internal pre-attenuation scarcely needed to be used. In conjunction with a plug-in limiter and a peak level limiter having a 'knee' function, it was hardly possible to overload the recording. You also had so-called magnetic 'tape saturation', which was able to 'stomach' the additional peaks.

A problem arose with the rather antiquated but universally used phantom power feed voltage of 48 volts. At that time it was still technically impossible to generate this 48 V operating voltage from the approximately 18 volts available. Naturally Nagra was aware of DC-DC converters but the square or triangular waveforms employed spread too far into the wanted signal. (In any case the converter was located, for reasons of modularity, at the most sensitive point in the chain: the microphone input.)

Here the firm Sennheiser assisted, bringing to market the then revolutionary HF condenser microphones for low voltage operation. Seen against 12-volt phantom supplies, the 10 volt 'T-Power' feed along the audio conductors (T stands for Tonaderspeisung [1]; Poland is right next to Germany) appeared like the Light of the World (presumably the reason for the internal 10 V reference voltage). For reasons that cannot be fathomed the 12 V phantom version sank without a whimper, whereas T-Power has held out a long time. Unfortunately this is not without some vagaries.

In this system, as its German name implies, the operating voltage is fed symmetrically along the two audio conductors, which indeed causes no interference to the wanted signal. That said, voltage-regulated amplifiers do not enjoy having a DC voltage on their input at all.

The special current inputs on the Nagra were fairly indifferent to this but the voltage inputs needed to have this DC voltage blocked with two capacitors. Voila: with the combination of the input resistance and the capacitor you have created a filter! Unfortunately, in terms of frequency range, this is definitely only a compromise solution.

If you accidentally connect a dynamic microphone to a T-Power input, all the bass notes disappear, because the DC voltage magnetizes the moving coil... But somehow we have tolerated these little problems.

Back to the recording apparatus once more. On the top side we see another audio head between the record and playback heads. This was the most vital element for all sync purposes: the pilot tone head (**Figure 10**).

To expand on this: in order to exclude any sync variations in studio operation, for instance from drift, the sep-mag (perfo) sound track material had precisely the same format and perforations as the film stock [2] (**Figure 11**). The film and

sound follower machines are driven by so-called synchronous motors, whose rotational speed is governed not by voltage but by frequency. Consequently the rotational speed of these motors is identical and synchronous, hard-locked to the AC line frequency.

The film cameras (mostly battery-driven) contained a small frequency generator that for 24 frames per second (later 25 fps) delivered a nominal speed of exactly 50 Hz. If the camera speed varied (resulting from many and various mechanical parameters), the frequency altered analogously. This frequency range was recorded in the middle of the sound track using the aforementioned pilot tone head.

To avoid any audible audio hum the slot of this sound head was rotated by 90 degrees, which eliminated the tone in the wanted signal by virtually 100 percent.

When transferring a tape recording to the AC line-driven sound follower machine, this frequency was compared against the line frequency and used to readjust electronically either the replay machine (in our case the Nagra) or the recording machine. For this purpose the Nagra provided the ability to make corrections of up to plus or minus 4 percent using the so-called Synchronizer. For greater values suitable accessories could be used, which happened only in extreme cases, however. An additional function related to this was the so-called 'Playback Procedure', which was employed for vocal numbers for example. Prior to starting, the AC line frequency was recorded onto the tape as a pilot tone. The Nagra now received one pilot tone from the camera, compared this with the other tone coming from the tape, leaving the Synchronizer to correct the difference. This system proved successful for decades.

The interface in the form of various plug connectors was truly 'open system' and the range of accessory devices was infinite, from large 3,000-foot tape reels, through mixing desks and crystal control devices, to wire free applications. The initial purchase price was definitely high for the period in question, the fully equipped version (without carrying case and mains power supply) costing the current equivalent of \$10,000. Three gadgets should not go unmentioned: the tape-length counter incorporated in the self-aligning roller, the stroboscope roller for speed control and a mechanical tape cleaner in the form of a metal shim (**Figures 12 and 13**).

