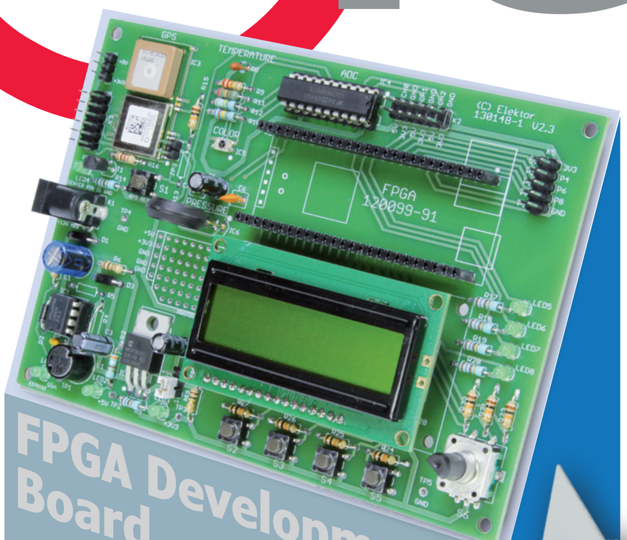
www.elektor-magazine.com

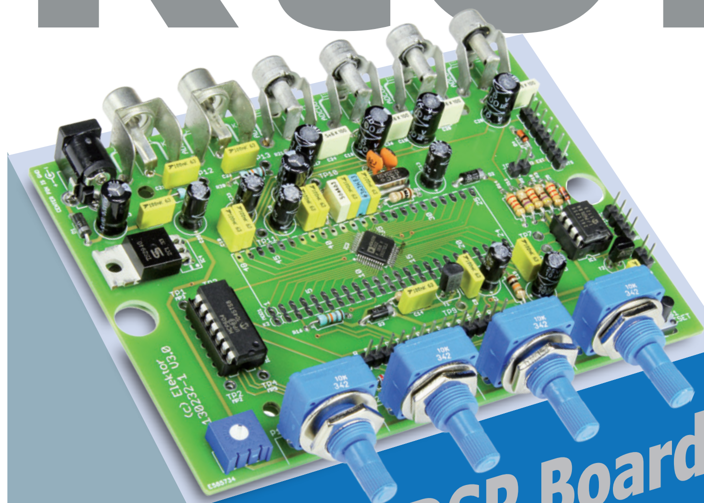
.magazine

January & February 2014

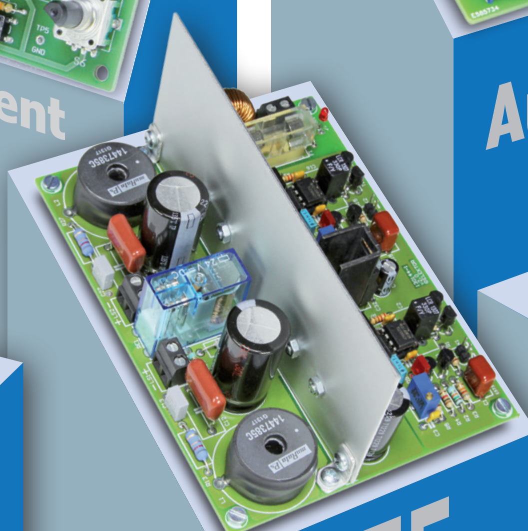
elektor



FPGA Development Board



Audio DSP Board



555 Class-D Amp

LED's Replace GU4 Halogen

Compact Tube Amplifier

EXTRA-THICK EDITION

132 PAGES OF TOP-CLASS DIY ELECTRONICS

US $14.00 - Canada $14.00

01

WHERE THE CHIP MEETS THE BOARD

DesignCon® 2014

DesignCon 2014 is the premier event dedicated to the chip, board and systems design engineering community involved in high-speed design with a focus on the pervasive nature of signal integrity at all levels of electronic design.

- Explore **100+ Technical Tutorials**, Panels & Case Studies for the latest design theories, methodologies and applications
- Meet **150+ Exhibitors** showcasing advanced design tools & solutions to your biggest SI, PI & verification challenges
- Attend **Keynotes**: Prof. Dr. Hermann Eul, VP & GM, Mobile Communications Group, Intel & J. Thomas Pawlowski, Chief Technologist & Fellow, Micron
- Share **ideas & insights** with peers & industry experts at the: Welcome Reception, DesignTOUR, Happy Hours, Chiphead Theater, Awards Programs & more

CONFERENCE: January 28-31 EXPO: January 29 & 30
Santa Clara Convention Center, Santa Clara, CA

REGISTER AT WWW.DESIGNCON.COM

Does your design need class-leading 32-bit MCU performance, code density and ADC throughput?

PIC32MZ offers 330 DMIPS and 3.28 CoreMarks™/MHz performance

Microchip's new PIC32MZ 32-bit MCUs achieve class leading performance, combined with ADC throughput of 28 Msps, 30% better code density and up to 2 MB dual-panel Flash with live update and 512 KB RAM.

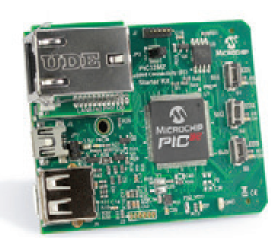
The PIC32MZ Embedded Connectivity (EC) family of 32-bit MCUs introduces a breakthrough in high-end embedded control with its class-leading performance and code density in addition to new levels of on-chip memory and peripheral integration.

With up to 2 MB of dual-panel Flash and 512 KB of RAM, the PIC32MZ offers 4x more on-chip memory than any other PIC® MCU, with fail-safe operation during live Flash updates. It is also the first PIC MCU to use the enhanced MIPS microAptiv™ core which adds 159 new DSP instructions that enable the execution of DSP algorithms at up to 75% fewer cycles than the PIC32MX families.

Advanced connectivity is supported over Hi-Speed USB, 10/100 Ethernet and two CAN 2.0b modules as well as multiple UART, SPI/I^{2}S, and I^{2}C channels. The optional on-chip crypto engine ensures secure communication with a random number generator and high-throughput data encryption/decryption and authentication.

DEVELOPMENT TOOL SUPPORT:

- Turn-key PIC32MZ EC Starter Kits
- Multimedia Expansion Board II
- PIC32MZ2048EC Plug-in Module for Explorer 16



PIC32MZ Embedded Connectivity Starter Kit (DM320006 or DM320006-C with crypto engine)

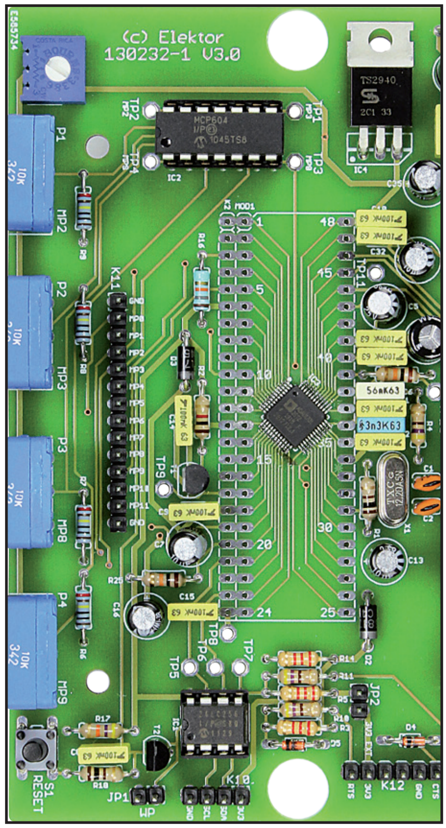
For more information, go to: www.microchip.com/get/eupic32mz

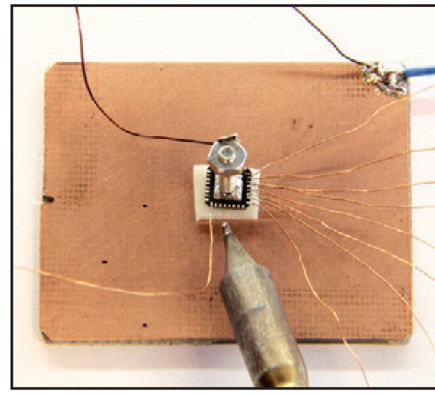
MICROCHIP

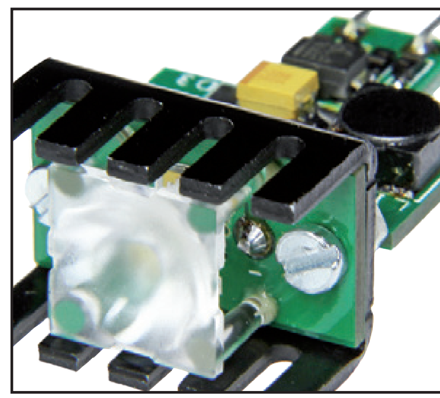
Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

The Microchip name and logo, and PIC are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks mentioned herein are the property of their respective companies.
©2013 Microchip Technology Inc. All rights reserved. DS60001247A. ME1092Eng11.13

Contents

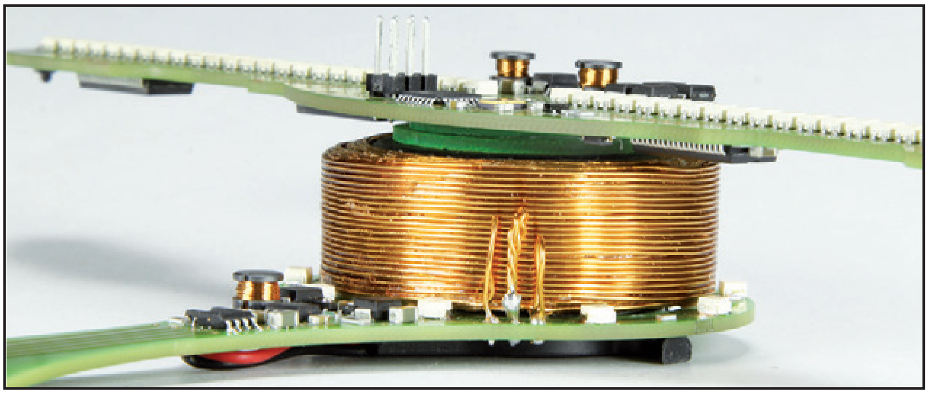


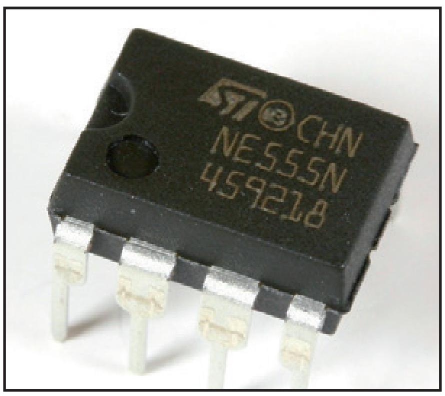


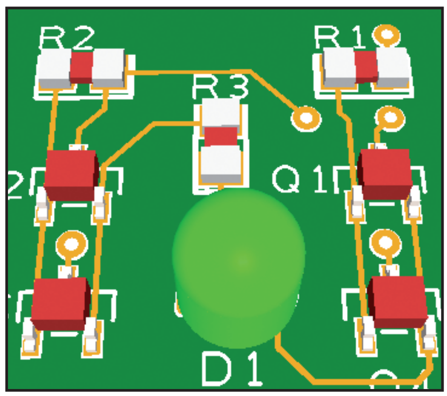


Volume 40 – No. 445 & 446

January & February 2014







Labs

Industry

Regulars

•Community

Volume 40, No. 445 & 446
January & February 2014

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

Elektor Magazine is published 10 times a year including double issues in January/February and July/August, concurrently by

Elektor International Media
111 Founders Plaza, Suite 300
East Hartford, CT 06108, USA
Phone: 1.860.289.0800
Fax: 1.860.461.0450

and

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

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

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

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

USA / Canada Advertising:
Peter Wostrel
Phone: 1.978.281.7708
E-mail: peter@smmarketing.us

UK / ROW Advertising:
Johan Dijk
Phone: +31 6 15894245
E-mail: j.dijk@elektor.com

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

Copyright Notice
The circuits described in this magazine are for domestic and educational use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, DVDs, 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 Publisher disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from schematics, descriptions or information published in or in relation with Elektor magazine.

© Elektor International Media b.v. 2014
Printed in the USA Printed in the Netherlands

132 pages—and not on a diet

I used to think it was hard to prevent total dominance in Elektor magazine of articles chewing over projects, technologies or programming methods related to microcontroller and embedded technologies, and with our new "kitchen" website www.elektor-labs.com well up and running I had reason to fear an overwhelming presence of bit crunchers and compilers. Yet, looking at the *Proposals* and *In Progress* columns on the labs website there is a reassuring number of projects and project ideas that are suitable for microcontroller-free diets. Three areas in particular, audio, radio, and test & measurement, seem to attract designs produced without any PICs, AVRs, ARMs, or Cortex-Xs acting as black boxes. That does not mean the projects are totally free from Raspberry Pis or Arduinos though, as these and similar boards are now (rightly) considered "just another useful component" rather than an intricate microcontroller in a cloud of code and protocols.
This extra thick edition of Elektor magazine intends to get you through the two first winter months of 2014 while attempting to give analog en non-microcontroller projects their due share of pages. To kick off, there's our DSP-without-the-math on page 12. There are two magnificent audio amplifiers, one 555 based (page 48) and one with tubes (page 26). The amps have low cost, ease of assembly, and vintage components in common—and by no means aspire to reach the realms of High End Audio. We could not resist though grilling them on our Audio Precision distortion analyzer. Besides the 555 another old faithful, the 2N3055, hails from the past—secured to the bottom of a tomato paste tin, on page 100 it's helping to serve a drink of water to birds of the feathered variety during the winter months. The RJ45 Cable Tester on page 38 also excels in not having a microcontroller. Nanoamps current measurements (page 98) and current pulses (page 87) are covered in two separate articles for the T & M fans among you. The latter article is part of a series due for continuation in the March 2014 edition. Thumbs up for that AC power supply from 1984 (page 122). Diets if applied should be moderate and balanced—we serve dishes with and without the opcode ingredient.
Enjoy!

Jan Buiting, Editor-in-Chief

The Team

Editor-in-Chief:	Jan Buiting
Publisher / President:	Carlo van Nistelrooy
Membership Managers:	Shannon Barraclough (USA / Canada), Raoul Morreau (UK / ROW)
International Editorial Staff:	Harry Baggen, Jaime Gonzalez Arintero, Denis Meyer, Jens Nickel
Laboratory Staff:	Thijs Beckers, Ton Giesberts, Wisse Hettinga, Luc Lemmens, Mart Schroijen, Clemens Valens, Jan Visser, Patrick Wielders
Graphic Design & Prepress:	Giel Dols
Online Manager:	Daniëlle Mertens
Managing Director:	Don Akkermans

USA
Carlo van Nistelrooy
+1 860-289-0800
c.vannistelrooy@elektor.com

United Kingdom
Carlo van Nistelrooy
+44 20 7692 8344
c.vannistelrooy@elektor.com

Germany
Ferdinand te Walvaart
+49 241 88 909-17
f.tewalvaart@elektor.de

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

Netherlands
Ferdinand te Walvaart
+31 46 43 89 444
f.tewalvaart@elektor.nl

Spain
Jaime Gonzalez Arintero
+34 6 16 99 74 86
j.glez.arintero@gmail.com

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

Sweden
Carlo van Nistelrooy
+31 46 43 89 418
c.vannistelrooy@elektor.com

Brazil
João Martins
+31 46 4389444
j.martins@elektor.com

Portugal
João Martins
+31 46 4389444
j.martins@elektor.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
zkoksal@beti.com.tr

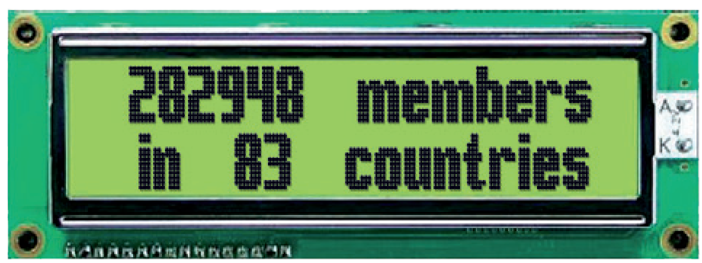
South Africa
Johan Dijk
+31 6 1589 4245
j.dijk@elektor.com

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

Our Network

elektor
VOICE COIL
circuit cellar
hifi.nl MUZIEK - FILM - LIFESTYLE
audioxpress
techthe future

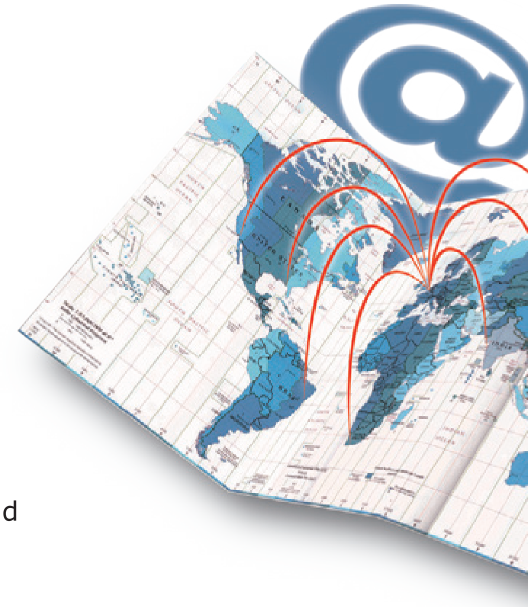
Connects You To



Supporting Companies

Not a supporting company yet?

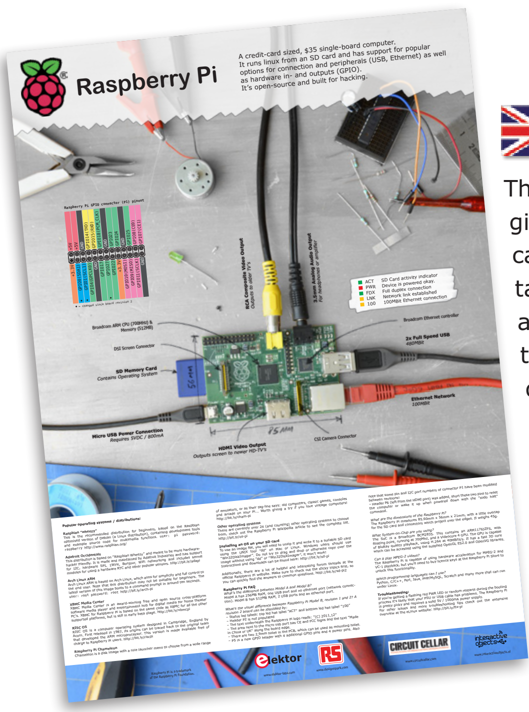
Contact Peter Wostrel (peter@smmarketing.us, Phone 1 978 281 7708,
to reserve your own space in Elektor Magazine, Elektor•POST or Elektor.com



Elektor World

Compiled by
Wisse Hettinga

Every day, every hour, every minute, at every given moment designers and enthusiasts are thinking up, tweaking, reverse-engineering and developing new electronics. Chiefly for fun, but occasionally fun turns into serious business. Elektor World connects some of these events and activities — for fun and business.



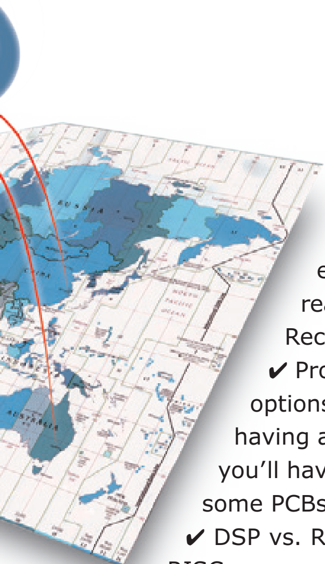
Postermania

These are interesting times for the publishing world. The new 'online and content' are giving the old, established media a run for their money, and who still believes you can make a decent profit from publishing something as old-fashioned as a book? Or take posters—in the old days Elektor often produced posters showing, for example, a bunch of connectors. You can still find them in classrooms. Is it just ancient history? To our surprise the Raspberry Pi poster did rather well earlier this year. The design was downloaded over 35,000 times and as you are reading this an Arduino poster is being made available—and there's more coming. And everything is downloadable. I know what you are thinking: "So, back to online anyway?" And yes, everything is downloadable—arguably the easiest way to deliver to your door.

The Elektor Studio

There are a few places in the Elektor offices that stand out. We have the permanent wire & meter mess of the Lab, the 'dungeons' (what secrets the walls could tell us...) and we have our attic. For years the attic remained unused and unloved, gathering dust. We decided to bring out the brooms and convert the space into a real video & photo studio. From our attic studio we bring you our monthly webinars in cooperation with element 14. Video will be implemented more and more in the new workings of Elektor—we are well past our first lessons and mistakes, and are now creating video manuals for an increasing number of products. The number and form of the webinars will also be looked at—you can look forward to more even more interesting sessions from us.

Patrick Wielders will be our man behind the camera and will manage the studio.



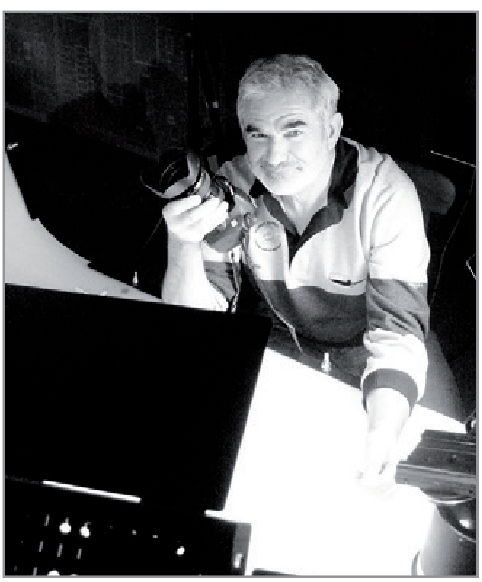
World Class Tips

For the past few months, Circuit Cellar magazine has been posting a weekly electrical engineering tip at CircuitCellar.com. The response has been excellent. Engineers from around the globe have been reading the tips and even submitting tips of their own. Recent tips have included:

✔ Prototyping Tips for Engineers: Pro engineer and CC columnist Jeff Bachiochi presents three prototyping options and discusses the pros and cons associated with each one. He writes: "There are two alternatives to having a PCB house manufacture your PCBs: do-it-yourself (DIY) and routing. If you choose a DIY approach, you'll have to work with ferric chloride (or another acid) to remove unwanted copper. You'll be able to produce some PCBs quickly, but it will likely be messy (and dangerous)." (http://bit.ly/1bB4a40)

✔ DSP vs. RISC Processors: Engineer Bernard Debbash covers a few fundamental differences between DSP and RISC processors. Some generic RISC processors like the NXP LPC2138 don't have a saturation function, he explains. "So it's important to ensure that the input values or the size of the variable are scaled correctly to prevent overflow. This problem can be avoided with a thorough simulation process." (http://bit.ly/1aBFPKb)

Visit the CircuitCellar.com website or follow the editorial team via Twitter (https://twitter.com/circuitcellar), Facebook (https://www.facebook.com/circuitcellar), and Google+, to get your weekly EE tips. Feel free to share you tips as well.



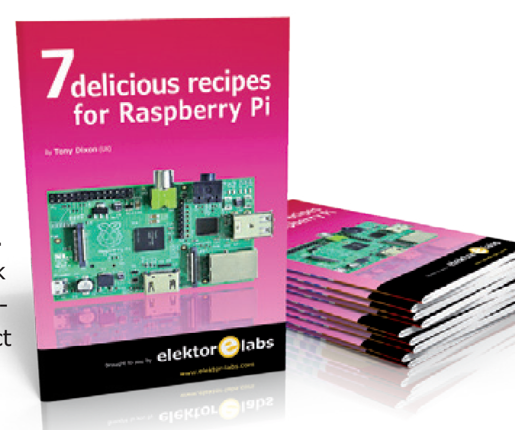
From Tracing Paper to Computer: Elektor Schematics Layout

If there is one characteristic of Elektor that stands out, it must be the way schematics are drawn. From the very start in the early 1970s Elektor has had a very specific way of schematic drawing, and the force behind this is... Mart Schroijen. For 36 years he has taken care of the drawings; initially using a set of Rotring pens and calque tracing paper, now it's all done with the help of computers and Patrick Wielders. Over the years Mart created thousands of drawings and also took many of the techy photographs appearing in the magazine.

The biggest advantage of the Elektor schematic drawing style is instant recognition. Sometimes we receive project proposals covering our own publications, or we spot illegal downloads of our published articles (fun Friday afternoon times). In all cases the Elektor style is instantly recognizable.

Raspberry Pi Cookbook

Last year Elektor members enjoyed a series of RPi projects and articles arriving via their Elektor.POST subscription every other week. The first seven projects have been compiled into a small cookbook allowing everyone to start off right away using a Raspberry Pi microcomputer. The booklet will be available to anyone launching a project on www.elektor-labs.com in January.



New Look for the Elektor Store

Since the start of the year, the familiar address www.elektor.com takes you to a fully revamped Elektor Store. The first thing you notice on the website is the fresh new layout based on the "less is more" principle. Thanks to Responsive Web Design technology, the site looks good on all types of devices, including computer monitors, tablets and even smartphones.