For a machine of this calibre there were nat-

urally appropriate headphones. At the time only the firm Beyer Dynamic could provide the most linear model, the so-called 'Normal-Telefon' (reference standard 'phones) with 2 x 25 ohms. In modified form these headphones remained in use for more than 50 years!

This recorder was a mono device (see **inset**). Later the stereo version followed of course and in the early days of the video era a version with timecode. At the end of the 1980s digital technology was still in its infancy but already in 1992 there appeared a digital version, which recorded four PCM tracks on standard tape and this with an unbelievable 24 bits!

Naturally development continues to move forward; today for instance there is a Nagra with flash memory and all the parameters necessary for modern digital sound recording!

(120570)

Internet Links

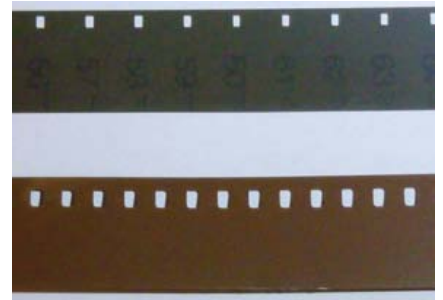
[1] www.ips.org.uk/faq/index.php?title=Tonader_Power

[2] http://en.wikipedia.org/wiki/Sound_follower

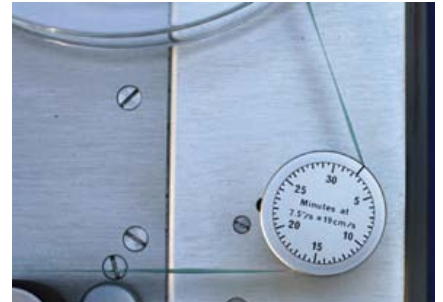
Retronics is a monthly section covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcome, please telegraph editor@elektor.com.

Why mono?

In case anyone is wondering why you would wish to use a mono device, here's the explanation. If you wish to record authentic speech with a stereo microphone or two spatially separated mikes, the sound source needs to occupy a defined position in the virtual acoustic listening room. If you now, as is common in films, alter the pick-up angle or the distances within the scene, effectively this creates a new position in the room. Human hearing registers spatial changes with great accuracy, which is why an edit made after this move could cause listeners/viewers irritation. A dissolve or cross-fade would not to disguise this change either. Therefore in practice the (mono) speech does not gain its final position in the room until the sound mixing stage. The end product is perfected by mixing the speech together with multi-channel 'effects' sound recorded spatially.



11



12



13

Hexadoku Puzzle with an electronic touch

In the rare case of not being out about buying presents or clearing snow (Aussies: BBQ'ing on the beach), why not relax for a while with this new Hexadoku brain teaser.

Enter the right numbers or letters A-F in the open boxes, find the solution in the grey boxes, send it to us and you automatically enter the prize draw for one of four Elektor Shop 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 grey boxes**.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for one Elektor Shop voucher worth £80.00 and three Elektor Shop Vouchers worth £40.00 each, which should encourage all Elektor readers to participate.

Participate!

Before January 1, 2013,

send your solution (the numbers in the grey boxes) by email to hexadoku@elektor.com

Prize winners

The solution of the October 2012 Hexadoku is: **75E2B**. The Elektor £80.00 voucher has been awarded to Steve Hasko (UK).

The Elektor £40.00 vouchers have been awarded to Michael Franke (Germany), Jean-Paul Lagaisse (Belgium), and Franz Maurer (Netherlands).

Congratulations everyone!