- However, what's even more important is that the search function has been dramatically improved—and we have to admit that it was not one of the major strengths of our previous online presence.
  The main menu gives you access to the following product categories: Books, Microcontrollers, PCBs, Kits & Modules, and even more. A new feature is that searches now depend on criteria specific to the selected product category. For example, you can search for microcontrollers based on the product number, but you can also restrict your search by selecting parameters such as operating voltage or package type. You can select a price range for searching in any product category, and if desired you can sort the results by price.
- Each product has its own page where all the product data is shown in an orderly manner. But that's not all: there you can also access pictures and user guides or manuals, and in many cases videos produced by the Elektor Labs end Editorial teams. The options for combined purchases and special offers are another attractive feature. That can save you a pretty penny! By the way, the range of payment options has been expanded and now includes MasterCard, Direct Debit (UK) and PayPal, to mention only the most important. However, we have dropped the Elektor Credits option. Regular and special editions of Elektor magazine can now be purchased directly in the store without the bother of credit units.
- The new Elektor Store at **www.elektor.com** no longer includes the well-known project pages. They have been moved to **www.elektor-magazine.com**. The project pages are the starting point for downloads for Elektor projects, such as software or PCB layouts in PDF format. You can access an individual project page quickly using the deep link **www.elektor-magazine.com/xxxxxx**, where "xxxxxx" is the six-digit article number that is always shown at the end of each article. However, the old links with the format www.elektor.com/xxxxxx still work; they are automatically redirected to the -magazine website. The forums are still available to Elektor readers as usual. You can access the reader forums via the link **http://forum.elektor.com**.
- Finally, a word about user accounts for the Elektor Store: if you have previously ordered something from the online store, you do not have to enter your data again.

Now it's time for us to stop talking and for you to see for yourself: simply surf to the new Elektor Store at www.elektor.com, where you are bound to find something attractive.

(130402-I)

Shop languages:

reichelt electronics

Your competent online partner for

Components
Power supply systems
Measuring technology
Shop & soldering technology
Home & security technology
Network technology
PC technology
Sat/TV technology
Communication

SUBSCRIBE NOW!

Newsletter in English

Receive fresh information every week on

- ✓ New products
- ✓ Best offers
- ✓ Reduced prices

Measuring technology – precise & inexpensive!

Professional Quality @ Discount Prices!

UNI-T

Digital memory oscilloscope

with outstanding performance data and USB interface!

- ✓ extra large full-colour display
- ✓ 25 MHz bandwidth
- ✓ 2 channels

- Scanning rate: 250 MS/s
- Rise time: <14 ns
- Storage depth: 25 kpts
- USB interface for export of readings

€ 249,00 (~ £ 209.41)
UTD 2025 CL

OUR BLOCKBUSTERS

UNI-T

Digital multimeter with USB interface

- LC display, backlit, 6000 counts
- Accuracy: 0.5 %
- Safety: CAT II 600 V, CAT III 300 V
- Measures resistance, frequency, capacitance, temperature as well as AC/DC current & voltage

UT 61C € 49,95 (~ £ 42.00)

Infrared
Thermometer

- Measurement range: –18 to 380 °C
- Distance/measurement point ratio: 10:1
- Preset emissivity: 0.95
- Resolution: 0.1 °C
- Repeat accuracy: ±0.5 °C or ±0.5 %
- Measurement accuracy: ±2 °C/±2 %

UT 300B € 34,95 (~ £ 29.39)

AC/DC digital
Clip-on measuring device

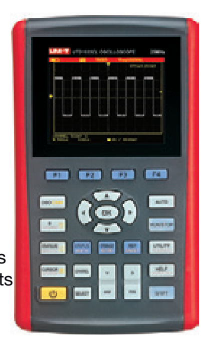
- Up to 400 A
- Clip: Ø 28 mm
- 3-¾ digit LC display, 3999 counts
- Safety: IEC 61010, CAT II 600 V, CAT III 300 V

UT 203 € 39,95 (~ £ 33.59)

UNI-T

Hand-held
Digital memory oscilloscope

Combines the functions of an oscilloscope and a digital multimeter.



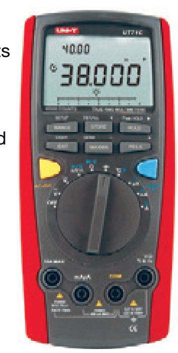
- 1-channel DSO with 25 MHz analog bandwidth
- Scanning rate: 200 MS/s (real time)
- High-resolution LC display, 8.89 cm/3.5", 320 x 240 pixels
- Digital multimeter: 4000 counts
- USB interface, software included

UTD 1025 CL € 295,95 (~ £ 248.89)

UNI-T

True effective value
Digital multimeter

- Backlit LC display, 40,000 counts
- Accuracy: 0.025 %
- Safety: CAT III 1000 V, CAT IV 600 V
- USB interface, software included
- Data logger function: 100 readings
- Measures resistance, capacitance, frequency, temperature as well as AC/DC current and voltage



UT 71C € 135,00 (~ £ 113.54)

UNI-T

True effective value
Digital table multimeter

Universal measuring device with outstanding features and a multi-function 40,000 count display!

- 4-¾ digit dual display, 40,000 counts
- Accuracy: 0.025 %
- Measures resistance, capacitance, frequency, temperature as well as AC/DC current and voltage
- Diode and continuity check
- Relative value measurement
- Data logger function
- USB/RS232 interface, software included
- Safety: CAT I 1000 V, CAT II 600 V

€ 179,00 (~ £ 150.54)
UT 804

Order now! **www.reichelt.com**

English Hotline: **+49 (0)4422 955-333**

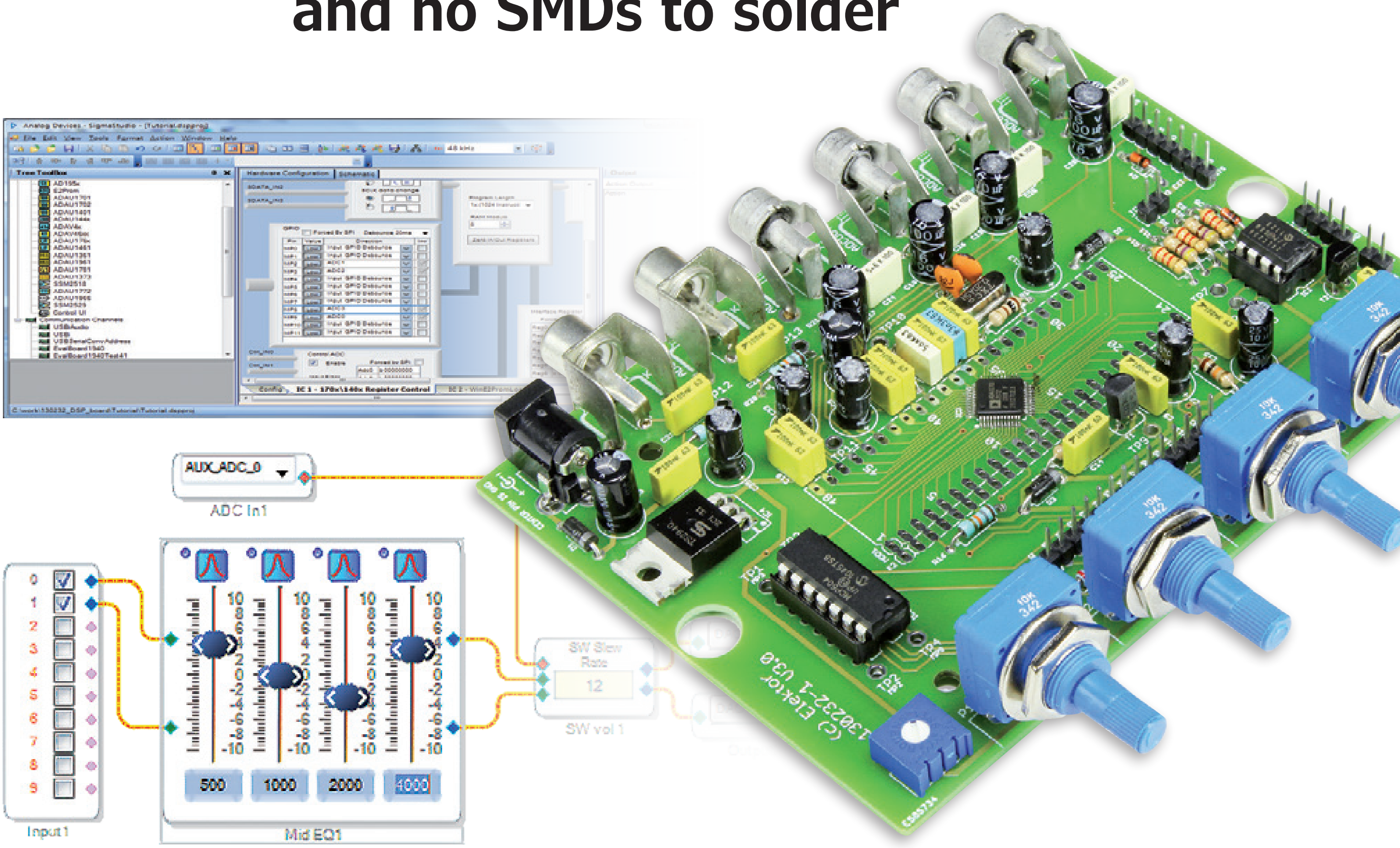
International payment method: PayPal

Daily prices! Prices as of: 12/12/2013

For consumers: The statutory right of withdrawal for consumers shall apply. All stated prices in € include the legal value added tax, ex works Sande/Germany, plus forwarding charges for the entire shopping cart. Our general terms and conditions shall apply exclusively (under www.reichelt.com/agb). Subject to prior sale. All brands, product names and logos are property of the respective manufacturers. Images can be similar. Subject to misprint, errors and changes in prices. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande/Germany (HRA 200654 Oldenburg)

ADAU1701 Universal Audio DSP Board

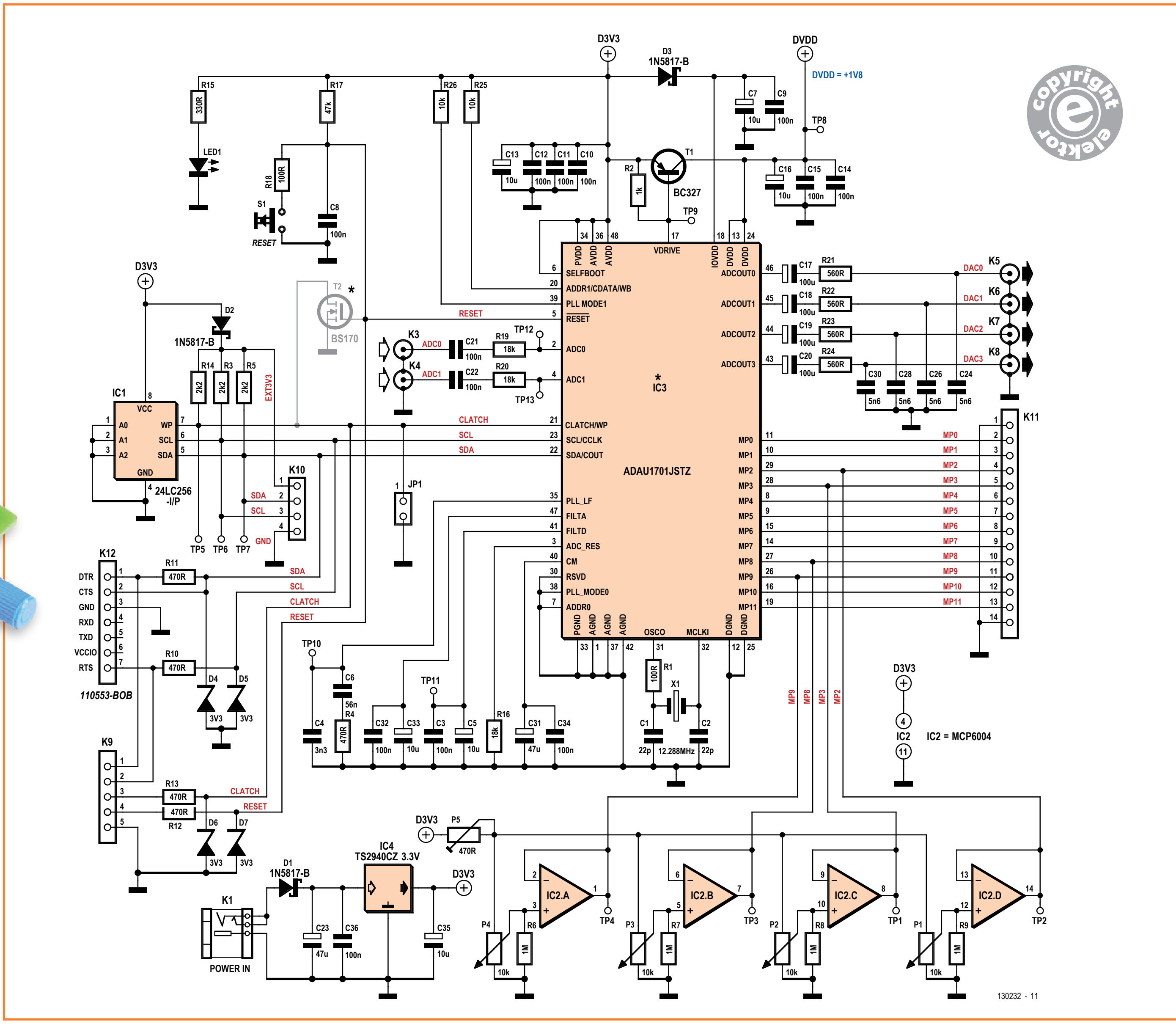
drag 'n drop DSP'ing — and no SMDs to solder

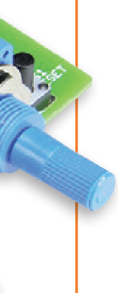


By **Ramkumar Ramaswamy** (India)

It seems that to even dip your feet in the ocean of DSP (digital signal processing) you need lots of SMD soldering skills and a ton of math. *Seems*. Let's end the anxiety: here is a Universal Audio DSP board for the DIYer. Based on the Analog Devices ADAU1701 DSP, this board has only through-hole components—except for the DSP itself, which Elektor PCB Service reflow-solder in place especially 4 U.

Analog Devices' ADAU1701 DSP chip represents a wonderful way of starting out and experimenting with DSP (digital signal processing). Together with the associated (free!) development environment called *SigmaStudio*, the chip is targeted at those who want an easy migration path from all-analog audio signal processing designs and moving into the digital domain. SigmaStudio requires no programming—just drag and drop prebuilt blocks such as "State Variable Filter" and *hey*





presto you have a circuit in minutes which can either be implemented as a DSP platform or simulated using this board and then implemented as an analog platform! So there you have it—an easy-to-assemble DSP board featuring analog and digital I/O, easily programmable with no special hardware, in an environment that allows you to go for a finished product that is DSP-based or analog. All with no math!

Out of the box, the ADAU1701 offers sampling rates of 44.1 kHz and 48 kHz. Higher sampling rates such as 96 kHz are possible but they require deeper configuration.

The board comes with the DSP chip (a 48-legged SMD component!) preassembled. Even better, Elektor's **semi-kit #120232-71** contains that same board and all through-hole parts, so there's also the fun of building it yourself.

DSP board h/w

The schematic shown in **Figure 1** is based on the standard "self-boot" constellation found in the ADAU1701 datasheet, with some additions. All parts except the ADAU1701 DSP chip (IC3) are standard through-hole (TH). Let's do a quick tour. Unstabilized 5 to 12 volts DC enters the circuit via

Figure 1.
Schematic of the ADAU1701 Audio DSP Board. Caution: Transistor T2 must not be fitted.

COMPONENT LIST

Resistors

(all fixed Rs 5%, .25W)
R1,R18 = 100Ω
R15 = 330Ω
R4,R10,R11,R12,R13 = 470Ω
R21,R22,R23,R24 = 560Ω
R2 = 1kΩ
R3,R5,R14 = 2.2kΩ
R25,R26 = 10kΩ
R16,R19,R20 = 18kΩ
R17 = 47kΩ
R6,R7,R8,R9 = 1MΩ
P1,P2,P3,P4 = 10kΩ potentiometer, linear law
P5 = 470Ω preset (trimmer)

Capacitors

C1,C2 = 22pF
C4 = 3.3nF
C24,C26,C28,C30 = 5.6nF
C6 = 56nF
C3,C8,C9,C10,C11,C12,C14,C15,C21,C22,C32,C34 ,C36 = 100nF
C5,C7,C13,C16,C33,C35 = 10µF 16V, radial, 2.5mm pitch
C23,C31 = 47µF 25V, radial, 2.5mm pitch
C17,C18,C19,C20 = 100µF 16V, 2.5mm

Semiconductors

D1,D2,D3 = 1N5817
D4,D5,D6,D7 = BZX79-C3V3 3.3V zener diode
IC1 = 24LC256-I/P, DIL8 case
IC2 = MCP6004, DIL14 case
IC3 = ADAU1701JSTZ (LQFP48 case)
IC4 = TS2940CZ-3.3
T1 = BC327
T2 = DO NOT FIT

Miscellaneous

K3,K4,K5,K6,K7,K8 = RCA/cinch socket, PCB mount
K1 = DC power adaptor connector
S1 = pushbutton w. tactile feedback
X1 = 12.288MHz quartz crystal
JP1 = 2-pin pinheader, 0.1'' pitch
K10 = 4-way pin header, 0.1'' pitch
K9 = 5-way pin header, 0.1'' pitch
K11 = 14-way pin header, 0.1'' pitch
K12 = 7-way pin header, 0.1'' pitch
Jumper
PCB, Elektor Store # 130232-1, comes with ADAU-1701JSTZ presoldered
Semi-kit, Elektor Store # 130232-71, includes PCB 130232-1 and all through hole parts

www.elektorpcbservice.com

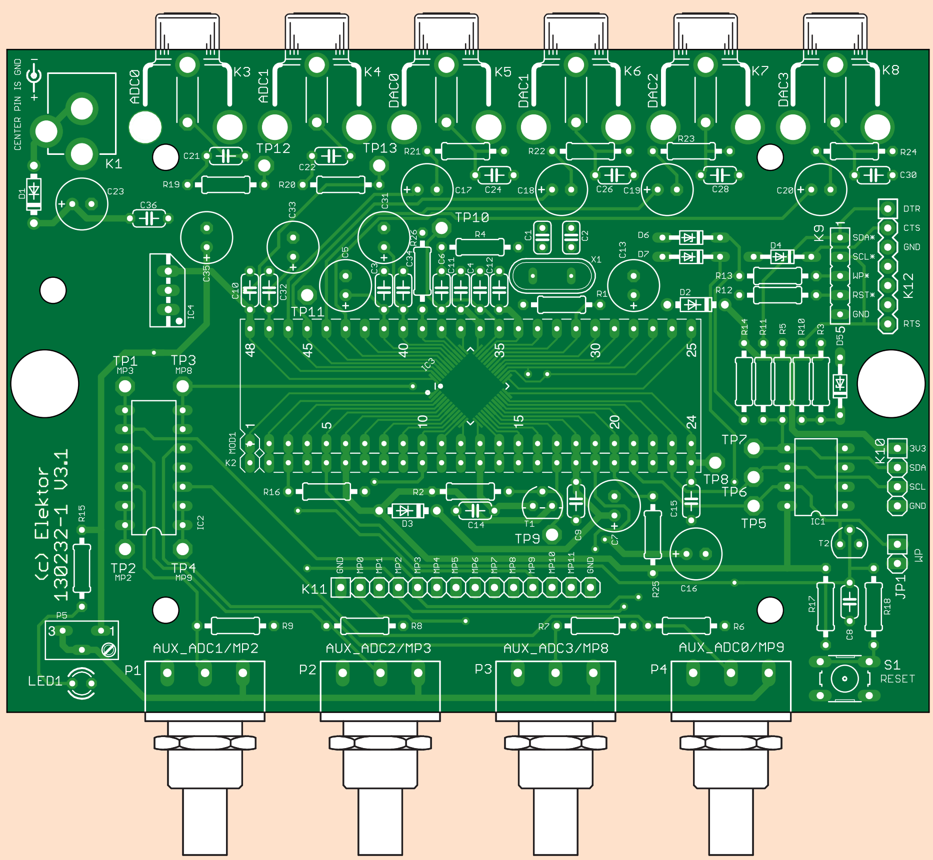


Figure 2.
Circuit board design for the ADAU1701 Universal Audio DSP Board. Conveniently the board comes with the DSP chip soldered in place. Do not fit T2.

socket K1. Schottky diode D1 provides protection against reverse polarity of the input voltage. All other circuitry on the board operates off a 3.3 V voltage supplied by regulator IC4.

The stereo analog input sound is applied to DSP audio inputs ADC0 and ADC1 via K3 and K4. The digitally processed audio from DAC0, DAC1, DAC2 and DAC3 is available on cinch connectors K5–K8.

Pots P1, P2, P3 and P4 are analog inputs for the DSP. They deliver a voltage from 0 V to a maximum level defined by preset P5. The pot wiper voltages are buffered by rail-to-rail opamps IC2.A through IC2.D.

Pushbutton S1 is used to reset the DSP via its $\overline{\text{RESET}}$ pin (5).

Jumper JP1 on the DSP's CLATCH line must be installed to program the on-board AT24CP EEPROM (IC1). Note that the DSP will not boot with JP1 fitted. After programming remove JP1 and press S1 to allow the DSP to boot.

K10 is for programming the EEPROM using a special programmer. K12 allows using the Elektor FTDI-BOB module # 110553 to program the EEPROM using the Elektor application *S.Studio to EEPROM Converter*. This is rather slow though.

K9 is for programming the EEPROM using custom hardware like an Arduino board. Its pins accept 5-V swing and are protected by 3.3 V zener diodes D6 and D7.

K11 finally gives access to all GPIO pins of the DSP. Four lines (2, 3, 8, 9) are also used by the potentiometers P1-P4. Note however that these four lines can only be used to access the outputs of the opamp IC2, and not as inputs.

DSP board assembly

Elektor Labs have designed a compact PCB for the circuit—the component layout is shown in **Figure 2**. This board is supplied with the DSP chip pre-assembled, so the rest of the construction should be easy. The introductory photograph and the one in **Figure 3** show the prototype assembled by Elektor Labs. Note that **transistor T2 must not be fitted**. It was required during an

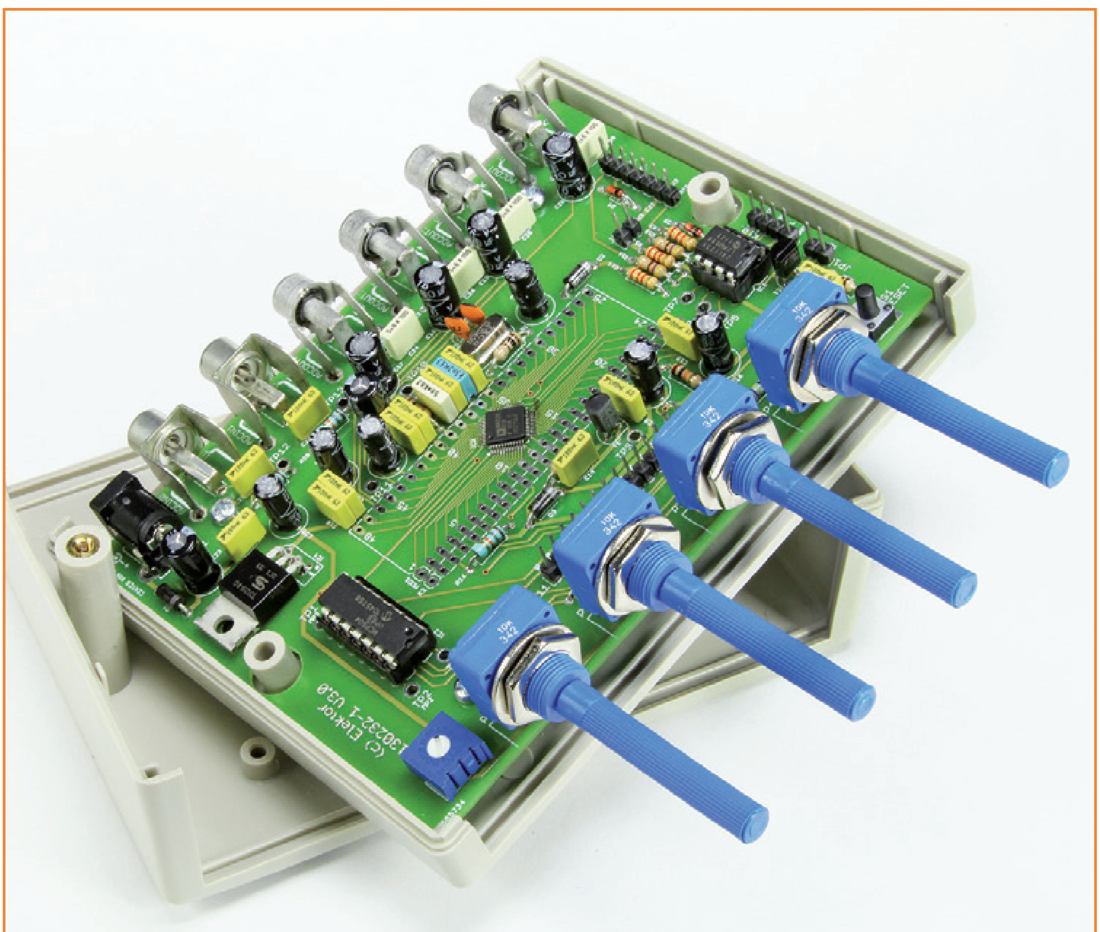


Figure 3.
View of the completed board and enclosure bottom half.