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

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

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

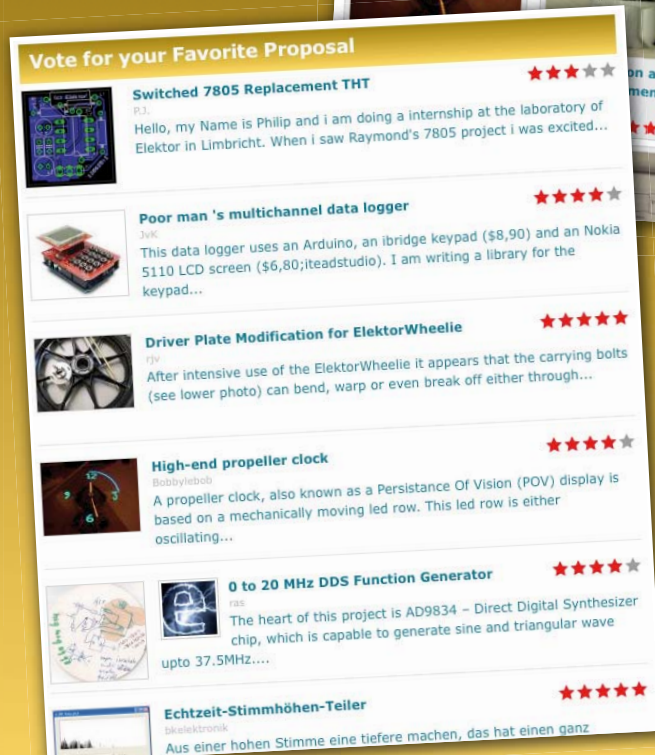
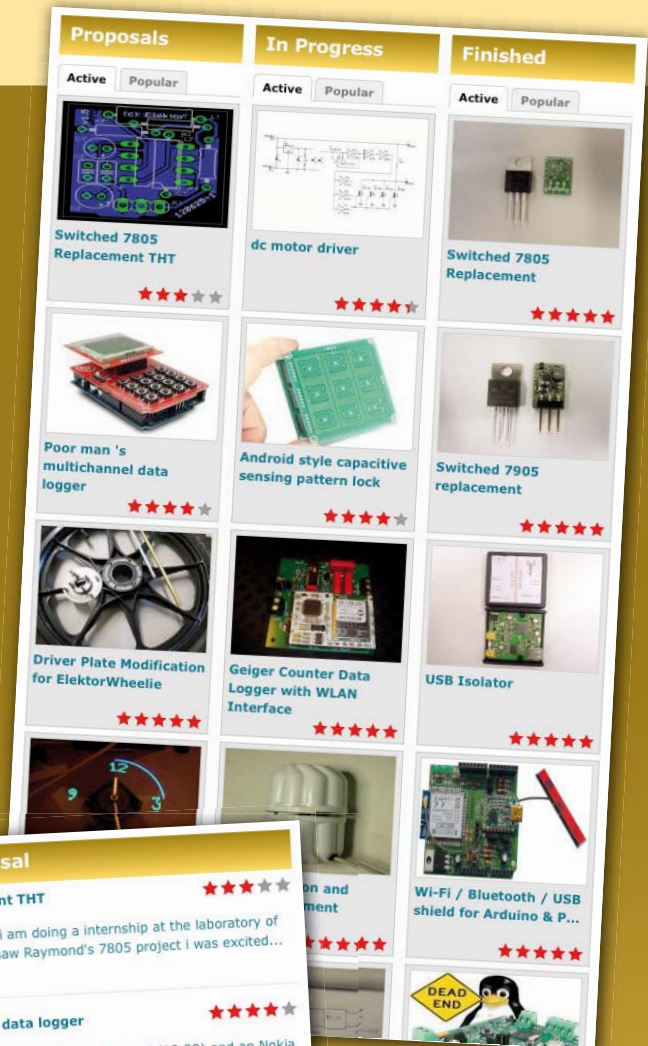
elektor@labs

Sharing Electronics Projects

Elektor.LABS is an online community for people passionate about electronics. Here you can share your projects and participate in those created by others. It's a place where you can discuss project development and electronics.

Elektor's team of editors and engineers assist you to bring your projects to a good end. They can help you write an article to be published in Elektor.MAGAZINE or even develop a complete product that you can sell in Elektor.STORE!

**GET
ELEKTORIZED**

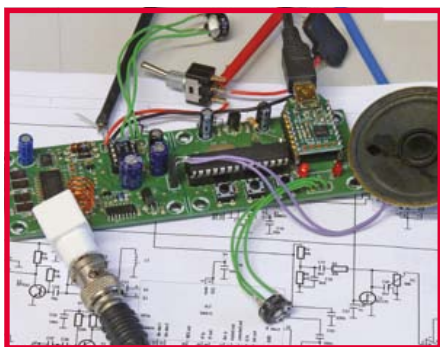


Join or Create a Project at www.elektor-labs.com



Next month in Elektor

Jumbo size January & February 2013 edition with a ton of ideas, tips, and projects



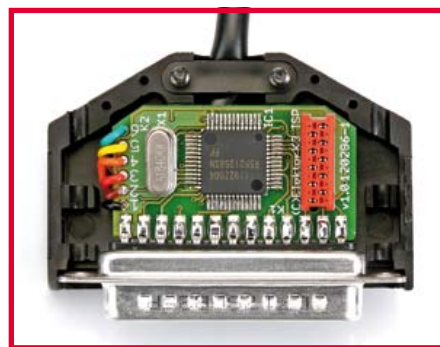
VHF Airband Scanner

Many aircraft enthusiasts have great interest in following the voice communication between aircraft and air traffic control towers. With this receiver, you can tune into all traffic in the 108-137 MHz VHF civil aviation band. It is easy to build, has only one mechanical adjustment and can even be operated from a PC with a USB connection. A single conversion VHF-FM receiver IC is used, which is tweaked to function as an AM receiver by clever use of the signal strength detector output.



Mini Multitester

You're busy with a microcontroller project and all sorts of things should be measured, such as output voltages, logic levels, pulses count and so on. What instrument to use? With this handy multi tester you have a versatile measuring device that combines a large number of functions in a small enclosure that you can stick with one hand. The device acts as a logic tester, voltmeter, pulse counter, TTL/UART receiver and transmitter, frequency meters and much more.



USB I/O Interface Cable *

Surely it's useful to employ the USB connection for your own measurements and control purposes. We developed an interface cable based on a USB-TTL cable from FTDI, to which a small PCB with an R8C microcontroller is added for fitting into a DB25 plug housing. The controller handles the communication with the PC, and at the output side offers 24 pins for the user to configure and do a variety of measuring and switching tasks.

** We regret that this article could not be published in the current issue as planned, also for 'USB I/O Interface Cable'.*

Article titles and magazine contents subject to change; please check the Magazine tab at www.elektor.com

Elektor USA January & February 2013 edition published January 9, 2013.

See what's brewing
@ Elektor Labs 24/7

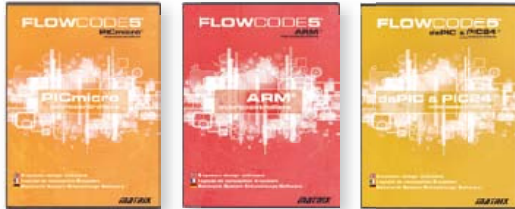
Check out
www.elektor-labs.com
and join, share, participate!

The screenshot shows the Elektor Labs website interface. At the top is the 'elektor labs' logo with the tagline 'Sharing Electronics Projects'. Below the logo is a navigation bar with links: Home, News, Proposals, In Progress, and Finished. A search bar and a 'Log in' button are also present. The main content area features a large banner for 'wind speed and direction measurement'. Below this, there are three columns: 'Proposals', 'In Progress', and 'Finished'. Each column has a sub-header 'Active' and 'Popular'. The 'Proposals' column shows a project titled 'Switched 7805 Replacement THT' with a 5-star rating. The 'In Progress' column shows a project titled 'dc motor driver' with a 5-star rating. The 'Finished' column shows a project titled 'Switched 7805 Replacement' with a 5-star rating. On the right side, there is a section titled 'About Elektor.LABS' with a video player. Below that is a 'Create a Project' section with a link 'Create a new project or enter a proposal' and a description: 'Get help, feedback & votes from other visitors, and maybe you will get Elektorized too!'. There is also a 'Challenges' section with a link 'Win a PIC & Development Kit! All challenges...'. At the bottom, there is a 'Join a Project' section with a link 'Switched 7805 Replacement THT'.