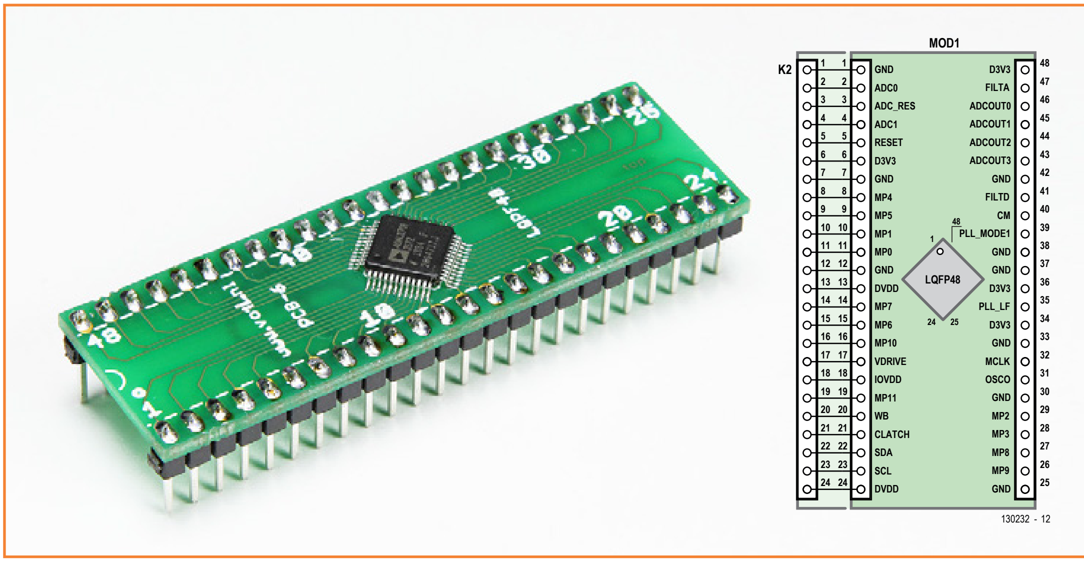
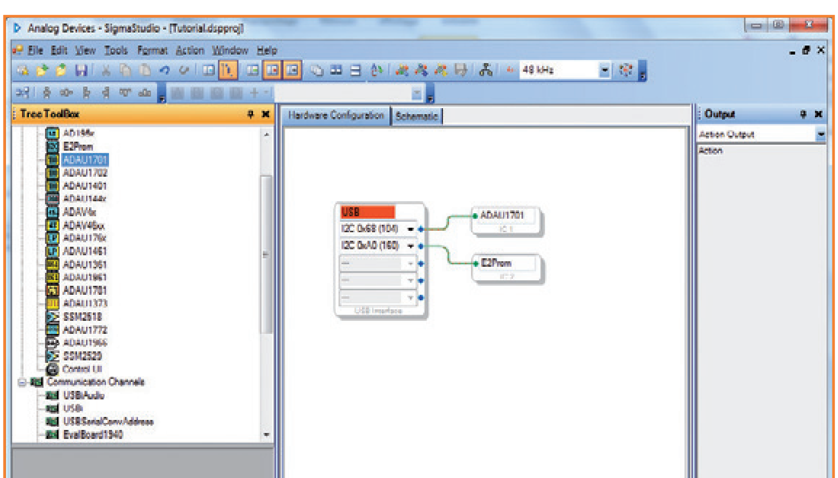


Figure 4.
Example of the ADAU1701 SMD chip fitted on an SMD-to-DIP adapter board—in this case a Voti type PCB-06 (LQFP48). For convenience, the board pinout and ADAU1701 DSP pin index is pictured also.

early phase of the project but causes malfunction in the final setup.

Those of you keen to have the ADAU1701 DSP chip *only* on a separate board for dropping into their own project (stomp box, audio processor, ham radio TRX) will be delighted to know that LQFP48-to-DIP adapter boards are available from various sources like Voti [1], Dipmicro [2], Proto Advantage [3], Schmartboard [4] and … EBay! The Schmartboard is our favorite for solderability, kind staff and the pinheaders included. **Figure 4** shows the Voti adapter board assembled by Labs. Our DSP board accepts both 0.6 inch and 0.7 inch wide adapter boards—either can go in PCB position MOD1, but only if the DSP chip is not fitted on the DSP main board. This provision should suit those of you having persisted in etching and drilling their own circuit board from the artwork provided [5].

Figure 5.
Creating a project in SigmaStudio.



DSP'ing the graphic way

DSP sadly but falsely remains associated with mathematics which tends to put people off. Happily, the Analog Devices' *SigmaStudio* environment follows an almost entirely graphical approach to configuring that complex DSP chip. With SigmaStudio it's drag and drop mostly instead of mathematics. Let's see how it's done in a Tutorial.

The prerequisites are:

- Windows 7 x86/x64, Vista, XP Professional or Home Edition with SP2;
- at least one COM port or a USB-to-serial adapter;
- .net framework 4 + 3.5;
- SigmaStudio 3.9 (download at [7], free but account needed);
- Elektor SigmaStudio to EEPROM Converter utility [5], an Arduino Uno or Mega board, or a I²C EEPROM programmer supporting the AT24CP.

Step 1. Create a SigmaStudio project

Start SigmaStudio. Select File → New Project and wait for the project to be created. You will see elements in the toolbox on the left; drag and drop "ADAU1701","USBi" and "E2Prom" blocks to the main window named Hardware Configuration. Now link one of the blue dots of the block

USBi to the green dot of the ADAU1701, link the other blue dot of USBi to the green dot of the E2Prom. The result should look like in **Figure 5**. Save the project (e.g. as “Tutorial”), in a folder of the same name.

Now Click on the “Schematic” tab. Notice the change of toolbox at the left.

- Under IO → Input, drag and drop an “Input” element.
- Under IO → Output, drag and drop an “Output” element twice.
- Under GPIO → Input, drag and drop the “Auxiliary ADC Input” element.
- Under Volume Controls → Adjustable Gain → Ext Control → Clickless SW Slew, drag and drop the “Single slew ext vol”.
- Right click on the “SW vol 1” block, select Grow Algorithm → 1. Ext vol (SW slew) → 1. Block “SW vol 1” should now have 3 inputs and 2 outputs.
- From Filters → Second order → Single precision → 2 Ch drag and drop “Medium Size Eq.”
- Right click on the “Mid EQ 1” block, select Grow Algorithm → 1. 2 Channel – Single Precision → 3. Block “Mid EQ 1” should now have four sliders.
- Connect the blocks as pictured in **Figure 6**.

Now we need to configure GPIO9 as an auxiliary ADC input.

- Return to the “Hardware Configuration” tab and then click “IC1 -170x\140x Register Control” tab at the bottom of the window.
- Select MP9 In the GPIO block and change “Input GPIO Debounce” to “ADC0”, then return to “Config” tab as illustrated in **Figure 7**.

Return to the Schematic and change the values of the slider frequencies in the equalizer and set the gain to your taste. The project is ready, save it. Press F7 or select from the menu Action → Link Compile Download. You will see a Comms error because *duh* you do not have the official programmer. At some point during the creation of the schematic you will see a message mentioning USB problems. You can safely ignore this message. Return to the Hardware Configuration Tab, right click on IC1 and select “Write Latest Compilation to E2PROM”. This will create an executable file that we can program into the EEPROM.

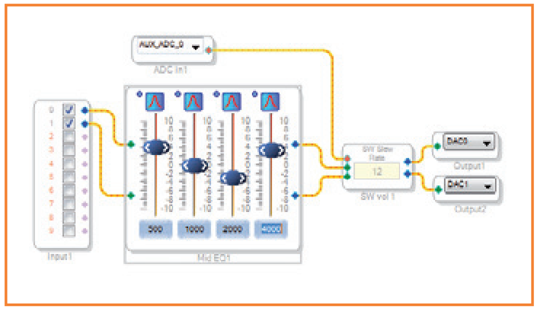


Figure 6.
These four sliders are the virtual equivalents of the pots on the DSP board. Here we’re creating an equalizer function. Look, no maths!

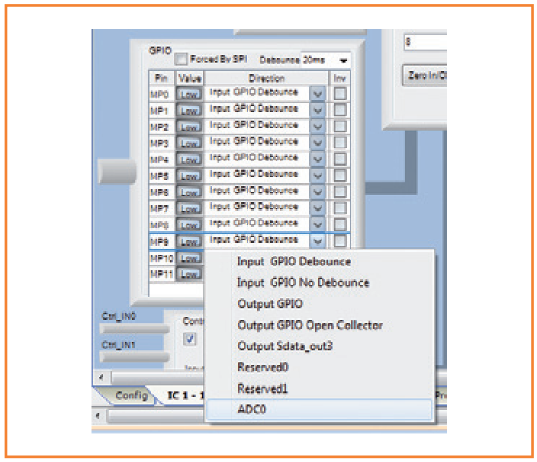


Figure 7.
Let’s do some GPIO configuring in SigmaStudio.

Step 2. Programming the EEPROM
Three possibilities exist—each is discussed below.

a) Programming over the serial port using the Elektor Sigma Studio Serial I²C EEPROM Programmer utility (it’s rather slow). See **Figure 8**.

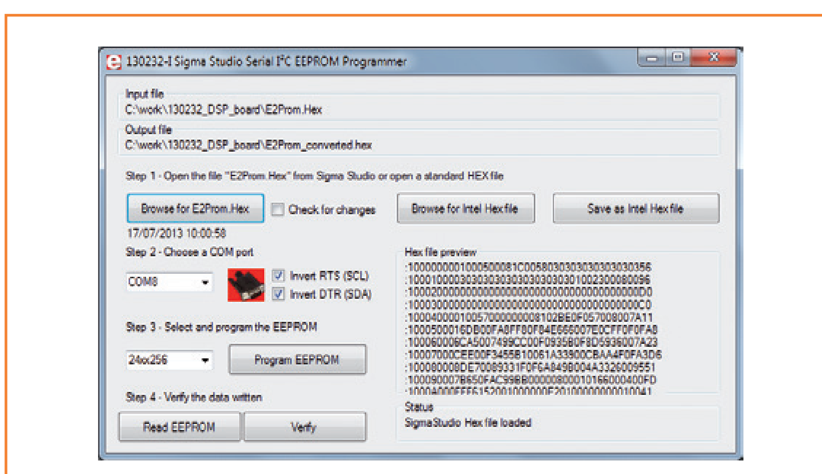


Figure 8.
Elektor’s ‘Sigma Studio Serial I²C EEPROM Programmer’ utility in action.

Creating a Burnable EEPROM File

One of the first things you will need to get hold of is the somewhat hard-to-find method to capture the EEPROM text file generated by SigmaStudio and convert that to something you can burn. I have distilled the procedure from various posts on the SigmaDSP forums and it is presented here for ready reference.

Step 1. Open the project file (.dspprog) in SigmaStudio, and navigate to the schematic page.
Step 2. Click Hardware Configuration tab. Press F7 or menu item Action → Link Compile Download.
(If the Analog Devices Eval Board is not connected, you will get a message saying "Communication Failure" that may be respectfully ignored.)
Step 3. Right click on the ADAU1701 box and choose "Write latest compilation to E2PROM." In the project folder (which contains the .dspproj file), look under the subfolder IC2 for the file E2Prom.hex.
Step 4. Open E2Prom.hex in a text editor. Use search and replace to remove all "," and "0x" and save the file.
Step 5. Open the HxD hex editor. Click New, copy all of the contents of the modified E2PRom.hex file and paste it into HxD.
Step 6. Use File-Export to export the file in one of the Intel Hex or Motorola S-Record formats listed. Burn the exported file into the EEPROM using your favorite EEPROM burner.

- Connect a stereo audio source to ADC0 and ADC1, and headphones or an amplifier to DAC0 and DAC1.
- Connect the DTR, RTS & GND lines of a serial port to K12 using either a real RS-232 port or a USB-to-serial converter like the Elektor BOB. Note that depending on the type of port you use you may need to invert or not the RTS and DTR signals. Checkboxes to do this are provided (default values are for Elektor BOB).
- Place a jumper on JP1
- Power the DSP board
- Launch the utility and open the file E2Prom.Hex that you created with SigmaStudio. A preview in Intel Hex format will be shown. You can also load a file in Intel Hex format if you have such a file
- Select the serial port in the utility
- Select the right EEPROM size and click the "Program EEPROM" button. Programming will start and may take several minutes depending on the size of the file. The slowness is due to the serial port driver.
- To verify if all went well first click "Read EEPROM" (this will again take a while), then click "Verify".
- If all went well, remove the jumper from JP1 and press the Reset pushbutton on the DSP board. Verify that you can adjust output volume by turning P4.

Note 1: This tool allows the conversion of an E2Prom.Hex file in proprietary Analog Devices format into a standard Intel Hex file that can be used with a commercial EEPROM programmer. Use the button "Save as Intel Hex File" to do this.
Note 2: This tool can also be used to read an EEPROM and save its contents as an Intel Hex file. To do so start the tool without loading a Hex file. Then click "Read EEPROM" (this will again take a while), followed by a click on the button "Save as Intel Hex File". By the way, it will read no more than 10,000 bytes as the maximum possible ADAU1701 program size is about 9,200 bytes.

b) Use an Arduino board as a programmer (recommended). To speed up programming we wrote an Arduino Sketch that accepts a file in E2Prom format and that will program it into the EEPROM. To make things even better we added automatic control of jumper JP1 and the Reset pushbutton. Connector K9 allows you to connect the Arduino board. The default pinout is:

Signal	Arduino pin
SDA	10
SCL	11
WP	12
Reset	13
GND	GND

The Sketch is of course available for download [5] (Dead Parrot not included).
Use a serial port terminal program to send the E2Prom.Hex file to the Arduino board (option "Send file" in Tera Term). The baud rate is 19,200 baud. Faster rates will not work well.

c) Use a dedicated EEPROM programmer. Refer to the manual of your programmer. You can use the Elektor utility mentioned under a) to create useable files. Connector K10 is available for direct connection to the EEPROM's I²C pins.

Step 3: try the examples

A few examples like a stereo chorus, short delay and a state variable filter with three outputs have been created and are contained in the free software archive [5].

That concludes the step-by-step programming. All EEPROM programming can be done with the DSP board powered.
K12 is for the Elektor USB-serial BOB. This 7-pin connector employs all seven connections at the front side of the BOB where normally a 5-pin connector is mounted. Signals RTS and DTS required here are located in-line just beside the 5 connections, allowing a 7-pin connector to be fitted easily.

The SigmaDSP Forum

The art of writing exhaustive manuals and Help files has deteriorated in the Internet Age—today a product is pushed out well before its documentation is complete. SigmaStudio is no exception, and there is lots of essential programming information that is unfortunately not in the Help files. To make full use of the platform you need to refer to the Analog Devices EngineerZone SigmaDSP forum [7] and look at the *I see's* and *aha's* that people have posted there. Needless to say there is a high probability you will end up registering and posting questions (and answers) there. For issues specifically related to this publication you're also more than welcome at the Elektor Forum.

(130232)

Internet References

[1] Voti 48-pin LQFP adapter board: www.voti.nl/winkel/catalog.html

[2] Dipmicro 48-pin LQFP adapter board: www.dipmicro.com/store/PCB-LQFP48-DIP48B

[3] Proto Advantage 48-pin LQFP adapter board: www.proto-advantage.com/store/product_info.php?products_id=2200109

[4] Schmartboard 48-pin LQFP adapter board: www.schmartboard.com/index.asp?page=products_smttodip&id=451

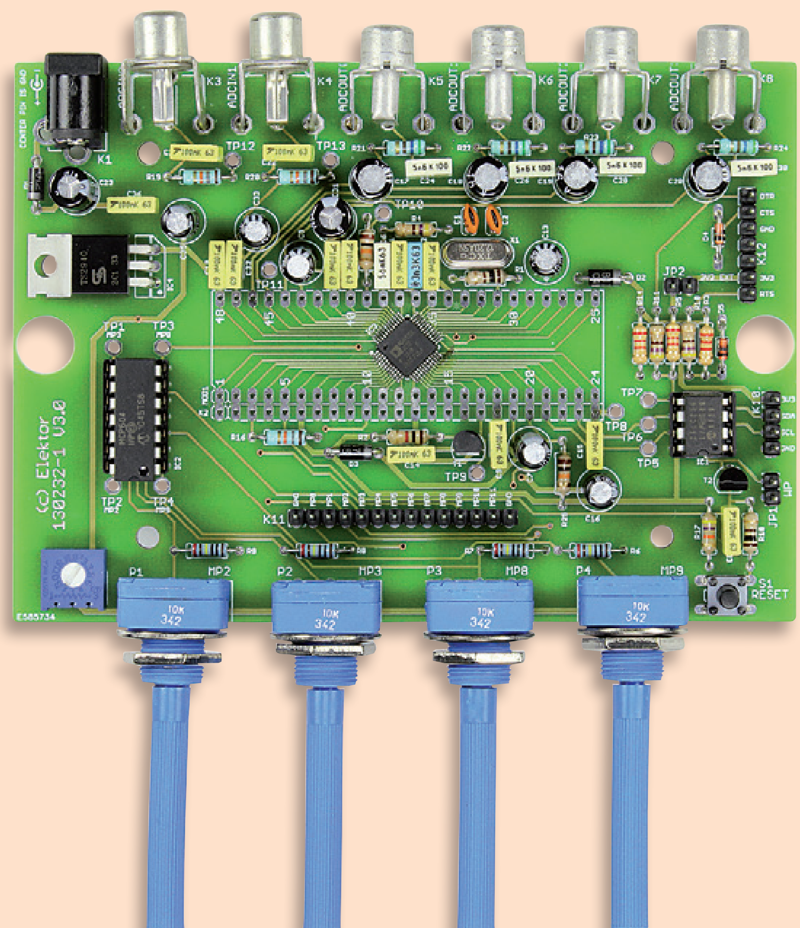
[5] www.elektor-magazine.com/130232

[6] Sigma Studio: www.analog.com/en/dsp-software/ss_sigst_02/sw.html

[7] SigmaDSP Forum: ez.analog.com/community/dsp/sigmadsp

Features

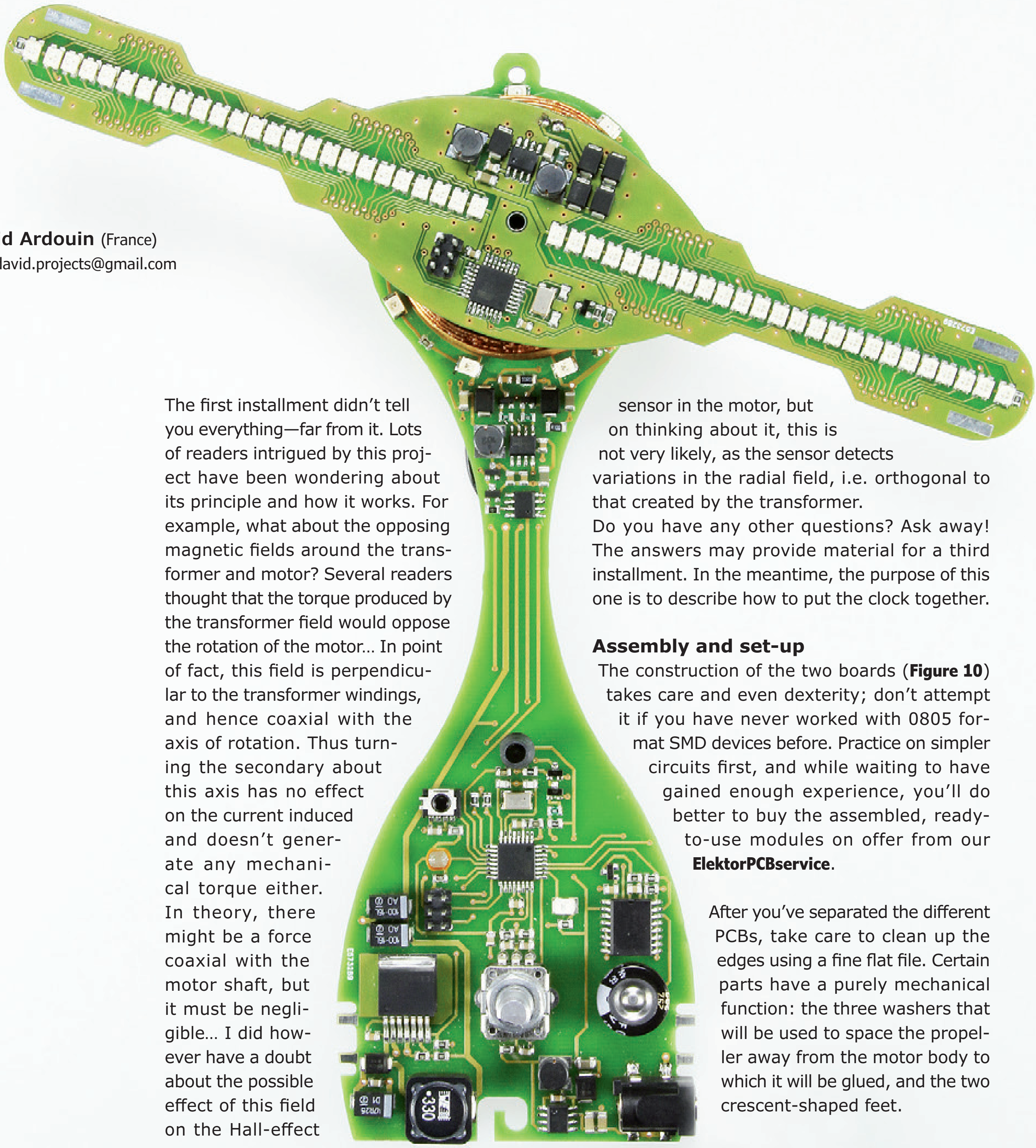
- Analog Devices ADAU1701 DSP
- Sigma Studio 3.9 graphic oriented DSP support software (free)
- Examples for stereo chorus; short delay; state-variable filter
- Design your own sound effects
- Stereo audio input
- 4 DAC outputs
- 4 pots feeding into aux ADC inputs
- 8 GPIO pins
- I^{2}C EEPROM for effects storage
- DSP chip preassembled on PCB
- Optional: DSP chip on 48-pin DIP carrier board (0.6 inch or 0.7 inch)
- Semi-kit available from Elektor (SMD presoldered PCB and TH parts)



UltiProp Clock (2)

Part 2

By **David Ardouin** (France)
ardouin.david.projects@gmail.com



The first installment didn't tell you everything—far from it. Lots of readers intrigued by this project have been wondering about its principle and how it works. For example, what about the opposing magnetic fields around the transformer and motor? Several readers thought that the torque produced by the transformer field would oppose the rotation of the motor... In point of fact, this field is perpendicular to the transformer windings, and hence coaxial with the axis of rotation. Thus turning the secondary about this axis has no effect on the current induced and doesn't generate any mechanical torque either. In theory, there might be a force coaxial with the motor shaft, but it must be negligible... I did however have a doubt about the possible effect of this field on the Hall-effect sensor in the motor, but on thinking about it, this is not very likely, as the sensor detects variations in the radial field, i.e. orthogonal to that created by the transformer.

Do you have any other questions? Ask away! The answers may provide material for a third installment. In the meantime, the purpose of this one is to describe how to put the clock together.

Assembly and set-up

The construction of the two boards (**Figure 10**) takes care and even dexterity; don't attempt it if you have never worked with 0805 format SMD devices before. Practice on simpler circuits first, and while waiting to have gained enough experience, you'll do better to buy the assembled, ready-to-use modules on offer from our **ElektorPCBservice**.

After you've separated the different PCBs, take care to clean up the edges using a fine flat file. Certain parts have a purely mechanical function: the three washers that will be used to space the propeller away from the motor body to which it will be glued, and the two crescent-shaped feet.

Note: the numbering of the illustration continues from the first article

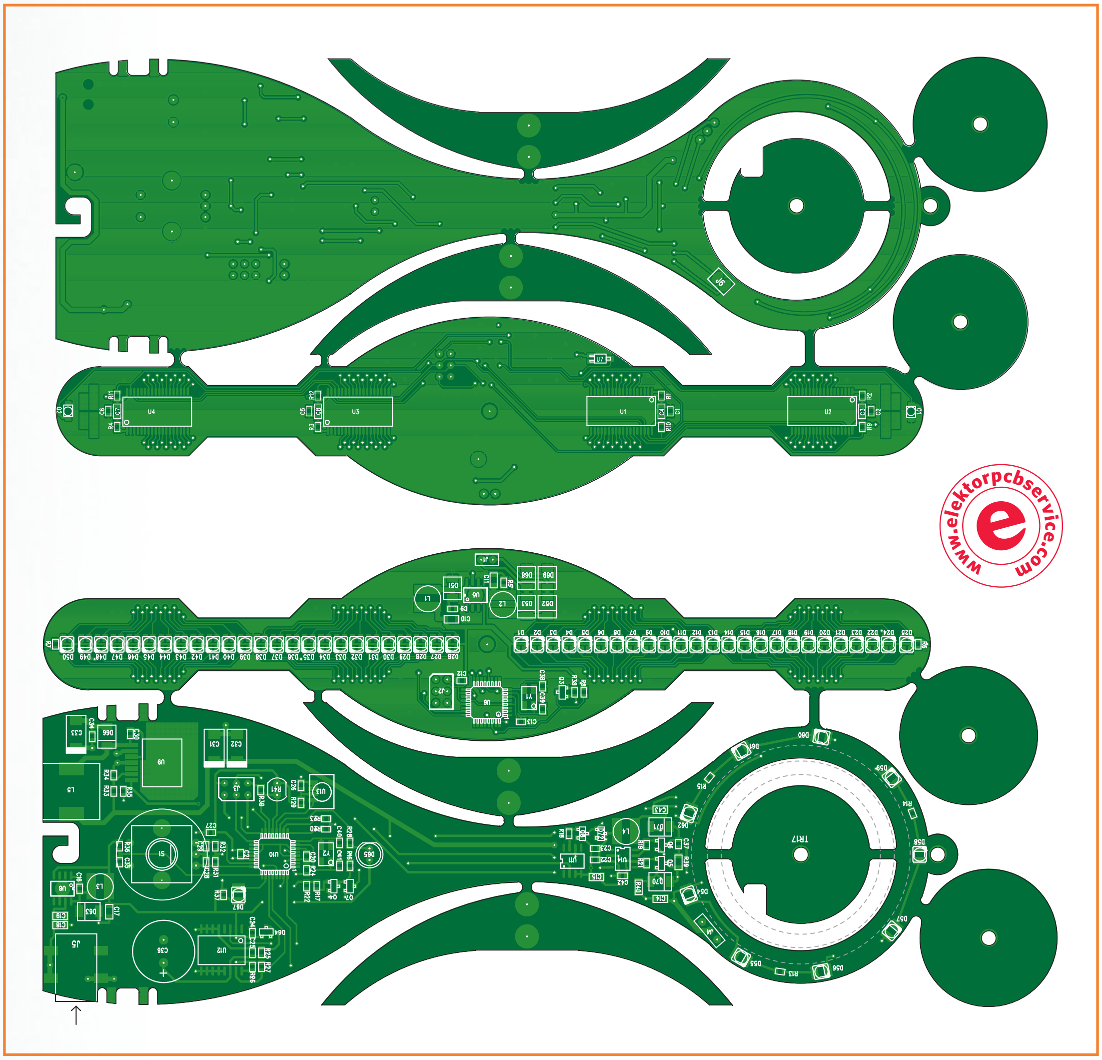


Figure 10a.
Clock PCB. In theory it's not necessary, but the rectangular lands at the ends of the propeller can be loaded with solder if necessary to correct any imbalance problems.

For soldering the fine-pitched integrated circuits, I first lay down a bed of solder short-circuiting across all the pins using a large-diameter bit. Then, using desolder braid, I remove the excess solder to eliminate all the bridges between the pins, leaving only the strict necessary to make the contact between the pins and the copper. ICs U5 and U8 have a solder pad on their underside. Solder their eight pins as normal, then solder the bottom pad from underneath by way of the hole in the PCB. If you are paying attention, you'll notice that on the PDF of the PCB this hole seems to be only 0.5 mm. Don't worry, on the circuit supplied by ElektorPCBservice, its diameter is

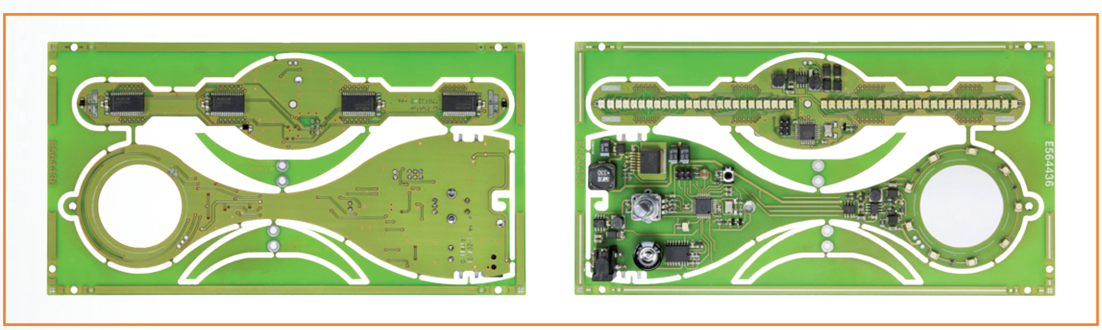


Figure 10b.
This version of the board didn't yet have the 3 spacer disks (see Figures 10a & 13a).

Figure 11a.
Setting the fuses for the Base Unit ATmega328.

Fuses	Extended	0xFC	Brown-Out activated @ 4.3 V
	High	0xD9	External 20 MHz oscillator, maximum start-up time
	Low	0xFF	

Figure 11b.
Setting the fuses for the Propeller ATmega328.

Fuses	Extended	0xFD	Brown-Out activated @ 2.7 V
	High	0xD9	External 20 MHz oscillator, maximum start-up time
	Low	0xFF	

indeed 3 mm, which will allow you to solder the component from below. Don't be afraid to fill the hole with solder to improve the thermal transfer. On the propeller, the alignment of the 50 diodes and the symmetry of the two blades must be perfect. When spinning, the tiniest error will be visible in the image produced. If you have read the first article properly, you'll also understand the importance of perfect symmetry between the rows of LEDs on the two halves of the propeller. You'll notice on the photos of the prototype that diode D65 is fitted with a sleeve in black heat-shrink sleeving, in order to obtain as narrow a light beam as possible, in order to ensure a clear-cut signal as the propeller passes.

The microcontrollers are programmed via an ISP link using the software supplied [2]. Watch out for the fuse settings (**Figures 11a & b**), which are different between the two devices: on the propeller, the brown-out detection is configured to 2.7 V to guard against voltage dips when there is a high current demand, which is inevitable with the way this part is powered.

Self-diagnosis

Before (!) moving on to the mechanical assembly and even before winding the transformer, you can check that the two boards work properly on your workbench, first independently and then making them communicate (even without the propeller spinning). First off, on the base, keep S1 pressed while powering up. The test diode D67 will then flash at 2 Hz, indicating that test sequences are being sent over the infrared data link: look closely, and you should see a reddish glow in D65. At the same time, the motor (which you will have hooked up temporarily) is

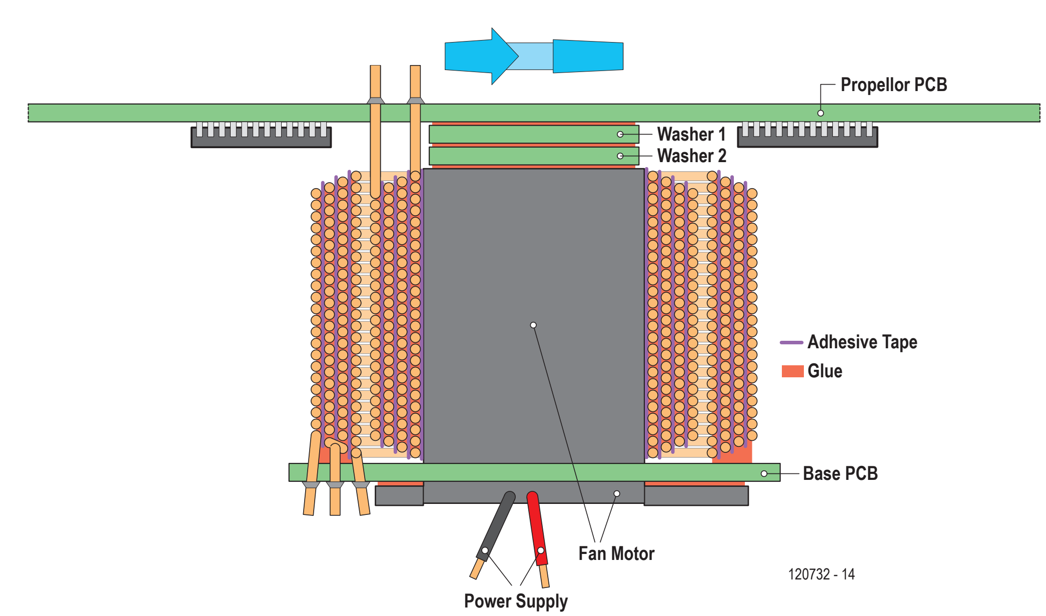


Figure 12.
Cross-sectional view of the assembly.

activated and should run at full speed; the voltage for the transformer is also present. Using an oscilloscope and temporary 1–10 kΩ resistors connected between pins 1-2 and 2-3 on J4, you will be able to observe the 50 kHz square-wave drive signal with an amplitude of 9 V.
Let's now test the propeller independently of its base. Power it via a 9 V battery on J1 (the polarity doesn't matter). The two LEDs closest to the center ought to light at once. If you now pass Q1 or Q2 in front of the infrared diode D65 on the base unit, a third LED lights indicating correct reception of the positioning signal. If you obtain this result, your propeller is ready for use.
Bring the central part of the propeller closer to the base, directing U7 towards the ring formed by D54–D62. The propeller circuit confirms the reception of the data sequences by lighting in turn the fifty diodes on the two blades. You can't find U7? Look under the propeller…
Were these tests satisfactory? Well let's move on to the assembly then!

Mechanical

The biggest step in the mechanical construction consists in modifying an 80 mm fan, common in computers, keeping only the motor itself, and winding the transformer onto it. To avoid having to go into the details of this operation here, I've prepared a document [1] that I invite you to read: it explains everything step by step in detail with pictures. Follow these instructions to the letter and you will obtain without any trouble (but with some patience) the vital transformer. The cross-sectional view of the top (**Figure 12**) and the photo (**Figure 13**) give an idea of the assembly of this spinning team. The primary, wound on the outside, is fixed to the base and so doesn't move. The motor stator is also glued to the base.

The secondary, with a slightly smaller diameter, is positioned in the center of the primary and

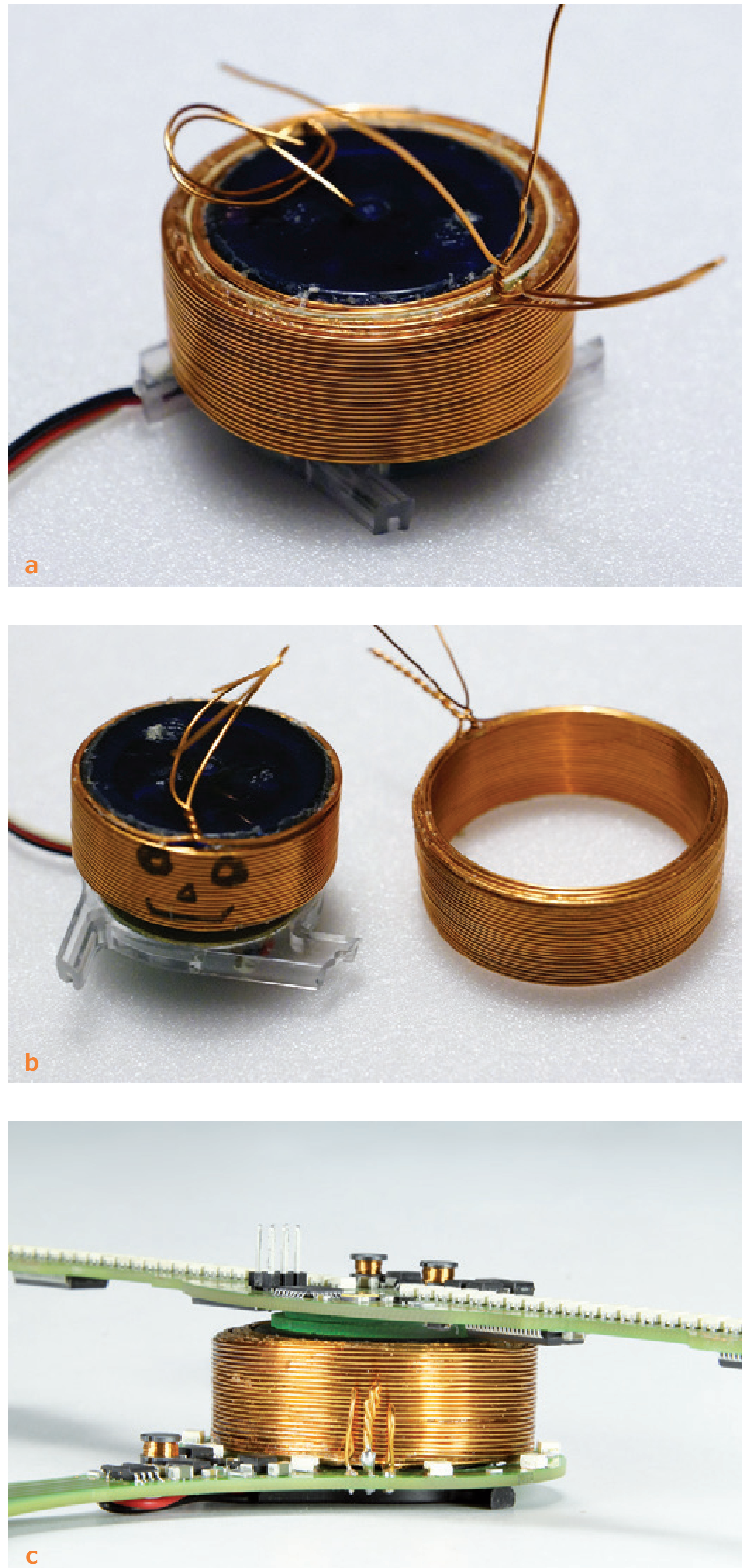


Figure 13.
Close ups of the transformer after winding (a), then separated (b) and lastly assembled (c). In (a) we can make out the strip of card which was used during winding to maintain a space between the primary and secondary windings, as well as in (c) the two green disk spacers that space the propeller away from the motor to which it is glued.

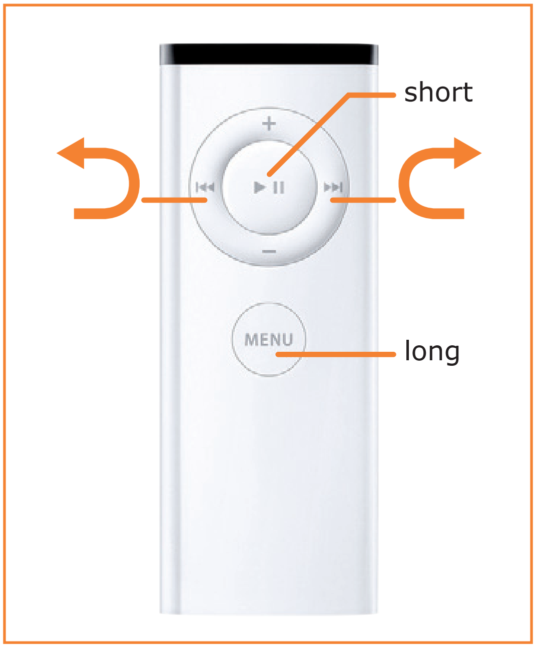


Figure 14.
Using the
Apple remote-control.

glued around the fan motor. So it turns with the propeller, which is itself glued to the motor hub. To keep the turns in place while winding the transformer, you'll need to insert between the layers of copper wire a good (!) thin, double-sided adhesive tape, then add instant glue as you go along to keep the turns finally in place.

The two crescent-shaped pieces are the feet, to be fitted by soldering. Fitted perpendicular to the base unit PCB, they will support your clock if it is placed on a flat surface. Don't solder them till everything else is finished.

If you prefer, the hole at the top of the base unit will let you hang your clock on a small nail on the wall or any other appropriate support. You can thread the power cable through the hook-shaped notch made for this purpose at the foot of the base unit. This cable mustn't get tangled up with the propeller! We also draw your attention to the mechanical fragility of certain components, in particular at the ends of the blades (Q1 and Q2); when the propeller turns at full speed, you must avoid these components hitting an obstacle.

The clock's not slow

There's ventilation in the air! If you have checked the operation of your two boards, in a few seconds and in a gentle waft of air you're going to discover the magic of your UltiProp Clock. All that is missing is a power supply between 10–14 V DC, capable of supplying at least 250 mA at cruising speed. When power is applied, the variables and peripherals are initialized, then the motor and transformer start up. On the propeller side, D1 and D26 light constantly to confirm the presence of the supply. A few moments later, LED D65 is lit in turn, to bring the display to life. The time is then shown in the top half, and a welcome message scrolls.

Once started, the clock can be controlled by either the remote-control (**Figure 14**) [3] or the rotary encoder. Press this button to put the system into stand-by, or to wake it up. Turn it to the right or left to scroll at will through several tens of display modes, with various combinations of colors and representation of the clock-face and the hands,

Component List

Resistors SMD 0805, 5%
R1–R5,R23,R30 = 47kΩ
R6,R7,R16,R22,R26,R27,R31,R32,R36,R37 = 4.7kΩ
R8–R12,R24 = 22kΩ
R13,R14,R15,R17,R19,R21,R29 = 100Ω
R25,R34 = 1.5kΩ
R28 = 330Ω
R33,R35 = 10kΩ
R38 = 1MΩ

SMD 1206, 5%
R39 = 10Ω
R40 = 0.1Ω

Capacitors Ceramic, SMD 0805, 20%
C1,C2,C5,C6,C12,C13,C20,C21,C24,C25,C27 ,C34 = 100nF 50V
C9,C16,C30 = 10nF 50V
C22,C23,C26,C28,C29,C35 = 1µF 10V
C37 = 1nF 50V
C38–C41 = 33pF 10V

Ceramic, SMD 1206, 20%
C3,C4,C7,C8,C10,C17 = 22µF 10V
C11,C18,C19,C43 = 1µF 50V
C14,C15 = 10µF 16V

Miscellaneous
C31,C32 = 100µF 16V, Vishay 593D
C33 = 47µF 25V, Vishay 593D
C36 = 0.1F 5.5V, Panasonic EECF5R5U104 (9692703)

Inductors
L1,L3 = 68µH 0.84A, Bourns SDR0604-680KL (1828011)
L2 = 220µH 0,38 A, Bourns SDR0604-221KL (1828016)
L4 = 1mH 0.12A, Bourns SDR0604-102KL (1828020)
L5 = 33µH 3A, Würth 744771133 (2082608)

Semiconductors
D1–D50 = LED, bicolor, PLCC4, choose your color, e.g. Vishay VLMKE3400-GS08 (1328370)
D51,D52,D53,D63,D66,D68,D69 = MBRS140T3G, ON Semiconductor (9557237)
D54–D62 = SFH421-Z, infrared LED, Osram (1226346)

with or without the seconds displayed, as well as the date and ambient temperature, and the choice between analog or digital display. Once you have made your choice, don't touch anything for ten seconds and this mode will be saved into non-volatile memory so it can be restored next time the unit is turned on. Press the encoder for more than two seconds to enter the configuration menu and set the time, date, choice of language, rotational speed in day and night modes, and the brightness, etc. You can navigate through this menu by short presses to select the required parameter and then by rotating to adjust the value.

Choosing zero speed in night mode allows the unit to go into stand-by and wake up again automatically according to the ambient light in the room, ideal if the project is fitted in a bedroom. This option is handy, but it can catch you out. If by chance your clock refuses to start up, this may be because there's not enough light and it thinks you are sleeping. So before you start looking for a fault, do make sure the surroundings are bright enough. Next time you adjust the parameters, remember to raise the stand-by threshold a little.

This is where our electronic clock adventure together ends. Over to you now to get stuck in. I hope this project will give you as much pleasure as I have derived from developing it, and that lots of you will come and share your experiences and enthusiasm in the Microcontrollers & Embedded area on the Elektor forum.

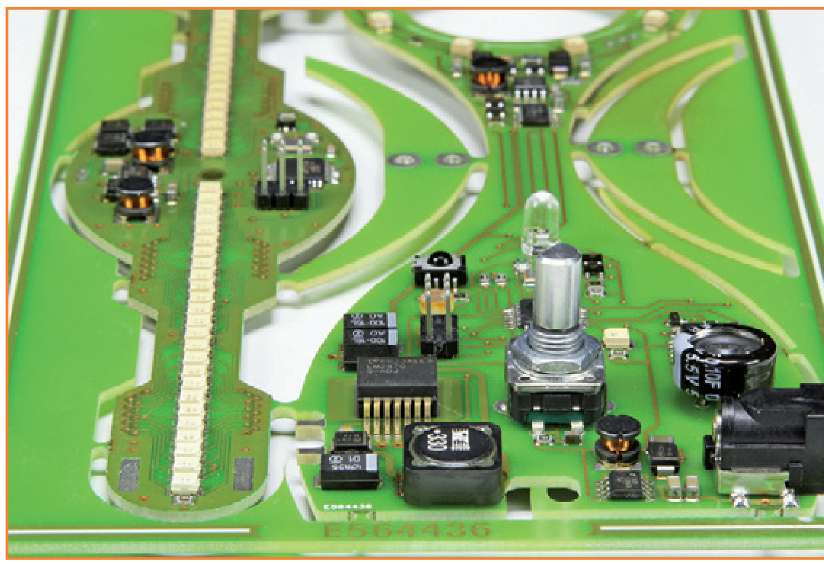
(130389)

Internet Links

[1] 120732 – UltiProp Clock Assembly.pdf

[2] Downloadable software, including source code: www.elektor-magazine.com/120732

[3] Remote control for Apple Universal Dock: http://store.apple.com/en/product/MC746ZA/A/apple-universal-dock



D64 = BAT54
D65 = VSLY5850, infrared LED, Vishay (1870807)
D67 = LED, orange, PLCC2
D70, D71 = SMBJ48A (1899472)
Q1, Q2 = TEMT1020, phototransistor, Vishay (1470165)
Q3, Q4, Q7 = BC847B
Q5, Q6, Q8 = IRLML0060, transistor, International Rectifier (1783927)
U1 à U4 = MAX6957AAX, Maxim [Digikey # MAX6957AAX+T-ND]
U5, U8 = LM22674M-5.0 (1679666)
U6, U10 = Atmega328P-AU (1715486)
U7 = SFH2400FA-Z, photodiode, Osram (1226452)
U9 = LM2670SD-ADJ (1286849)
U12 = DS3231S, Maxim (1593292)
U13 = TSOP6238, 38kHz infrared receiver, Vishay (4913220)
U14, U11 = TC4427EOA, Microchip (9762647)

Miscellaneous

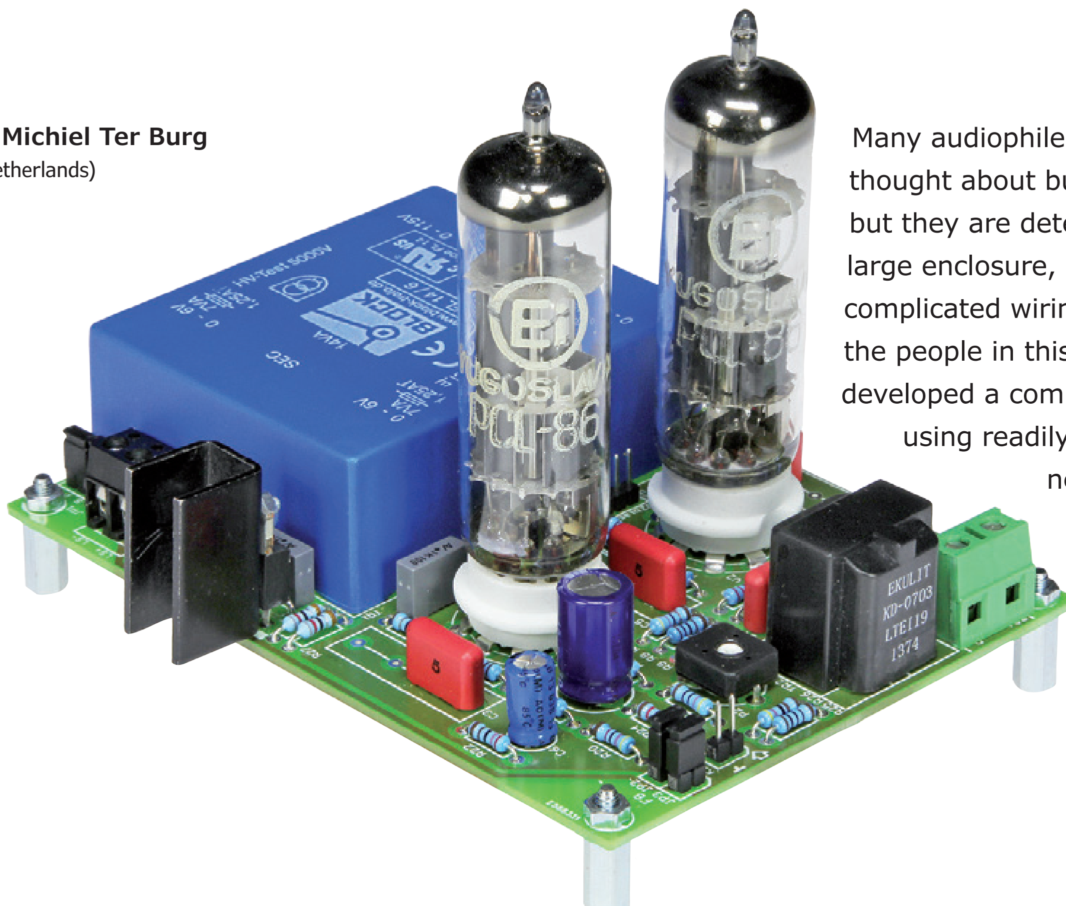
Y1,Y2 = 20MHz quartz crystal, TXC 7B-20.000MAAJ-T (1841988)
S1 = PEC11-4215F-S0024 rotary encoder, Bourns (1653380)
R41 = VT935G, photoresistor, Excelias Tech. (1652638)
J2, J3 = 6-pin pinheader (2x3), 0.1'' pitch
J4 = 3-pin pinheader, 0.1'' pitch
J5 = mini-jack socket, 2.1mm, Cliff DC10AS (1889309)
J6 = ventilator, 80mm, 2000-3000 rpm
TR1 = transformer with two concentric windings [1]
Enameled wire, 0.56mm diam. (23 AWG), approx. 25m (75 ft.)
Instant glue and double sided adhesive tape (see construction manual [1])
PCB nos. 120732-1 through -7

Numbers only in parentheses () are Farnell/Newark/element14 order codes

Compact Tube Amplifier

Using ordinary power transformers

By **Michiel Ter Burg**
(Netherlands)



Many audiophiles have occasionally thought about building a tube amplifier, but they are deterred by the high cost, large enclosure, hefty transformers and complicated wiring harnesses. For all the people in this group, the author has developed a compact, low-cost option using readily available components, so that everyone can enjoy the warm sound of a tube amplifier in their living room or den.