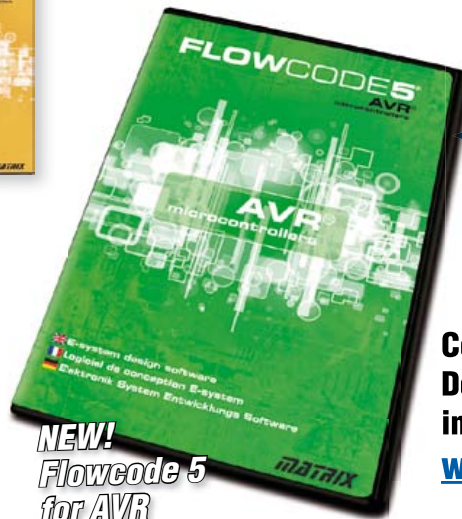
Create complex electronic systems in minutes using Flowcode 5

FLOWCODE5

Design → Simulate → Download



Flowcode is one of the World's most advanced graphical programming languages for micro-controllers (PIC, AVR, ARM and dsPIC/PIC24). The great advantage of Flowcode is that it allows those with little experience to create complex electronic systems in minutes. Flowcode's graphical development interface allows users to construct a complete electronic system on-screen, develop a program based on standard flow charts, simulate the system and then produce hex code for PIC AVR, ARM and dsPIC/PIC24 microcontrollers.



NEW!
Flowcode 5
for AVR

Convince yourself.
Demo version, further
information and ordering at
www.elektor.com/flowcode

%%%

HAMEG
Instruments
A Rohde & Schwarz Company

hameg.com/xmas2012

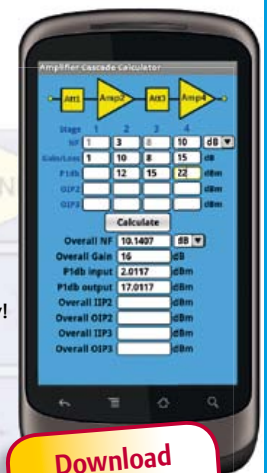
Elektor RF & Microwave Toolbox

→ for Android

The RF & Microwave Toolbox contains 34 calculation and conversion tools for RF, microwave and electronics in general. Whether you are an RF professional, radio-amateur, astronomer or hobbyist, this app puts some of the most important tools right at your fingertips. Literally!

Highlights:

- Amplifier cascade (NF, Gain, P1 db, OIP2, OIP3)
- Field intensity and power density converter (W/m², V/m, A/m, Tesla, Gauss, dBm, W)
- PCB Trace calculator (impedance/dimensions)
- PI and T attenuator
- Antenna temperature (Kelvin)
- EMC (EIRP, ERP, dBμV/m)
- Filter Design (Butterworth, Chebyshev, prototype)
- And much more



Download
your app now!



Further information at

www.elektor.com/rf-app

Ultrasonic Distance Sensing Made EZ

www.maxbotix.com

USB-ProxSonar®-EZ™

- Great for Kiosks applications
- USB interface
- Easy integration
- 1 inch resolution
- MSRP \$49.95



I2CXL-MaxSonar®-EZ™

- Incredible noise immunity
- I2C interface
- 1cm resolution
- UAV's and robotics
- Automatic calibration
- Starting at \$39.95



LV-ProxSonar®-EZ™

- People sensing made easy
- Multi-sensor operation
- Calibrated detection zone
- Low power requirements
- Easy to use
- MSRP \$29.95





The world's first book with NFC technology integrated inside

1 **Catch the Sun**

The oldest known contactless connectivity technology dates back 2000 years to the Han dynasty in China. In that era, the Kongming lantern was invented: a small hot air balloon used primarily for transmitting military signals. The Kongming balloons have today been replaced by chips. Near Field Communication, or NFC, provides wireless connectivity over short distances based on semiconductor technology. This book links both technologies together.