To obtain a wide hi-fi bandwidth and an output power of 10 watts or more, you usually need big transformers for the power supply and driving the loudspeakers. However, in practice it doesn't take a lot of power to achieve a good sound level in an average living room with a pair of reasonably efficient loudspeakers. With regard to the frequency range, most music has virtually nothing the low bass range (below 50 Hz) or the high treble range (above 12 kHz), so you can achieve savings there as well.

This means that if you are willing to relax your requirements a bit, it is possible to build a small tube amplifier with a few watts of output power per channel using ordinary PCB-mount power transformers as the output transformers. The other components are also readily available, for example from mail order companies, and not expensive. The tube used here is a PCL86 (nearest US equivalent: 14GW8), a reasonably modern audio tube (dating from 1961) that combines a preamp triode rated at 0.5 W with a power pentode rated at 9 W. The design uses two of these tubes, with the triodes wired as a phase splitter and the pentodes operating in a conventional push-pull configuration.

The lower limit of the bandwidth is really determined by the primary inductances of the transformers, while the upper limit is determined by the coupling factor of the output transformer. The coupling factor *K* is very low with this sort of transformer because the primary and secondary windings are in separate sections to meet an insulation voltage spec of 5 kV.

The input transformer used here (originally intended for a light organ) results in a fairly low input impedance of approximately 1 kΩ, which is necessary to obtain a reasonable open-loop bandwidth. This should not cause a problem if the input signal comes from a good preamp or the headphone output of a computer, MP3 player or similar source.

Schematic

Figure 1 shows the schematic diagram of the amplifier. The audio input signal on connector K1 is fed to input transformer TR2. The secondary of this transformer drives the two triode sections of the tubes in opposite phase via resistors R8 and R12. The anodes of the pentode sections of tubes V1 and V2 are connected to the primary winding of transformer TR1 (an encapsulated PCB-mount power transformer), which is recast as an output transformer here. The output signals from the triodes are fed to the control grids of the pentodes via networks C3/R3 and C5/R16. The two secondary windings of TR1 drive the loudspeaker. You can fit jumpers on JP1 to connect these windings in series or in parallel. JP2 and R26 provide the negative feedback path from the output to the input. You can experiment with different values for R26, or you can disable negative feedback entirely by removing the jumper on JP2. A jumper can be fitted on JP3 to connect the input ground to the loudspeaker output.

Trimpot P2 sets the quiescent current of the tubes. It adjusts the negative bias on the control grids of the pentodes, which is derived from the negative filament voltage V_{ff}. P1 adjusts the balance between the two tubes.

Header K3 allows the quiescent currents of the two tubes to be measured individually using a voltmeter. The scale ratio is 10 volts per ampère, so with a current of 25 mA (the recommended value) the reading is 250 mV. You should let the tubes warm up properly before setting the quiescent current.

All the filaments are connected in series, and the filament current should be 300 mA. To prevent excessive filament current at switch-on (inrush current) due to the low cold resistance of the filaments, a simple current source built around an LM337 is included on the circuit board. A simple negative voltage supply with an output of at least 31 V is all you need for the filament current.

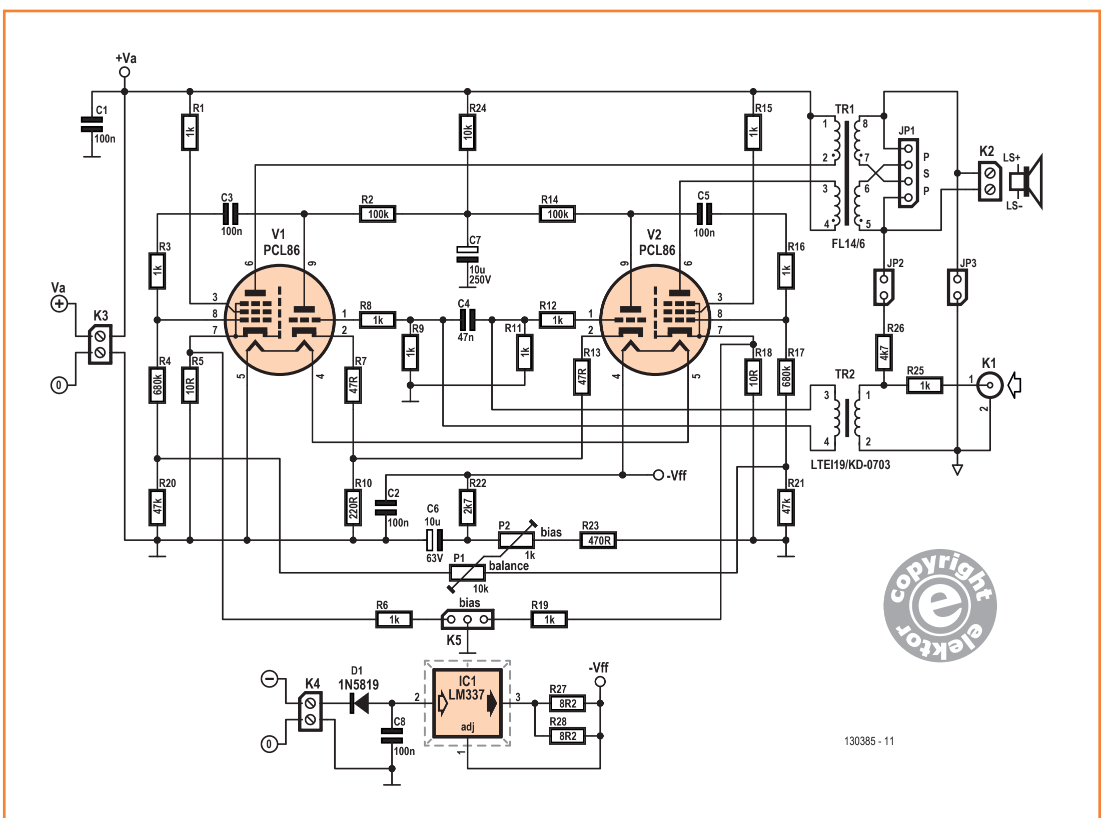


Figure 1.
This simple tube amplifier is built around a pair of PCL86 (14GW8) tubes.

The allowable range of the anode supply voltage V_a is 160 V_{DC} to 200 V_{DC}. A suitable power supply for the amplifier is described below.

PCB design

Elektor Labs designed a double-sided PCB layout for the tube amplifier. The PCB design is posted at [1]. The board has a large ground plane on one side to shield the mostly high-impedance components on the other side, with extra-wide isolation on account of the high anode voltage. The input transformer is located as far away from the output transformer as possible to minimize the magnetic coupling between the two transformers.

Component List

Amplifier

Resistors

(1%, .6W, 350V):
R1,R3,R6,R8,R9,R11,R12,R15,R16,R19,R25 = 1kΩ
R2,R14 = 100kΩ
R4,R17 = 680kΩ
R5,R18 = 10Ω
R7,R13 = 47Ω
R10 = 220Ω
R20,R21 = 47kΩ
R22 = 2.7kΩ
R23 = 470Ω
R24 = 10kΩ
R26 = 4.7kΩ
R27,R28 = 8.2Ω
P1 = 10kΩ trimpot, 0.15W, horizontal mounting
P2 = 1kΩ trimpot, 0.15W, horizontal mounting

Capacitors

C1,C3,C5 = 100nF 250V, 5%, MKP, 5, 7.5, 10, 15 or 22.5mm pitch
C2,C8 = 100nF 100V, 10%, MKT, 5 or 7.5mm pitch
C4 = 47nF 250V, 10%, MKP, 5, 7.5 or 10 mm
C6 = 10µF 100V, radial, 2.5mm pitch, 6.3mm diam.
C7 = 10µF 250V, radial, 5mm pitch, 10 mm diam.

Semiconductors

D1 = 1N4007
IC1 = LM337

Tubes

V1,V2 = PCL86 or 14GW8

Miscellaneous

TR1 = power transformer, Block type FL 14/6, 2x115V primaries; 2x6V secondaries; 14 VA
TR2 = 1:5 audio transformer, e.g. LTEI19/KD-0703 (Conrad Electronics # 515701-89)
K1,K5,JP2,JP3 = 2-pin pinheader, 0.1'' pitch
K2,K4 = 2-way PCB screw terminal block, 5mm pitch
K3 = 2-way PCB screw terminal block, 7.5mm pitch
JP1 = 4-pin pinheader, 0.1'' pitch
3 or 4 jumpers for JP1, JP2, JP3
V1,V2 = tube socket, ceramic, 9-pin Noval, PCB mount
Heatsink for IC1, 30K/W (e.g. Fischer Elektronik SK 12 SA 32)
PCB # 130385-1, see [1]

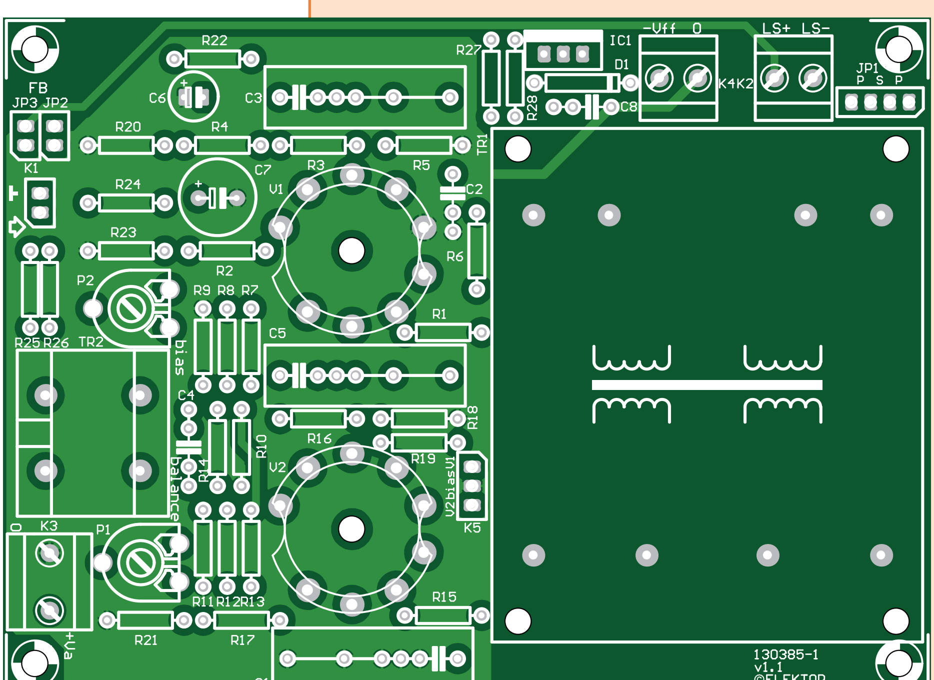


Figure 2.
The PCB layout for a single amplifier channel; two of these must be assembled for a stereo amplifier.

Power Supply

Resistors

R1 = 270kΩ, .5W, 350V
R2 = 10kΩ, .25W, 250V

Capacitors

C1–C4 = 4.7nF 400V
C2–C8 = 47nF 100V ceramic
C9 = 100µF 350V
C10,C11 = 1000µF 50V

Semiconductors

D1–D4 = 1N4007
D5–D8 = 1N5819
D9,D10 = LED, low-current

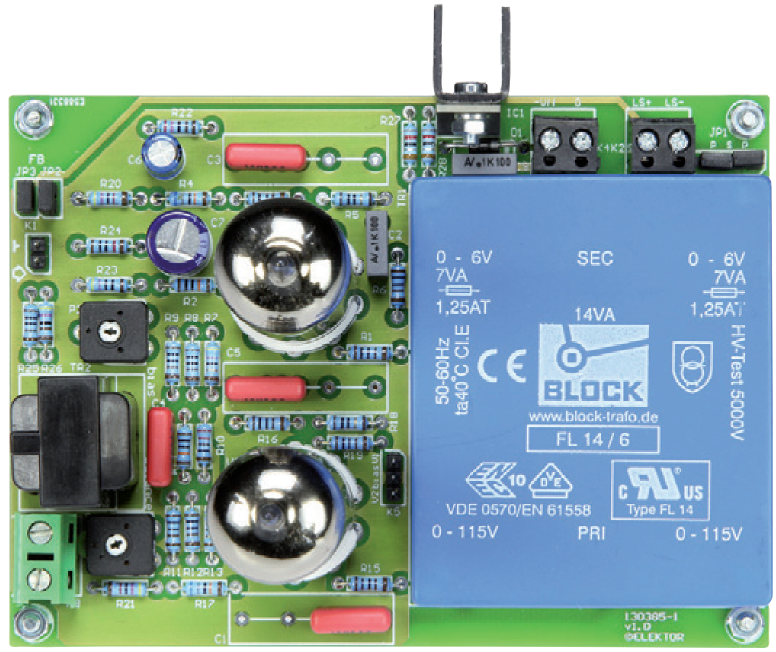
Miscellaneous

F1 = fuse, 200mA/T (115VAC: 400mA/T) with holder
F2 = fuse, 750mA/T with holder
K1,K2 = 2-way PCB screw terminal block, 7.5mm pitch
K3 = 2-way PCB screw terminal block, 5mm pitch
TR1 = power transformer, Block type FL 30/12, 2x115V primaries; 2x12V secondaries, 30VA
TR2 = power transformer, Block type FL 18/12, 2x115V primaries; 2x12V secondaries, 18 VA

To keep the PCB as compact as possible, several tracks are routed underneath the input transformer. As a consequence, this amplifier must never be powered directly from the rectified AC line voltage.

The photos clearly show how everything should be mounted. All of the components are normal leaded types. PCB screw terminal blocks are used for the supply and loudspeaker connections. Due to the high anode voltage, a terminal block with a lead spacing of 7.5 mm is used for K3. A heat sink is fitted on the LM337 regulator. The size depends on the level of the supply voltage; the temperature rise should preferably be kept below 40 degrees.

After assembling the circuit board, you should immediately fit several jumpers before connecting the supply voltages. First fit a jumper on JP2 (negative feedback) and JP3 (ground connection between input and output). You can use pinheader JP1 to select either series or parallel connection of the two secondary windings. Fit two jumpers to connect them in parallel, or fit one jumper in middle of JP1 for to connect them in series.

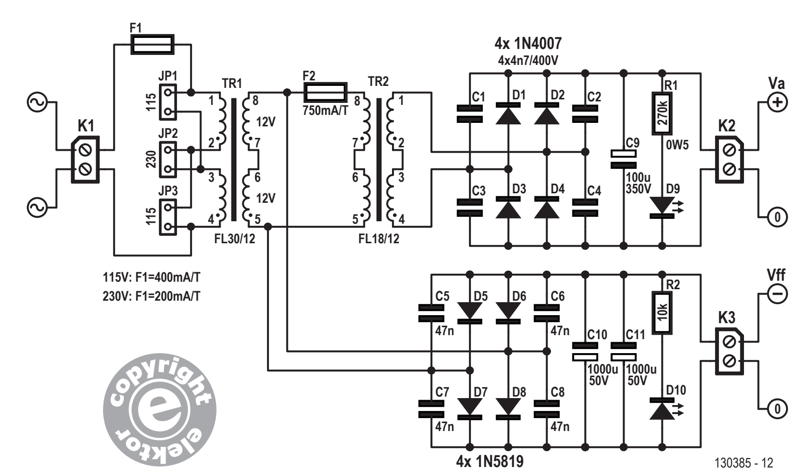


Power supply

There are various ways to implement the power supply for the tube amplifier. For the prototype, the author built an external switch-mode supply with a transformer for galvanic isolation from the AC line. The guys at Elektor Labs opted for another approach with a power supply using two PCB-mount power transformers with their secondary windings connected back to back. This results in good galvanic isolation from the AC line without the need for a transformer that may be hard to obtain. It also allows the filament voltage to be derived from the secondary voltage of the first transformer by means of bridge rectifier D5–D8, filter capacitors C10/C11 and associated components. This is why the first transformer is a 30-watt type and the other, an 18-watt type. If you are on a 115 V grid fit jumpers JP1 and JP3, and a 400 mA slow-blow fuse for F1. The high voltage at the output of TR2 will not be the targeted 230 volts, but instead lower because the data sheet says that the no-load voltage of TR2 is 1.22 times higher than the nominal voltage. The actual voltage is therefore 188.5 V (230/1.22). This will drop by an additional factor of 1.22 under load, so we end up with a plate voltage in the vicinity of 154 V. As PCB transformers of this sort are not designed to be used backwards, large losses will occur when the filter capacitors are being charged by short current spikes. In our test setup, the measured voltage with no input signal was slightly more than 188 V_{DC}, dropping by ten volts or so at maximum output power. Pay attention to the voltage ratings of the various capacitors in the amplifier circuit. With a higher supply voltage, it is advisable to use 350-V types instead.

For both supply voltages there is an LED indicator that shows whether the voltage is present. The LED for the high voltage supply also forms a

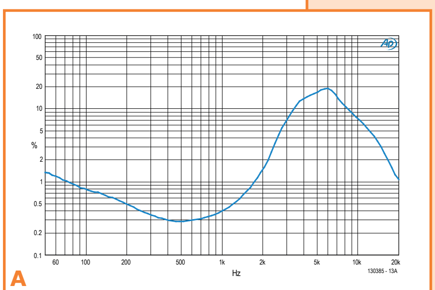
Figure 3.
A suitable power supply configuration for the amplifier. Here two power transformers are connected “back to back”.



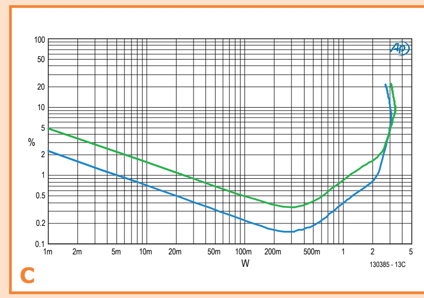
A Few Measurements

Plot A shows the total harmonic distortion plus noise versus frequency at an output power of 1 watt into 8 Ω at 1 kHz. The two secondary windings are connected in parallel, and feedback jumpers JP2 and JP3 are fitted. This clearly shows the effect of the inferior characteristics of a power transformer in comparison to a real output transformer.

Plot B shows the frequency characteristic of the amplifier at 1 watt into 8 ohms. The two secondary windings are connected in parallel. The curve with the larger bandwidth was measured with negative feedback (input level 850 mV); the other curve was measured without negative feedback (input level 235 mV). For comparison, the two curves were normalized at 1 kHz.



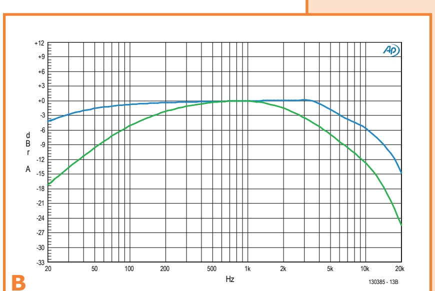
A



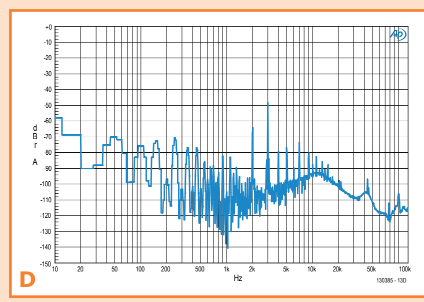
C

Plot C shows the total harmonic distortion plus noise versus output power with an 8-Ω load. The two secondary windings are connected in parallel. The green curve is with negative feedback, and the blue curve is without.

Plot D shows the Fourier spectrum of a 1 kHz signal with 1 W into 8 Ω and the secondary windings connected in parallel. The total harmonic distortion plus noise is 0.4%. Along with the two harmonics of the 1 kHz signal, a broad noise spectrum can be seen. It is caused by the ripple on the supply voltage.



B



D

Measured Performance

Input sensitivity	(parallel connection with negative feedback):	1.7 V (P_{max} = 3.1 W)
	(series connection with negative feedback):	1.3 V (P_{max} = 1.04 W)
Input impedance	at 1 kHz:	1.01 kΩ
Continuous output power	(parallel connection):	> 3 W
	(series connection):	approx. 1 W
Power bandwidth	(parallel connection with negative feedback):	28 Hz – 6.2 kHz
		max. 12 kHz at 0.1 W
	(series connection with negative feedback):	46 Hz – 4.8 kHz
S/*N* ratio at 1 W / 8 Ω	(parallel connection with negative feedback):	> 64 dB (> 72 dBA)
THD + noise at 1 W / 8 Ω	(parallel connection with negative feedback):	0.4%
	(series connection with negative feedback):	5%
Damping factor at 1 W / 8 Ω	(parallel connection with negative feedback):	2.75
	(series connection with negative feedback):	1.25
Current consumption	on high voltage rail with parallel connection:	52 mA (quiescent; V_a = 188 V)
		59 mA (P = 3 W, V_a = 181 V)

load for the supply, to prevent the voltage from rising too high right after switch-on when the tubes are not yet drawing any current.

If you wish, you can experiment with a somewhat higher supply voltage and different settings for the tubes. The author managed to achieve an output power of about 10 watts (at 10% THD) with the prototype, using a beefier power supply with an anode voltage of 200 V and a higher quiescent current.

Putting it all together

After you have assembled the amplifier board and the power supply, you can adjust the quiescent current and the balance after letting the tubes warm up for a while. As already mentioned, you can start with a quiescent current of 25 mA per tube (250 mV on K5) and adjust the balance so that the currents through the two tubes are exactly the same.

Then you're ready to start listening to the amplifier. The author obtained the best results with the two output windings connected in parallel and a 4.7-kΩ resistor in the feedback path, as indicated on the schematic.

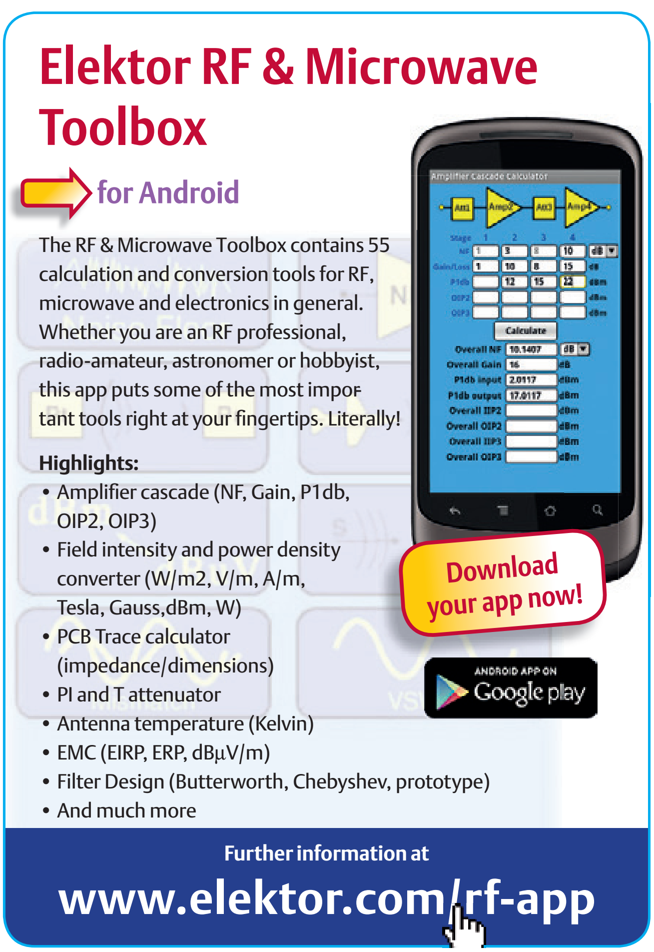
With various speakers (either box enclosure or panel type, and regardless of the impedance) it sounds like what you would expect from a tube amplifier: good detailing at low volume and smooth overdriving at high signal levels. Although the measured bandwidth at full power may be somewhat disappointing, in practical listening situations you don't miss anything because the bandwidth is fairly large at lower output levels, extending to above 10 kHz.

(130385-I)

Internet Link

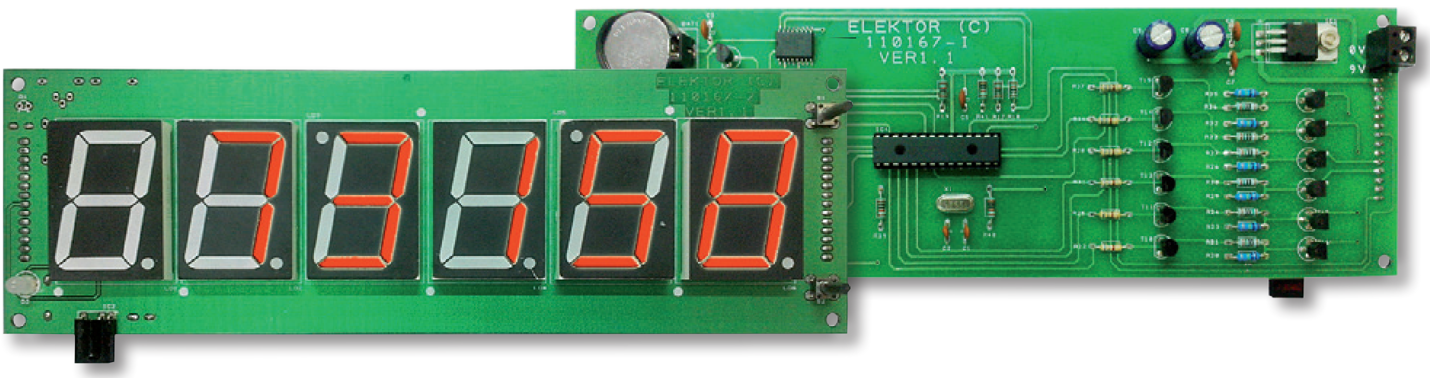
[1] www.elektor-magazine.com/130385

Advertisement



Pretty Accurate Digital Wall Clock

.000202943 % slow or fast max.



By **Niras C.V.**
(India)

The crucial component in this project is a Maxim IC type DS3231, qualified by its manufacturer as an "extremely accurate I^2C real time clock (RTC) with integrated temperature compensated crystal oscillator (TCXO) and crystal." Maxim also says that the integration of the crystal resonator enhances the long-term accuracy of the device, guaranteeing a maximum error of less than 64 seconds over a year, and over a temperature range 0 to 40 ºC (32 to 104 ºF). The device incorporates a battery input which maintains running of the device in the absence of external power.

Circuit description

Hyper-accurate as it may be, the DS3231 is unable to do anything useful without the help of a few other components. Referring to the schematic in **Figure 1**, A PIC16F876A microcontroller (IC4) is used to interface to the RTC (IC3) as well as drive a bunch of 7-segment LED displays (discussed further on). The microcontroller has an I^2C port which makes for easy interfacing to the DS3231 RTC.

Features

- PIC16F876A controlled
- Maxim DS3231SN RTC chip
- 1.5 inch red 7-segment LED displays
- Max. error 64 seconds per year
- Optional IR remote control
- 9V @ 500 mA max. power supply (DC adapter)
- Free C source code
- Free DesignSpark PCB files

The microcontroller clock operates at 20 MHz due to quartz crystal X1 and its two load capacitors C1 and C2, not forgetting the appropriate control word in the microcontroller "config settings". Port lines RA0-RA3, RA5 and RC1 of the PIC micro switch individual LED displays on and off through drivers/level changers T10-T21. Note the use of an npn/pnp transistor pair on each line to handle (1) the level conversion from 5-V swing (PIC side) to 9-V swing (display side), and (2) feeding current from the 9-V rail to each display via its CA (common anode) terminal under multiplex control.

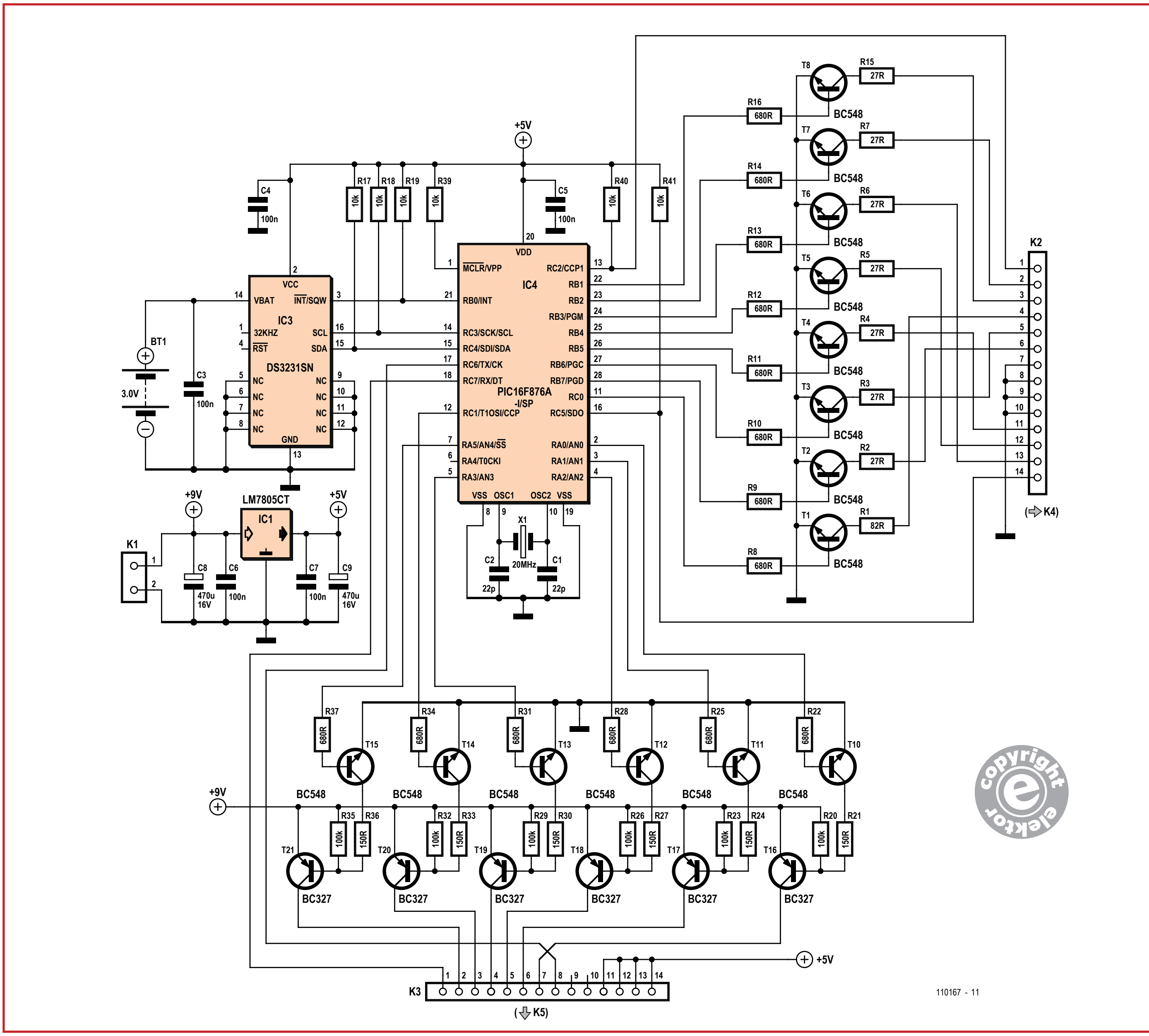


Figure 1. Schematic of the microcontroller section. K3 connects to the common anodes of the 7-segment displays, K2 to the bussed segments.

The supply voltage for the PIC micro is regulated with a 5-V linear regulator (IC1), and the displays are connected to a higher (9-V) unregulated supply. This makes the design capable of driving bigger displays with a larger voltage drop per segment—such as 6.8 V—due to more LEDs in series. PORTB pins RB1-RB7, and RC0, activate the individual LED segments through an array of switching transistors T1-T8.

A 9-V, 500-mA unregulated power supply (power adapter) is sufficient for the circuit to provide good brightness, and a CR2030 lithium battery (BT1) is used as a backup supply for the RTC. On a prototype of the clock, a current consumption of 320 mA was measured on account of the display section operating in multiplex mode.

The design is for a simple 24/12 hour clock which incorporates six 7-segment LED displays and two input switches. The display section schematic is shown in **Figure 2**. One of the attractions of this design is it 1.5” bright red 7-segment LED displays, so the clock can be read easily from considerable distances. The displays are multiplexed at 1 kHz but provide sufficient average currents for high brightness. The individual segments (a-g) of displays LD1-LD6 and the decimal point (dp)

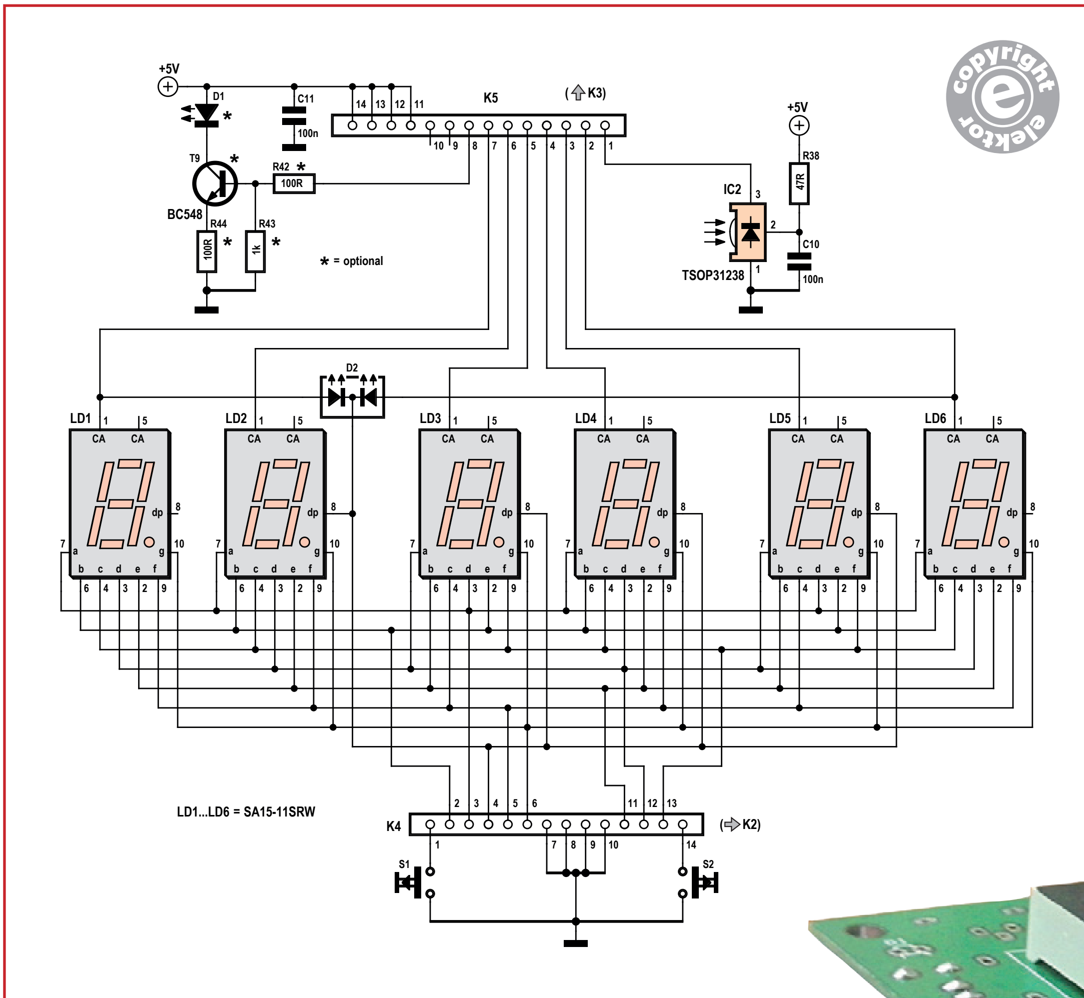
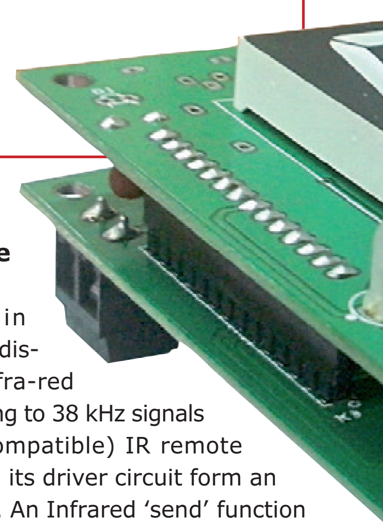


Figure 2.
Schematic of the display section. LD3 and LD5 are mounted upside down with adapted control in software, to create colons between hours and minutes, and minutes and seconds.

of the displays LD2-LD5 are connected together and brought out to pins on K4. Displays LD3 and LD5 are mounted upside down on the board (note the dots!) to create an oblique colon (:) between hours and minutes, and minutes and seconds. The PCB design takes care of this. The LED displays are multiplexed through their common-anode (CA) terminals, brought out to pins on K5 connecting to K3 on the microcontroller board. Pushbuttons S1 and S2 for controlling the clock functions are connected to the microcontroller also by way of K4 on the display board connecting straight to K2 on the microcontroller board. The display brightness can be adjusted as a user setting stored in the PIC micro. This function may be implemented using the flag 'DisplayOn' in the code.

Finally, on the display board, a bicolor LED (D2) is used to indicate AM (green) or PM (red if the clock operates in 12 hour mode.



Infrared remote control: software ramifications

The TSOP31238 in position IC2 on the display board is an infra-red receiver IC responding to 38 kHz signals from an RC5 (or compatible) IR remote control. LED D1 and its driver circuit form an infrared transmitter. An Infrared 'send' function was originally not implemented, hence if you do not need it, D1 and associated parts may be omitted.

As the Wall Clock project progressed, changes were made to the source code, as follows:

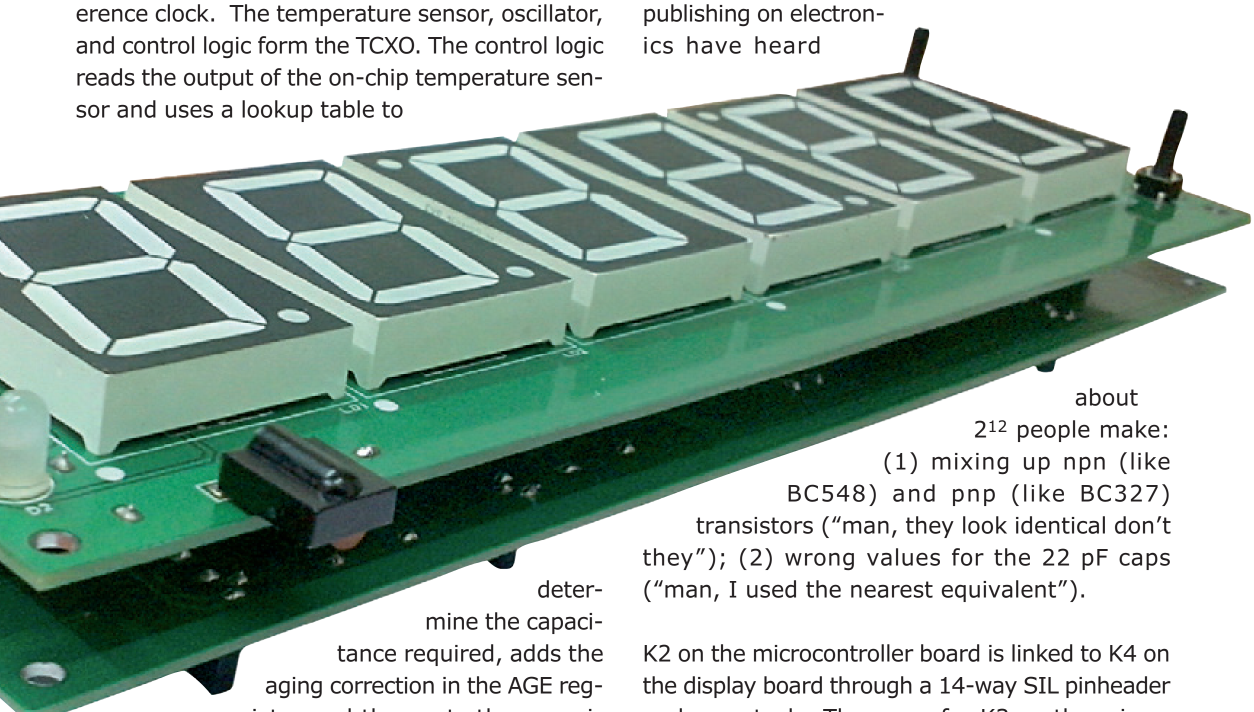
- Xtal changed to 20 MHz enabling the PIC to execute RC5 protocol section without any hiccups in the original code.
- RC5 protocol section implemented in the

interrupt routine, microcontroller now scans port pin RC7 status; checks received signal against RC5 protocol.
- In the main section: addresses and commands get extracted from the received RC5 data.
- If the command received is 16 or 17 (address and all other commands except 16 & 17 ignored) the program takes it as buttons S1 or S2 being pushed, respectively.
- Implemented Watchdog functionality in-order to avoid any uncertainty in the program.

With IR reception implemented you can control the clock from a distance wielding your (RC5 compatible) remote. Don't tell the kids!

Real Time Clock (RTC) type DS3231

The DS3231 is a serial RTC driven by a temperature compensated 32-kHz crystal oscillator (TCXO), and provides a stable and accurate reference clock. The temperature sensor, oscillator, and control logic form the TCXO. The control logic reads the output of the on-chip temperature sensor and uses a lookup table to determine the capacitance required, adds the aging correction in the AGE register and then sets the capacitance selection registers. The AGE function is not used in this project—use of the aging register is not needed to achieve the given accuracy. With the clock source, the RTC provides seconds, minutes, hours, day, date, month, and year information, all accessible through the I2C bus. The device monitors its VCC level to detect power failures and to automatically switch to the backup supply when necessary.



On first power up the PIC microcontroller initializes the RTC to generate a 1-Hz square wave at the $\overline{\text{INT}}$/SQW pin by writing 0x00 to the RTC's control register. This is connected to the external interrupt (INT) of the microcontroller, effectively setting the PIC's INTF flag on each High-to-Low transition at RB0/INT. This is used to initiate a reading of the RTC's time registers. A 1-Hz (i.e. 1-second) flashing colon can also be derived from polling the status of RB0/INT.

Construction

In terms of hardware the clock is divided into two sub-circuits: microcontroller/driver and display. Each is built on its own circuit board of which the component overlays are shown in **Figure 3**. Excepting the RTC, the entire design is implemented in old skool through-hole (TH) parts so should be easy to build if you apply care and precision in reading and soldering. Here are just two mistakes Elektor tech staff in their 35_{hex} years of publishing on electronics have heard about 2^{12} people make: (1) mixing up npn (like BC548) and pnp (like BC327) transistors ("man, they look identical don't they"); (2) wrong values for the 22 pF caps ("man, I used the nearest equivalent").

K2 on the microcontroller board is linked to K4 on the display board through a 14-way SIL pinheader and receptacle. The same for K3 on the microcontroller board and K5 on the display board. These connections allow the display board to be stacked on top of the microcontroller board.

Operation

The circuit has only two pushbuttons to perform user control and adjustments. Press S1 for 1 second to take the circuit into time adjusting mode. Blinking digits means they're open to having the

COMPONENT LIST

Combined for microcontroller board, display board

Resistors

(5%, 0.25W)

R1 = 82Ω
R2-R7,R15 = 27Ω
R8-R14,R16,R22,R25,R28,R31,R34,R37 = 680Ω
R17,R18,R19,R39,R40,R41 = 10kΩ
R20,R23,R26,R29,R32,R35 = 100kΩ
R21,R24,R27,R30,R33,R36 = 150Ω
R38 = 47Ω
R42-R45 = 1kΩ

Capacitors

C1,C2 = 22pF ceramic
C3-C7,C10,C11 = 100nF
C8,C9 = 470µF 16V radial

Semiconductors

D1 = LED, red, 5mm
D2 = LED, bicolor, e.g. Kingbright L-93WEGW
IC1 = LM7805CT
IC2 = TSOP31238 (Vishay Semiconductor)
IC3 = DS3231SN (Maxim Integrated Products)
IC4 = PIC16F876A-I/SP, programmed, Elektor Store # 110167-41
LD1-LD6 = SA15-11SRWA 7-segment display, CA (Kingbright)
T1-T15 = BC548
T16-T21 = BC327

Miscellaneous

BT1 = 3V Lithium button cell with holder.
K1 = 2-pin PCB screw terminal block, 0.2’’ pitch.
K2/K4, K3/K5 = 14 pin SIL connector pair (pinheader and receptacle).
S1,S2 = pushbutton, PCB mount, tactile feedback, e.g. Alps SKHHALA010.
X1 = 20MHz quartz crystal.
PCB # 110167-1 (microcontroller board).
PCB # 110167-2 (display board).

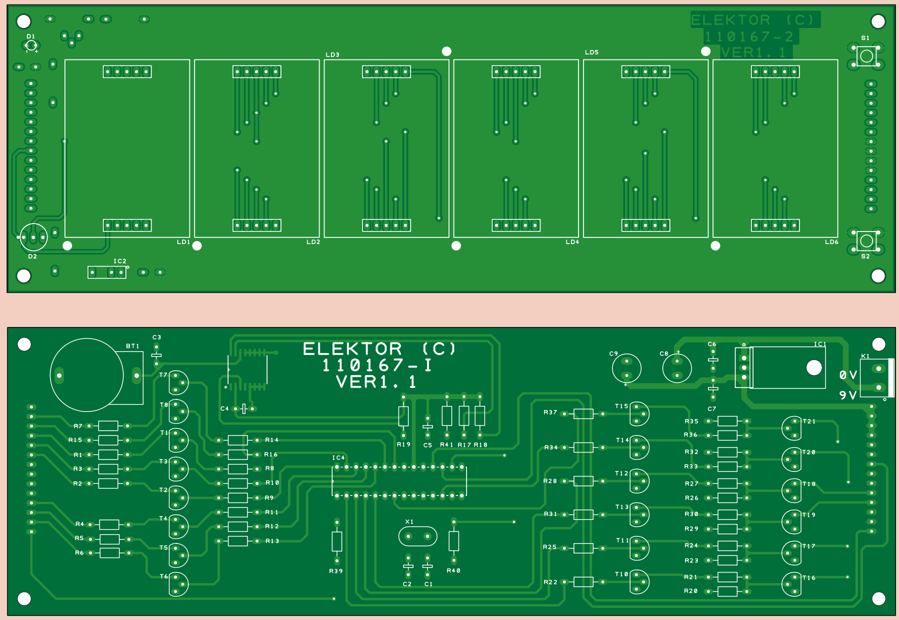


Figure 3. Separate printed circuit boards were designed for the microcontroller display sections, both using DesignSpark PCB. Boards shown at XX per cent of actual size. The display board accommodates Kingbright 1.5-inch (38 mm) SA15-11SRWA 7-segment displays.

displayed value changed by you, the user. Digits can be selected individually by a short press of S1—from seconds, through minutes, hours, 24/12 hour selection, to exit. Pressing S2 increments the selected digit to its highest value then rolls over, except for the ‘seconds’ digits, these will be changed to zero. Also, if the seconds exceed 30, the minutes’ digits will increment by one. Pressing S2 (‘up’) with the clock not in time adjustment mode shows the temperature in ºC with a minus sign for really cold rooms. The temperature sensor has an “accuracy” of ±3 ºC and a resolution of 0.1 ºC.

Conclusion

The circuit provided here is a basic wall clock with sizeable (1.5”) displays. Features such as alarm or synchronization between PC via infrared linking are optionally possible. The schematic, PCB design and PCB Gerber files produced by Elektor Labs India Dept. using DesignSpark PCB are available for free downloading at [1]. The same applies to the PIC source code written in C.

The project is expressly pitched at everyone wishing to extend it with their own functionality by adding software of their own creation, like display brightness adjustment mentioned above. Let the designer and the community at www.elektor-labs.com know how you are getting along.