Catch the Sun is the world's first book with NFC semiconductor technology integrated inside, while the content of this high-tech book is about the beautiful magic of low-tech ballooning. The book has multiple NFC chips inside that allow the book to connect to the internet, simply by touching an NFC-hotspot in the book with your NFC-enabled smartphone or tablet.

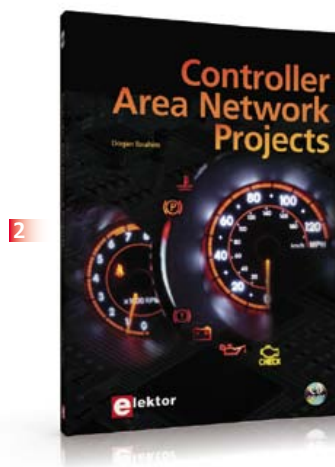
128 pages • ISBN 978-9-07545-861-9

£35.50 • US \$57.50

Free mikroC compiler CD-ROM included

2 **Controller Area Network Projects**

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of



microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters.

260 pages • ISBN 978-1-907920-04-2

£29.50 • US \$47.60

More than 75,000 components

3 **CD Elektor's Components Database 7**

This CD-ROM gives you easy access to design data for over 11,100 ICs, 37,000 transistors, FETs, thyristors and triacs, 25,100 diodes and 2,000 optocouplers. The program package consists of eight databanks covering ICs, transistors, diodes and optocouplers.

A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data.

ISBN 978-90-5381-298-3 • £24.90 • US \$40.20



LabWorX 2

4 **Mastering Surface Mount Technology**

This book takes you on a crash course in techniques, tips and know-how to successfully introduce surface mount technology in your workflow. Even if you are on a budget you too can jumpstart your designs with advanced fine pitch parts. Besides explaining methodology and equipment, attention is given to SMT parts technologies and soldering methods. Many practical tips and tricks are disclosed that bring surface mount technology into everyone's reach without breaking the bank. A comprehensive kit of parts comprising all SMT components, circuit boards and solder stencils is available for readers wishing to replicate three projects described in this book.

282 pages • ISBN 978-1-907920-12-7

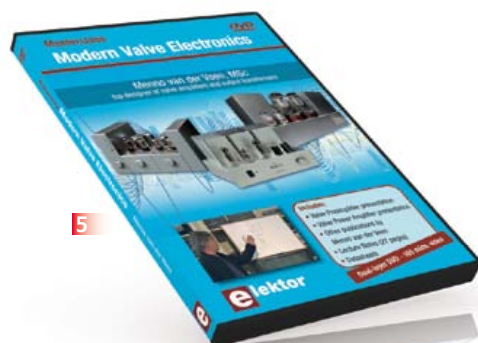
£29.50 • US \$47.60

Dual-layer DVD: 165 mins. video

5 **DVD Modern Valve Electronics**

This filmed seminar (presented by Menno van der Veen) starts with a short discussion of the classic approach using valve load line graphs, followed by current sources and current foldback techniques. Next, the negative effect of cathode electrolytics is covered as well as reducing supply

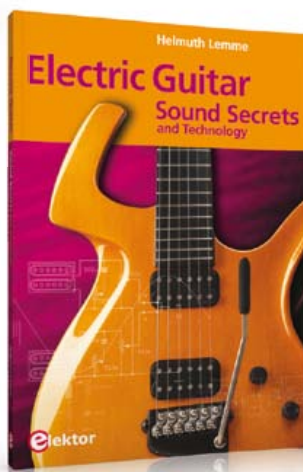
Books, CD-ROMs, DVDs, Kits & Modules



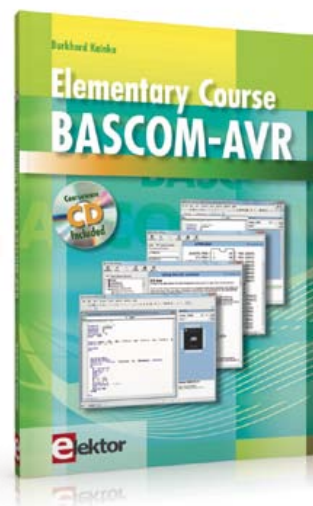
5



6



7



8



9

voltage interference. With the help of state of the art measurement techniques the (in)correctness of feedback is proven, while also clarifying what's happening deep within the core of the output transformer.

ISBN 978-1-907920-10-3 • £24.90 • US \$40.20

Avoid interference and earth loops

6 USB Isolator

If your USB device ever suffers from noise caused by an earth loop or if you want to protect your PC against external voltages then you need a USB isolator. The circuit described in Elektor's October 2012 edition offers an optimal electrical isolation of both the data lines as well as the supply lines between the PC and the USB device.

Populated and tested Board

Art.# 120291-91 • £62.30 • US \$101.40

Sound Secrets and Technology

7 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 forty years. Their underlying sound is determined largely by their electrical components. But, how do they actually work? This book answers many questions simply, in

an easily-understandable manner. For the interested musician (and others), this book unveils, in a simple and well-grounded way, what have, until now, been regarded as manufacturer secrets. The examination explores deep within the guitar, including pickups and electrical environment, so that guitar electronics are no longer considered highly secret. With a few deft interventions, many instruments can be rendered more versatile and made to sound a lot better – in the most cost-effective manner.

287 pages • ISBN 978-1-907920-13-4

£29.50 • US \$47.60

Free Software CD-ROM included

8 Elementary Course BASCOM-AVR

The Atmel AVR family of microcontrollers are extremely versatile and widely used. In Elektor magazine we have already published many interesting applications employing an ATmega or ATtiny microcontroller. The majority of these projects perform a particular function. In this book we focus more on the software aspects. Using lots of practical examples we show how, using BASCOM, you can quickly get your own design ideas up and running in silicon.

224 pages • ISBN 978-1-907920-11-0

£34.95 • US \$56.40

Meet BOB

9 FT232R USB/Serial Bridge/BOB

You'll be surprised first and foremost by the size of this USB/serial converter – no larger than the moulded plug on a USB cable! And you're also bound to appreciate that fact that it's practical, quick to implement, reusable, and multi-platform – and yet for all that, not too expensive! Maybe you don't think much of the various commercially-available FT232R-based modules. Too expensive, too bulky, badly designed, That's why this project got designed in the form of a breakout board (BOB).