(110167)

Internet Reference

[1] www.elektor-magazine.com/110167

elektor
PCB Service
powered by Eurocircuits

PCBs
Multilayer
Prototypes
Small series

Professional Quality
Trusted Service
Secure Ordering

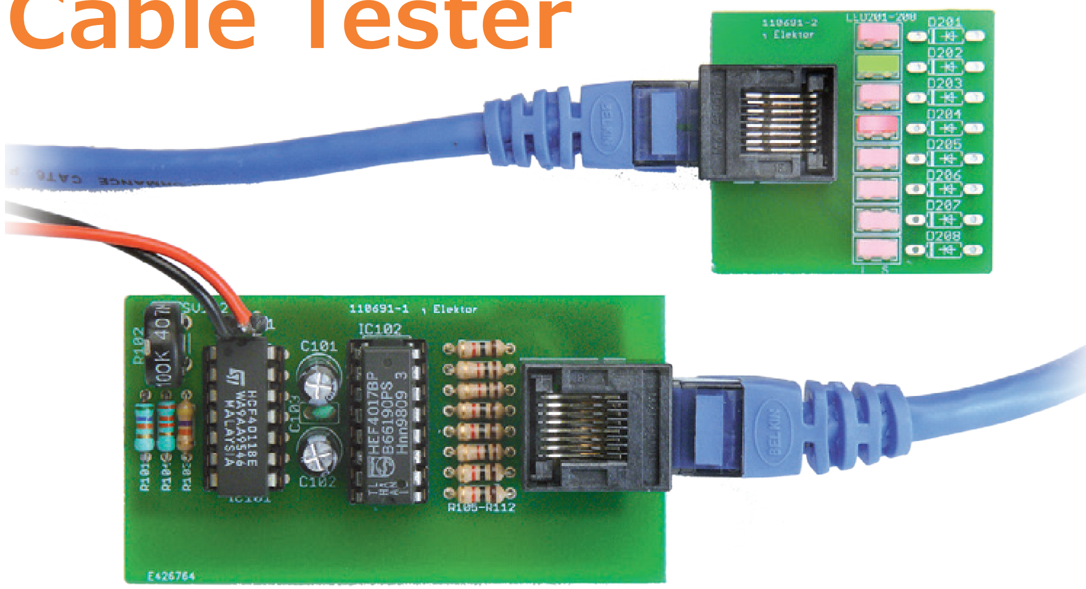
Elektor PCB Service at a glance:

- 4 Targeted pooling services and 1 non-pooling service
- Free online PCB data verification service
- Online price calculator available
- No minimum order value
- No film charges or start-up charges

Delivery from 2 working days

Further information and ordering at www.elektorpcbservice.com

RJ45 'Running-Lights' Cable Tester



By Flemming Jensen
(Denmark)

Remarkably, it has no microcontroller (hooray!) and no common ground lead (huh?).

This cheap & cheerful circuit is indispensable if you are suspicious about a defective RJ45 cable in the patch cabinet or anywhere else for that matter. Connect the Master (sender) to one end of the cable under test (C.U.T.) and the Slave (receiver) to the other end. If the LEDs light up in succession then the cable is okay. If not, flaunt the cable to the IT Manager, drop it in His trash bin (with a fanfare) and ask a raise as a reward for keeping the company in business.

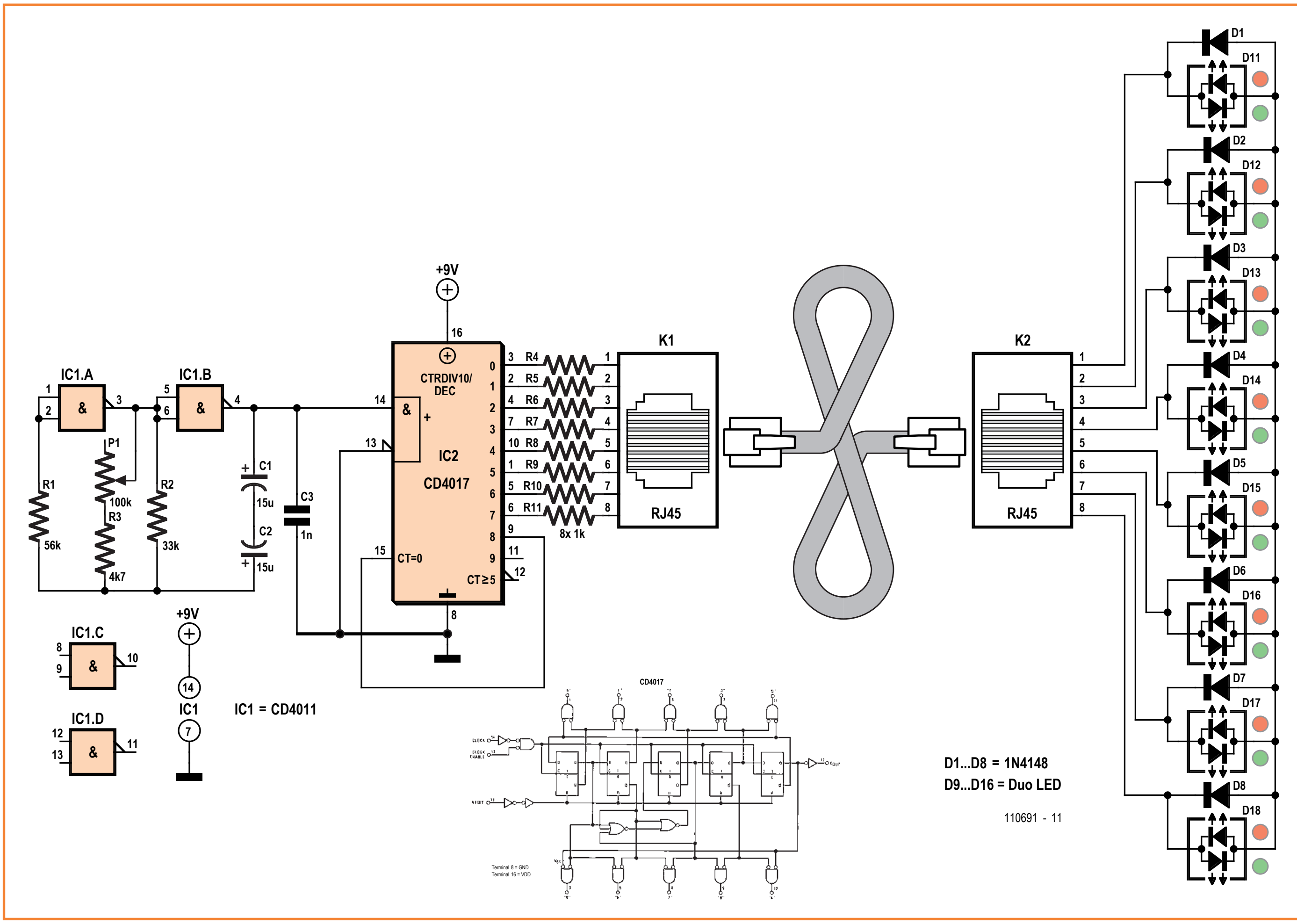
Let's look at this clever money making gadget then. Remarkably, it has no microcontroller (hooray!) and no common ground lead (huh?).

Each of the 4017's counter outputs CT0–CT7 has its own LED and wire in the cable under test but no wire is required for the ground return current. When one of the 4017 counter's outputs is logic High due to the clock pulses from IC1 a current flows through the corresponding LED and normally returns to ground (GND) via diodes D1-D8 or the reverse connected LEDs, by way of the Low outputs of the 4017. You either fit eight dual colour LEDs in positions D11-D18, or eight diodes in positions D1-D8 and eight high efficiency LEDs in positions D11-D18.

No buffers are needed between the 4017 and the LED array at the other end of the C.U.T. considering the IC is capable of sourcing the necessary current. Current limiting resistors R4–R11 are required though in view of the 9-volt supply voltage. The speed of the running lights is adjustable on preset P1 within a certain range.

The LEDs tells a thing or two on the cable under test, as follows.

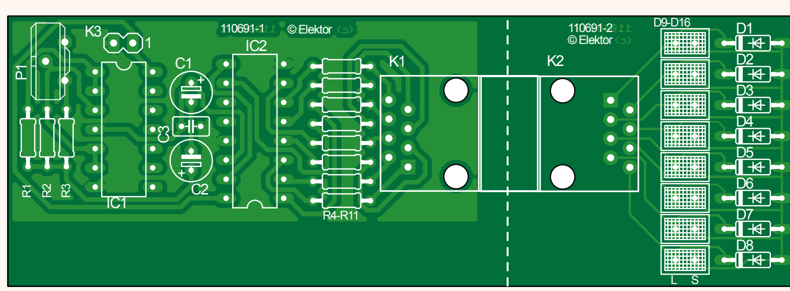
- RJ45 cable all right and straight through: LEDs light in succession like a small running lights.
- One or several wires are shorted: one or several LEDs light all the time.
- One or several wires broken: one or several LEDs off permanently.
- One or several wires connected to a wrong pin: running lights erratic (jumping around).



The Master and Slave circuits are built as separate units pictured here. The two boards are separated by sawing along the dashed line on the component overlay. Note again that on the Slave board you fit either the dual LEDs or the single color LEDs and diodes D1-D8. The photograph shows the first option.

(110691)

COMPONENT LIST



Resistors
R1 = 56kΩ
R2 = 33kΩ
R3 = 4.7kΩ
R4–R11 = 1kΩ
P1 = 100kΩ preset, vertical

Capacitors
C1,C2 = 15µF 16V radial
C3 = 1nF

Semiconductors
IC1 = CD4011 or HEF4011
IC2 = CD4017 or HEF4017
D1–D8 = 1N4148*
D9–D16 = LED, dual-colour low-current, or single colour LED*

Miscellaneous
K1,K2 = RJ45 CAT5E socket, PCB mount, e.g. Farnell # 2060718

PCB # 110691

* either/or (see text)

Multi I/O for FPGA Development Board (2)

Programming in VHDL

By **Andreas Mokroß, Dominik Riepl, Christian Winkler** and **Professor Thomas Fuhrmann** (Germany)

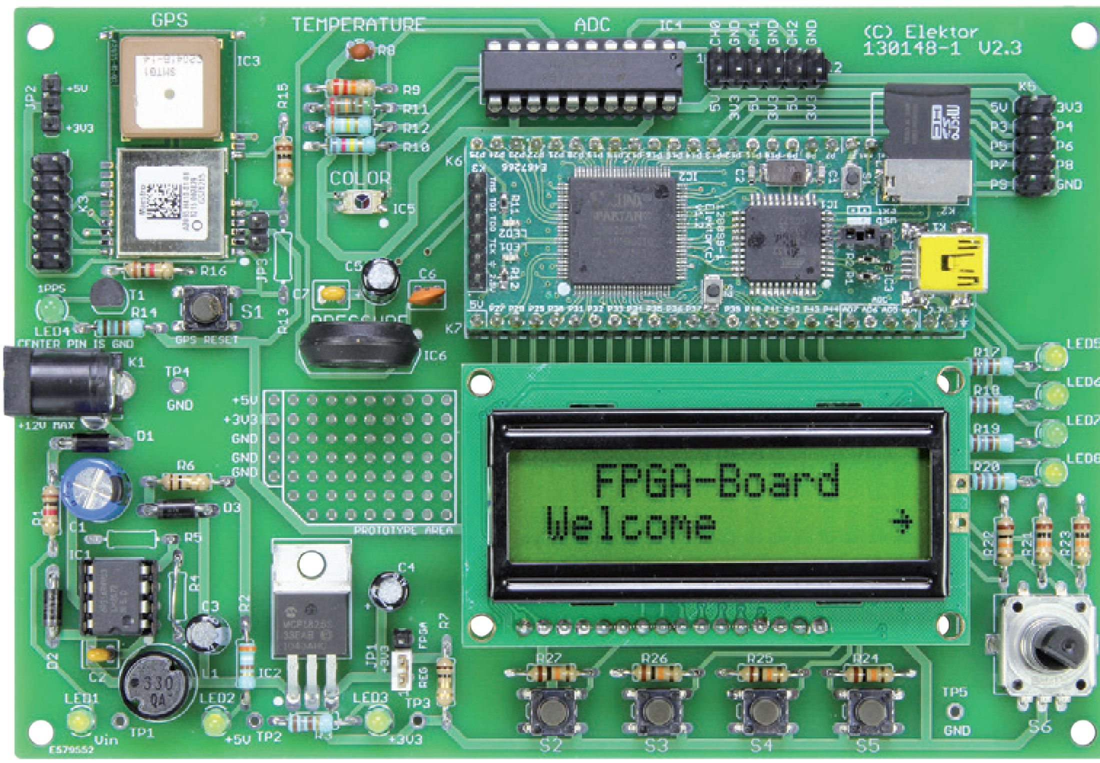


Figure 1.
The board showing the welcome screen.

In the December 2013 edition we described the FPGA development board hardware. With this board it's a simple job to interface the FPGA to real world devices and events. The extensive range of peripherals include sensors and an LC display. The next step is to program the FPGA and get them talking to one another.

The Elektor FPGA board was originally featured in December 2012. It has the part number 120099-91 and can be ordered directly from the Elektor Store. This FPGA board alone does not contain any peripheral chips. To make it more useful as an educational aid an expansion board was developed by the students at the Ostbayerischen Technischen Hochschule in Regensburg, Germany. The expansion board has a number of peripheral sensors and an LCD which interface to the FPGA board. They can be easily configured and controlled using VHDL and put to use in all sorts of applications. The complete development environment with both boards is shown in **Figure 1**. In part 1 [1] of this project the development board hardware was described, in this second part we describe how these peripherals can be controlled using VHDL. This will give you a good grounding in the technology so that you can go on to use it for your own applications.

The complete project has been developed using the XILINX ISE 14.5 Design Suite which is freely available to download from the Internet [2].

A hierarchical approach

For large projects in VHDL it is sensible to approach the design in the way you would a software project by breaking down the program into small manageable parts. Unlike software VHDL doesn't use functions or classes, instead we use descriptions of hardware, the so-called Module. Each module is described by its own file and has its own defined interface to the outside world. A module functions as a self-contained unit and can be simulated with the help of a Test-Bench. It can be seen as a functional black box which integrates into the complete project. The use of modules greatly simplifies the process of system debugging.

Based on the sensors, display, control elements and their control, the VHDL project is divided into the following areas (see **Figure 2**):

- `menu_control`: The central module; this is where all the data comes together and the menu functions are taken care of.
- `taster`: Interfaces to the pushbuttons with debounce logic.
- `lcd`: Drives the LCD and loads the display data.
- `gps_control`: Controls the GPS module and receives Global position information and time of day.
- `ADC_control`: Controls the A/D converter.

At the top level the individual modules are combined and no logic process is described. The connections are made in the sub-modules defined in the `top_level` description using the `component` key word. The compiler is thus informed which components are used and what inputs and outputs are available. Connections to the module are defined in the `port map`, where the input and output of the sub modules are linked to the signals defined in the top level.

While VHDL is a Hardware Description Language, they get translated directly into gates and because the modules work in parallel it is not necessary to consider the order of the modules. All modules have a connection to the 8 MHz clock so that every process is synchronous with the clock. Apart from this the module's data output has an enable signal which goes to a logic '1' state for one clock period when all the processing in the module is completed. It indicates that output data is stable and can be used elsewhere.

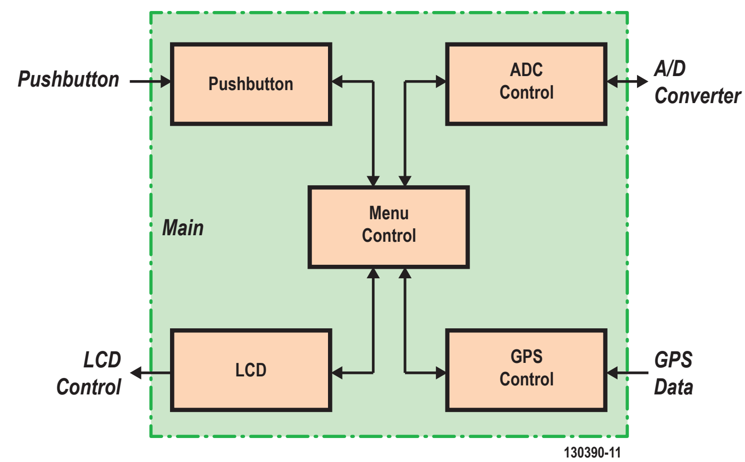


Figure 2.
The project's block diagram.

Package: self_defined_types

This Package contains the global definitions for the complete project and frequently used data types and conversions to make them more readable.

It defines the following data types:

- `BYTE`: An 8-bit wide array of type `std_logic`
- `BYTE_ARRAY`: An arbitrary width array of type `BYTE`

In addition the function `HEXtoASCII` is defined which converts a 4-bit wide `std_logic_vector` with a hexadecimal value into a displayable ASCII character.

The packages are linked into this and any other library using the command:

```
library work;
use work.self_defined_types.all;
```

The ADC Module

The first module described here is used to control the A/D converter and read the output data. The module uses a 3-bit long `std_logic`-input vector `in_channel` to select the analog channel and an 8-bit long `std_logic`-output vector `out_adval` to output the data. The other inputs and

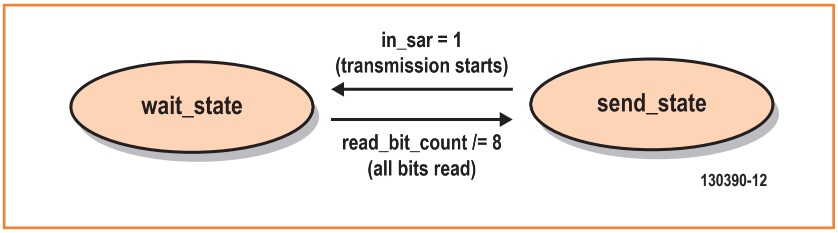


Figure 3.
State diagram of the ADC read processes.

outputs are necessary to communicate with the IC (`in_sar`, `in_do`, `out_clk`, `out_di`, `out_not_cs`, `out_not_se`).

The Clock Process

According to the data sheet of the A/D converter [3] it should be provided with a clock signal in the range from 10 kHz to 400 kHz. For this application we use 100 kHz. The clock is generated a dedicated process, see **Listing 1**.

The counter `cnt_clock` is incremented on each rising edge of the 8 MHz input clock `in_clk` and the internal clock `adc_clk` at `(const_divider – 1) / 2` and `(const_divider – 1)` which is inverted, so that the time that the timer runs from 0 to reset corresponds exactly to a clock period of the generated clock.

The constant `const_divider` is the ratio of the clock frequency of the external oscillator and the required converter clock frequency:

$$const\_divider = \frac{oscillator\ clock\ frequency}{converter\ clock\ frequency}$$

Apart from this the Enable signal `rising_clk` and `falling_clk` are generated for state machine communication with the converter.

The reading process: implementing a State Machine

For the sake of clarity control of the FPGA functions are implemented in modules using state machines. The following implementation will be for the A/D converter. The state transition diagram of the machine is given in **Figure 3**. Each state is represented by a circle and transition to another state is indicated by an arrow labeled with the transition condition.

Two states are required to read data from the ADC: In `wait_state` the state machine waits until data is available indicated by the SARS output from the ADC going to a logic ‘1’. The state machine then jumps to the `send_state` which reads the 8-bits from the converter.

To describe the state diagram in **Figure 3** using VHDL a separate data type `state_type_read` is defined with all the possible states and then a `signal` that is this type. The compiler converts this construct into a timer. For a developer the approach used here is more easily understandable.

It is written as a process using a case statement, in which all the possible states of the machine are described, see **Listing 2**.

The two separate states of the machine can now be described:

- `wait_state`: When the A/D converter outputs a new word the SAR status output is set to a logic ‘1’. This makes the machine state change to `read_state`.
- `read_state`: This is where the ADC serial

Listing 1

```
constant const_divider: integer := 80;
signal cnt_clock: integer range 0 to const_divider - 1 := 0;
signal rising_clk: std_logic;
signal falling_clk: std_logic;
signal adc_clk: std_logic := '0';

process(in_clk)
begin
  if rising_edge(in_clk) then
    if cnt_clock = const_divider - 1 then
      adc_clk <= '1';
      rising_clk <= '1';
      cnt_clock <= 0;
    elsif cnt_clock = (const_divider - 1)/2 then
      adc_clk <= '0';
      falling_clk <= '1';
      cnt_clock <= cnt_clock + 1;
    else
      rising_clk <= '0';
      falling_clk <= '0';
      cnt_clock <= cnt_clock + 1;
    end if;
  end if;
end process;
```

output value is read bit by bit and starting with the MSB, written into a shift register. When 8-bits have been read the Value-Enable-Bit `int_val_en` is set to logic '1' and returns to the `wait_state`. After the change this will be reset to logic '0' again.

The ADC outputs a new bit on every falling clock edge of the 100 kHz clock. Each bit is read on the rising edge of the clock with help from the Enable signal from the clock process.

This implementation is used in the same way by all the other state machines so only the individual states and transitions will be described and not the principle itself.

The sending process

A/D conversion is initiated by sending a telegram to the A/D converter. This is performed using a separate process. According to the ADC data sheet (data sheet [3], Figure 20) it reads control signals from the FPGA on rising clock edges. The A/D converter outputs a new bit at rising clock edges. To ensure that the data is stable the implemented state machine reads the value of these bits on the falling clock edge.
The state diagram showing the sending process structure is given in **Figure 4**. The process is implemented as a state machine with three states:

- `wait_state`: Waits for the Channel_Enable signal to start the conversion. When this signal is logic '1' the chip select signal is set to logic '0' and the first bit of the telegram is sent. The state machine changes to the `send_state` state and decodes the remaining bits in the telegram.
- `send_state`: In this state bits are sent one after another on the data line to the A/D converter. The number of sent bits are counted until all the bits have been sent then the state changes to `wait_for_rec_ready`.
- `wait_for_rec_ready`: In this state the wait for the conversion process of the analog voltage is finished. This is done with help of the `int_val_en` signal. When reading out is finished communication with the A/D converter is ended by switching the chip select signal to logic '1'. The machine returns to the output state `wait_state` and the next request can be processed.

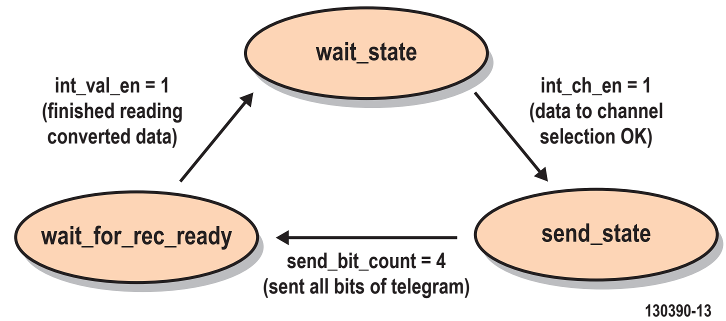


Figure 4.
State diagram of the ADC send processes.

Listing 2

```
type state_type_read is (wait_state, read_state);

signal read_state_machine : state_type_read := wait_state;

process(in_clk) begin
  if rising_edge(in_clk) then
    if rising_clk = '1' then
      case read_state_machine is
      when wait_state =>
        read_bit_count <= 0;
        int_val_en     <= '0';

        if in_sar = '1' then
          read_state_machine <= read_state;
        end if;

      when read_state =>
        if read_bit_count /= 8 then
          int_adval (7 downto 1)   <= int_adval (6 downto 0);
          int_adval (0) <= in_do;
          read_bit_count <= read_bit_count + 1;
        else
          read_state_machine <= wait_state;
          int_val_en <= '1';
        end if;
      end case;
    end if;
  end if;
end process;
```

The pushbutton input module

The process which reacts to pushbutton activity is contained in a separate VHDL module. Its main function is to perform contact debouncing (see article on the board hardware [4]).

To achieve debounce a counter is started when the signal level from the pushbutton input changes state. The counter is used to produce a delay so that the signal level is only valid once this counter has finished counting. A long press would result in the counter timing out several times and registering several presses. To avoid this situation the active pushbutton is polled at every rising clock edge to check if the press has already been registered. The pushbutton input signal state is compared with its state stored when the counter last elapsed. When the two states are the same (long press detected) the counter is reset otherwise it runs until the debounce delay time finishes. Any contact bounce will be finished before the counter reaches the end of its counting period. The signal level is now stored to temporary memory and a short pulse is output.

Listing 3

```
generic (WAIT_40MS: integer := 320100;
         WAIT_4_1MS: integer := 32900;
         WAIT_1_52MS: integer := 12200;
         WAIT_100US: integer := 900;
         WAIT_38US: integer := 400;
         WAIT_450NS: integer := 10);
```

Listing 4

```
when send_data =>
  out_enable <= '1';
  if wait_counter = WAIT_450NS - 1 then
    state <= wait_state;
    wait_counter   <= 0;
  else
    wait_counter <= wait_counter + 1;
  end if;

when wait_state =>
  out_enable <= '0';
  if wait_counter = wait_time - 1 then
    if prev_state = '0' then
      state <= init;
    else
      state <= write_data;
    end if;
    wait_counter <= 0;
  else
    wait_counter <= wait_counter + 1;
  end if;
```

The LCD Module

The LCD has a parallel interface so all control and data information is sent in the form of parallel words. After each word it is necessary to introduce a wait period to allow the LCD board controller to process the information. It is therefore necessary to generate some wait periods. This is achieved with the `generic` command where constants valid in the module are placed. These are defined in the Entity declaration, as in **Listing 3**. The wait period is defined by the integer value which defines the maximum value of the counter clocked at 8 MHz.

The LCD state machine

A state machine with six states is implemented to control the LCD. **Figure 5** shows a simplified state diagram for the LCD. The state machine starts with `start_up`. Firstly there is a 40 ms delay introduced to allow for the LCD to power up. Next is the `init` state to initialize the LCD.

After initialization it automatically jumps to the `wait_for_data` state and stays here until `in_data_en` (an external input) is logic '1'. This indicates that display data is available to be written to the display.

Next it jumps to the `write_data` state, where all the data is written to the display. For every character written the LCD interface requires a '1' of at least 450 ns on its Enable input. This is taken care of after each character is sent out in the `send_data` state. In here the `out_enable` is set followed by a 450 ns wait.