PCB, assembled and tested

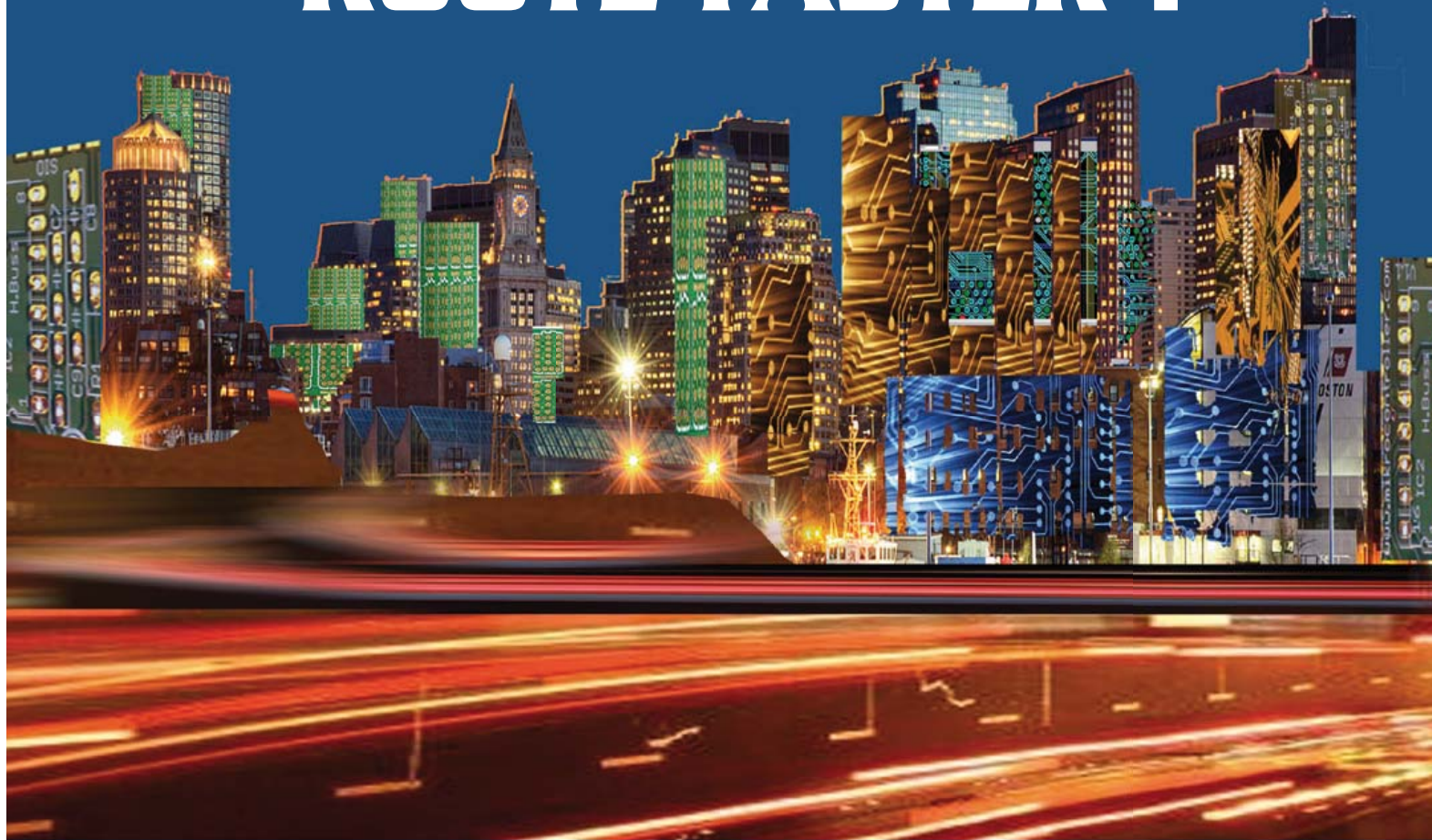
Art.# 110553-91 • £12.90 • US \$20.90

More information on the
Elektor Website:

www.elektor.com/store

Elektor
78 York Street
London - W1H 1DP
United Kingdom
Tel.: +44 20 7692 8344
Email: order@elektor.com

ROUTE FASTER !



WITH PROTEUS PCB DESIGN

Our completely new manual router makes placing tracks quick and intuitive. During track placement the route will follow the mouse wherever possible and will intelligently move around obstacles while obeying the design rules.

All versions of Proteus also include an integrated world class shape based auto-router as standard.

PROTEUS DESIGN SUITE **Features:**

- Hardware Accelerated Performance.
- Unique Thru-View™ Board Transparency.
- Over 35k Schematic & PCB library parts.
- Integrated Shape Based Auto-router.
- Flexible Design Rule Management.
- Polygonal and Split Power Plane Support.
- Board Autoplacement & Gateswap Optimiser.
- Direct CAD/CAM, ODB++, IDF & PDF Output.
- Integrated 3D Viewer with 3DS and DXF export.
- Mixed Mode SPICE Simulation Engine.
- Co-Simulation of PIC, AVR, 8051 and ARM7.
- Direct Technical Support at no additional cost.

Prices start from just £150 exc. VAT & delivery

labcenter  www.labcenter.com
Electronics

Visit our website or
phone 01756 753440
for more details

Labcenter Electronics Ltd. 53-55 Main Street, Grassington, North Yorks. BD23 5AA.
Registered in England 4692454 Tel: +44 (0)1756 753440, Email: info@labcenter.com