The display requires a processing time of 38 µs after each character is sent to the display. This is generated by using a `wait_state` before the next character is sent.

After each `wait_state` elapses it returns to `write_data` state until there are no more characters left to send to the display.

Listing 4 shows the relevant section of VHDL code which handles `send_data` and its `wait_state`.

GPS

The VHDL description of the GPS module is made up of four individual modules. The top module for GPS control is `gps_control` which contains three sub-modules `gps_serial_parallel`, `gps_checksum` and `gps_parser`.

GPS control (gps_control)

In the top module `gps_control` the other sub-modules referenced above are declared as `components` and linked to the corresponding ports. The process to turn the GPS module off and on is in this module. In addition a short process flashes an LED each time a valid GGA-type sentence is read.

The Conversion Process (gps_serial_parallel)

In the `gps_serial_parallel` module the serial UART protocol data from the GPS module is converted into a one byte wide parallel signal. Using a previously calculated divider constant the communication speed with the GPS module (here we use 4,800 bit/s) can be adapted as necessary.

It is important that the sampling points of the received GPS data stream are synchronized to the data rate. To achieve this, the falling edge of the start bit at the beginning of every byte is detected, the GPS data `gps_data` input signal is shifted into the vector `data_shift` using the internal 8 MHz clock and compared with the bit sequence '1110'.

Once the falling clock edge is detected the following GPS data will be sampled one half of a bit width later i.e. mid-bit, and then shifted into the `int_data` shift register until a complete byte has been received.
The valid data is now stored in `int_data` and written to the parallel data output `out_data` and the Enable-signal set.

Checksum calculation (gps_checksum)

The `gps_checksum` module calculates the checksum on all the transmitted data bits and compares it with the checksum value sent from the GPS module. This ensures that there are no errors in the received sentence. When an error is detected the corrupted sentence is discarded.
A state machine with five states is used to read-in, calculate, validate and them output the result:

- `reset`: All of the signals used for these calculations are first reset to zero. When valid data from the serial/parallel converter (`out_data_enable` = 1) is available the state of `zeichen_in` changes, as soon as a '$' symbol is detected in the data. This symbol is the GPS sentence start character.
- `zeichen_in`: This detects where the checksum begins in the received sentence and changes to the `checksum_in_1` state. An 'if' condition is used to detect an asterisk which marks the end of the sentence data. The two characters following the asterisk are the two-byte sentence checksum.
- `checksum_in_1`: This reads in the first checksum character. This is XOR'ed with the sum of the input characters. When it is not valid the signal `int_checksum_err` will be assigned logic '1'.
- `checksum_in_2`: The second checksum character is read in here. Otherwise identical to `checksum_in_1`.
- `output`: When the checksum is valid then the signal `int_checksum_ok` is given the value logic '1' otherwise it has the value '0'.

Reading GPS data (gps_parser)

The `gps_parser` module filters out the relevant information from the received GPS data stream and prepares it for further processing. The GPS module sends all its data sentences sequentially

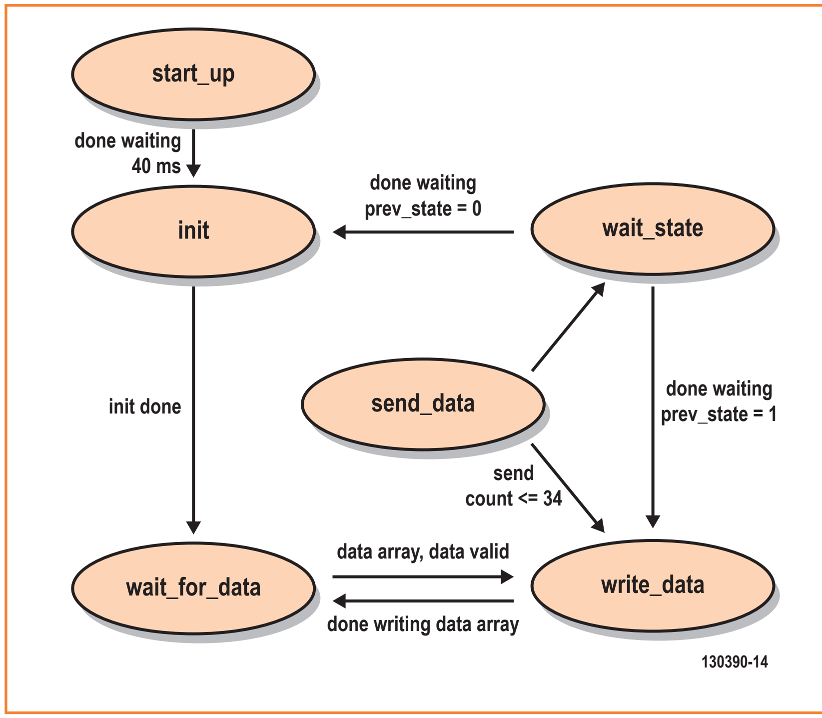


Figure 5.
State diagram of the LC display.

according to the NMEA protocol. It is necessary to identify the sentence of interest (for our purposes the GGA sentence containing positional fix information) and recover it from the data stream. A '$' character identifies the start of every new data sentence. A state machine checks when this occurs. Following this character is the 'GPGGA' sequence which is the preamble to the GPS data of interest to us. There is a state for each data sentence of interest in which the data is read in. Each data field in the sentence is separated by a comma and this is used to change the state of the machine. One after another all the data is read in and sent to the corresponding output.

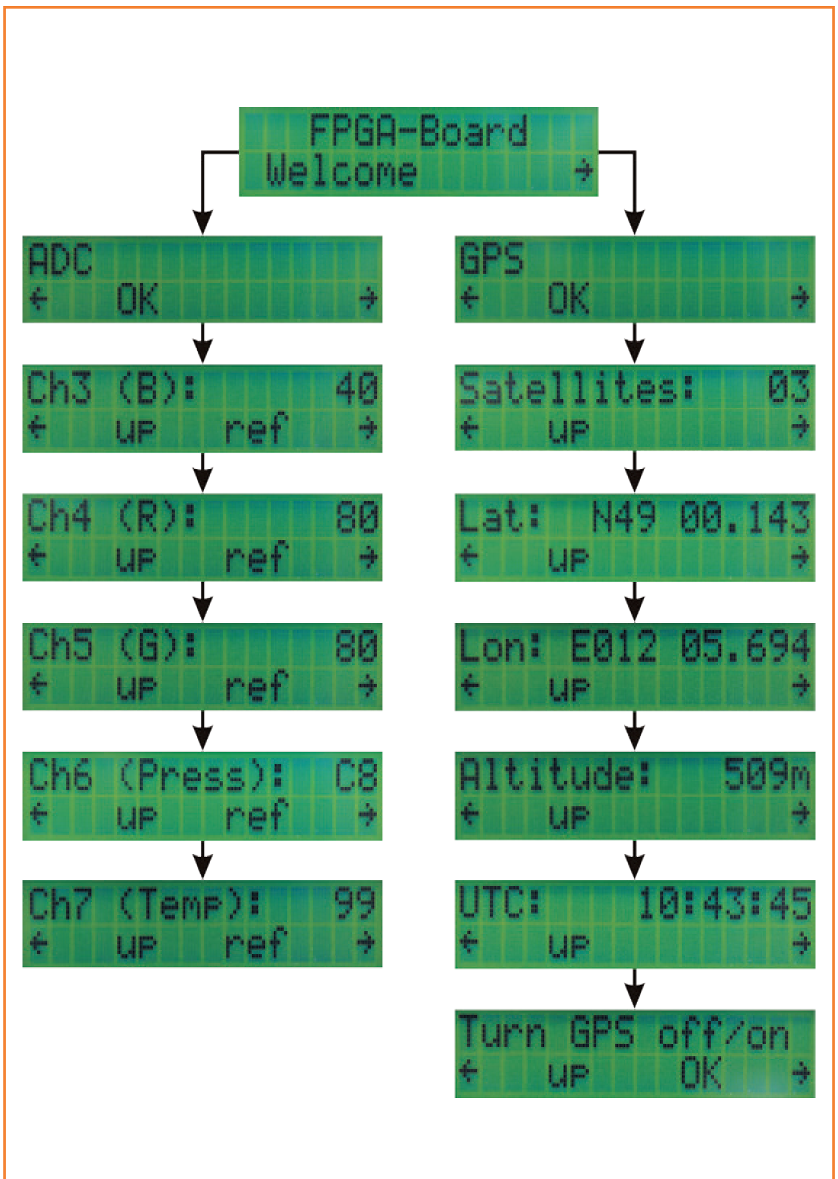


Figure 6.
The menu options.

Menu control in VHDL

A menu has been implemented on the display to allow user control of the GPS module and A/D converter. Pushbuttons under the display allow intuitive interaction with displayed menu options. **Figure 6** shows the menu layout. After the start screen there is an option to select sub menus 'GPS' or 'ADC'.

The GPS sub-menu firstly gives you the option to turn it off or on. Other pages give you the option to view additional information such as your current longitude and latitude. The command 'up' returns you to the next level up in the menu structure.

Selecting 'ADC' from the menu allows you to select a channel of the A/D converter. The measured values are displayed on a page in the sub-menu. Pressing 'ref' (refresh) causes the ADC to make another measurement of the displayed channel and update the display with the new value.

The GPS menu

The menu control is also built with a state machine and can be easily restructured (by changing the state transition diagram) or expanded (add new states). The menu structure and associated state machines are described using the GPS menu as an example.

Each page in the menu has a corresponding state of the state machine for control of the menu. The current state is stored in the signal `state`. The machine starts in the `state_init` state and then changes to the `state_welcome` state.

Now the welcome screen is shown. A press of the pushbutton on the right changes to the `state_gps` state. This state builds the highest level of the GPS menu. In the lower line of the display are arrows pointing to the left and right. Pressing the button beneath the arrow changes the state to `state_adc` and now the A/D converter menu options are displayed.

Staying in the GPS menu you can press OK to get to the first GPS sub-menu, here you have the option to switch the GPS module on and off (`state_GPS_toggle`). When powering down the GPS module it is important to observe the correct power-down sequence to avoid any possible internal memory data corruption. On one level with the state to switch on and off there are display options for GPS data such as longitude and latitude which can be selected using the left and right pushbuttons.

As an example we can show how the longitude is displayed on the LCD. In the `state_longitude` state it will (automatically when valid data is available) assign to the vector elements of `out_lcd_line` the display data elements. In the VHDL description, for example, the data element `in_lon_pre` is assigned to the vector element `out_lcd_line1(5)`. At the fifth position on the first display line is the value of `in_lon_pre` which in this case will be either the letter 'E' or 'W' i.e. east or west of the central meridian. In accordance to this principle each position of the display will be assigned the character to be displayed. After this process `refresh_lcd` refreshes the display and displays the characters.
The principles of state changing and display of data described above also operate in the same way for the other menu pages.

To sum up

As an example project we have demonstrated a menu driven control of the FPGA expansion board. All the Modules consist of systematically implemented state machines. This project used up 40 % of the gates and look up tables in the FPGA. There is still enough in reserve for additional applications. It would be fairly easy, for example to add a function to convert the A/D output values into a temperature reading or a voltage level.

(130390)

Internet Links

[1] Multi I/O for FPGA Development Board, Part 1: www.elektor-magazine.com/130148

[2] Xilinx ISE: www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools/v2012_4---14_5.html

[3] Data sheet for the A/D converter: www.ti.com/lit/ds/snas531b/snas531b.pdf

[4] www.elektor.com/130390

Advertisement

555 Class-D Audio Amplifier

A novel use for an old-timer

By **Frederik Crevits**
(Belgium)

The celebrated '555' IC was originally developed as a timer device, but over the years this golden oldie has been used in all sorts of other applications, including some never imagined by its inventors. This article describes a small class-D stereo audio amplifier built around the oscillator and modulation sections of a 555. The simple design makes it easy to build the circuit yourself.

Thousands of different applications can be implemented using the popular 555 timer IC. Here this IC is used as the basis for a simple class-D audio amplifier that operates without overall negative feedback. Only standard components are used in the circuit. In addition to making the circuit easy to build yourself, this makes it a good learning project. Furthermore, this project shows that class-D does not have to be exotic or difficult.

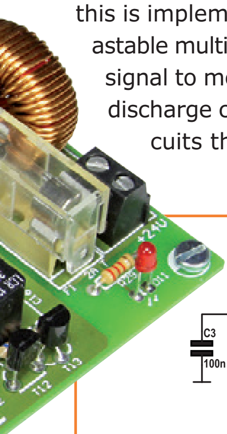
The key component of this circuit is the NE555 timer IC. Originally developed in 1971 (ancient times in the Integrated Circuit world), it is still enormously popular due to its simplicity, reliability and versatility.

A pair of MOSFETs (type IRF530) driven by a voltage level shifter enable the circuit to deliver sufficient output power. To avoid making the circuit unnecessarily complicated, we opted for a single supply voltage, which means that an output capacitor is required.

Schematic

First a comment about the schematic diagram in **Figure 1**, which shows a complete stereo amplifier. To avoid having to mention a whole raft of component numbers in the following circuit description, we limit ourselves to the components in the upper channel.

In a class-D audio amplifier, the analog audio signal is converted into a pulsewidth modulated signal that drives the output transistors. Here this is implemented by employing the 555 as an astable multivibrator and using the analog audio signal to modulate the voltage for the charge/discharge capacitor (C2). In standard 555 circuits this capacitor is usually charged by



Measured Performance @ Elektor Audio labs

• Input sensitivity:	580 mV (*THD+N* = 1%)
	830 mV (*THD+N* = 10%)
• Input impedance:	11 kΩ
• Continuous output power:	6.3 W (*THD+N* = 1%)
	10.8 W (*THD+N* = 10%)
• Power bandwidth:	11 Hz – 37.5 kHz (–3 dB)
	21.5 Hz – 31 kHz (–1 dB)
• Signal to noise ratio:	69 dB (1 W / 8 Ω; *B* = 22 Hz – 22 kHz)
• Total harmonic distortion + noise:	0.23% (1 kHz; 1 W / 8 Ω)
• Channel separation:	42 dB (100 Hz, P_{max})
	54 dB (1 kHz, P_{max})
	60 dB (20 kHz, P_{max})
• Current consumption:	0.8 A at 2 x 6.3 W / 8 Ω

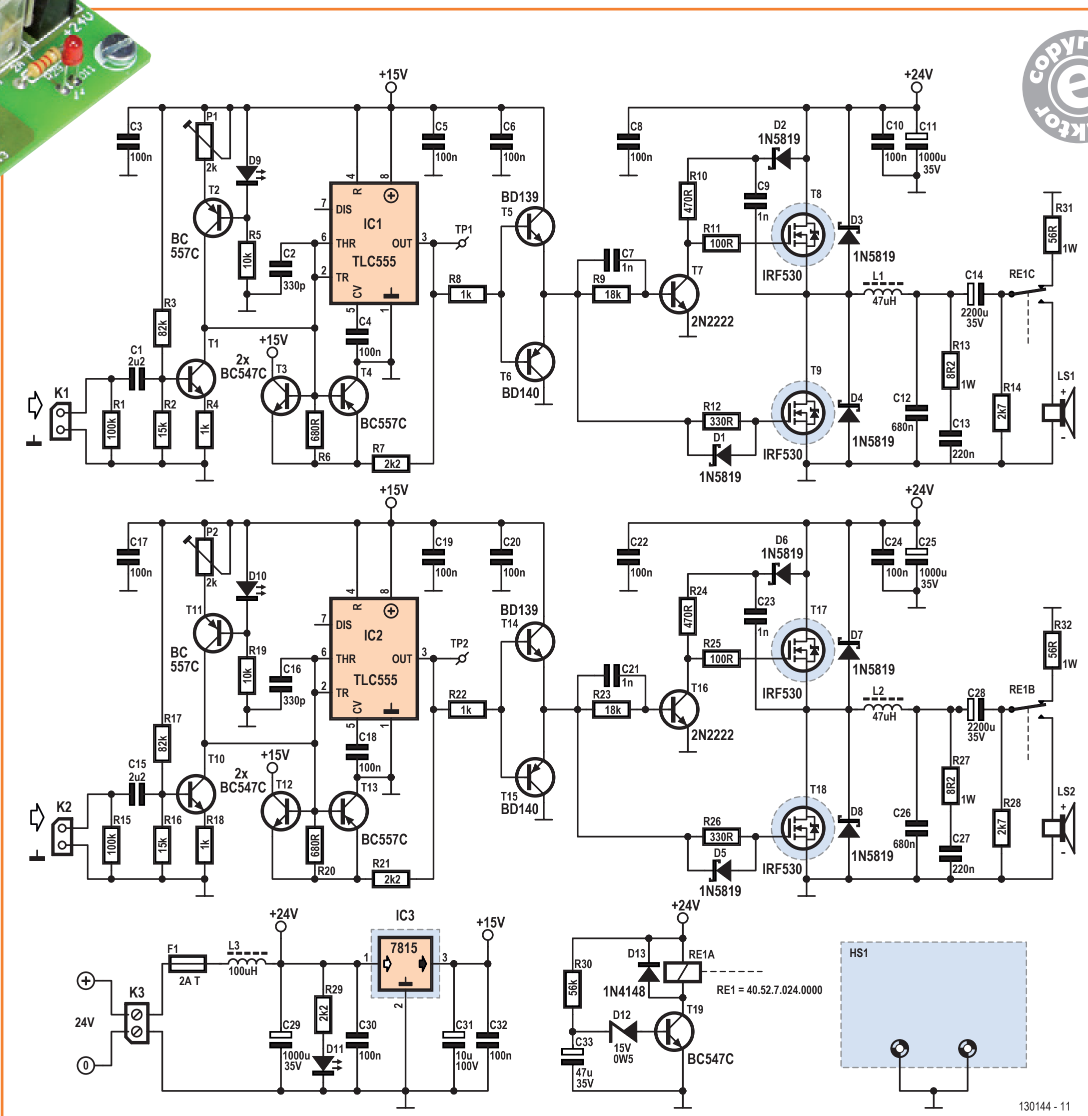


Figure 1.
Full stereo version of the class-D amplifier based on the 555 IC. A simple switch-on delay circuit is included.

A Nearly Perfect V-to-I Converter

Although the arrangement with T3 and T4 dramatically improves the linearity, it is somewhat less than perfect. The current through the capacitor is the sum of the current determined by the base–emitter voltage over R6 and the base current of T3 or T4. Since the collector current of T3 or T4 depends on the current through R7, the base–emitter voltage (and therefore the base current) also depends on this current. This relationship is not linear, and another factor is that the two transistors are not perfectly complementary, so they have slightly different gain curves.

Component List

Resistors

R1,R15 = 100kΩ
R2,R16 = 15kΩ
R3,R17 = 82kΩ
R4,R8,R18,R22 = 1kΩ
R5,R19 = 10kΩ
R6,R20 = 680Ω
R7,R21,R29 = 2.2kΩ
R9,R23 = 18kΩ
R10,R24 = 470Ω
R11,R25 = 100Ω
R12,R26 = 330Ω
R13,R27 = 8.2Ω, 1W
R14,R28 = 2.7kΩ
R30 = 56kΩ
R31,R32 = 56Ω, 1 W
P1,P2 = 2kΩ multiturn preset (Vishay Sfernice T93YB202KT20)

Capacitors

C1,C15 = 2.2µF 50V, 5mm or7.5.mm pitch (e.g. Panasonic ECQV1H225JL)
C2,C16 = 330pF 1%, polystyrene, 7,18mm pitch (e.g. LCR Components EXFS/HR 330PF ±1%)
C3–C6,C8,C10,C17–C20,C22,C24,C30,C32 = 100nF, X7R, 0.2’’ pitch
C7,C9,C21,C23 = 1nF MKT, 5mm pitch
C11,C25,C29 = 1000µF 35 V, radial, 12.5mm dia. 5mm pitch (e.g. Rubycon 35YXF1000MEFC12.5X25)
C12,C26 = 680nF polypropylene, 15mm pitch (e.g. Panasonic ECWF2684JAQ)
C13,C27 = 220nF metallized film (MKT), 5mm pitch
C14,C28, = 2200µF 35V, radial, 18mm diam. 5mm or 7.5mm pitch (e.g. Panasonic EEUTP1V222)
C31 = 10µF 100V, radial, 6.3mm diam., 2.5mm pitch
C33 = 47µF 35V, radial, 8.5mm max. diam., 2.5mm pitch

Inductors

L1,L2 = 47µH, 21mΩ/8.5A, pot core type (Murata Power Solutions 1447385C)
L3 = 100µH, 35mΩ/5A ring core type (Würth Elektronik 7447070)

Semiconductors

D1–D8 = 1N5819
D9,D10 = LED, red, 2x5mm rectangular
D11 = LED, red, 3mm
D12 = 15V/0.5W zener diode
D13 = 1N4148
T1,T3,T10,T12,T19 = BC547C
T2,T4,T11,T13 = BC557C
T5,T14 = BD139
T6,T15 = BD140
T7,T16 = 2N2222
T8,T9,T17,T18 = IRF530
IC1,IC2 = TLC555CP
IC3 = 7815

Miscellaneous

K1,K2 = 2-pin pinheader, 0.1’’ pitch
K3,LS1,LS2 = 2-way PCB screw terminal block, 0.2’’ pitch
RE1 = Relay, 24V, 1200Ω, 8A, DPDT-CO (e.g. Finder 40.52.7.024.0000)
F1 = glass fuse, 2A(T), slow-blow, with PCB mount holder and cover
TP1,TP2 = 1 pin of pinheader
HS1 = heat sink for MOSFETs, aluminum plate 130 x 50 mm (5.1 x 2 inch), 1-2mm thick
4 x isolating washers for TO-220 case (e.g. Bergquist SIL-PAD K-10, .006’’, TO-220)
4 x isolating bush, 3mm
Heat sink for IC3, 30K/W (e.g. Fischer Elektronik SK 12 SA 32)
Switch-mode power supply, sec. 24V, 2.5A min.
PCB # 130144-1 [1]

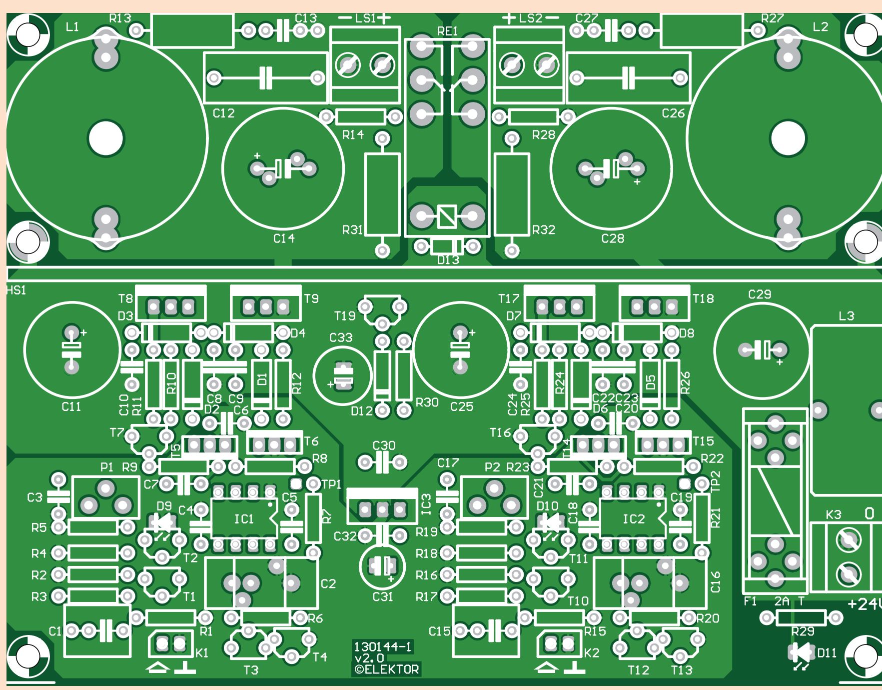


Figure 2. The PCB layout is uncrowded, with room for an aluminum plate heat sink for the MOSFETs—although the dissipation is low with the class-D architecture.

a constant voltage. This leads to nonlinearity because the charge and discharge curves are always exponential. That's not an especially good basis for building an amplifier.

To eliminate this problem, here we charge the capacitor with a constant current instead of a constant voltage. This is handled by a current source built around T2 and a voltage to current converter built around T3 and T4. This results in a fairly linear triangular voltage waveform on C3, and the ratio between the PWM output signal of the 555 and the input signal is close to linear. The input signal on connector K1 affects the charge/ discharge time of the capacitor via T1, and in this way modulates the output signal. The switching frequency is approximately 250 kHz.

A buffer stage consisting of a BD139 and a BD140 complementary pair (T5 and T6) at the output of IC1 prevents excessive loading by the downstream circuitry.

The resulting PWM signal drives a push-pull output stage with two MOSFETs (T8 and T9), which are able to deliver enough current to drive a 4-Ω or 8-Ω loudspeaker. It is essential to ensure that T8 and T9 never conduct at the same time, since that would short out the supply voltage. However, the dead time (the time when neither of the MOSFETs is conducting) must be kept as short as possible in order to minimize harmonic distortion.

This creates a dilemma. With the 15 V supply voltage for the 555, it is not possible to deliver very much power to a 4-Ω loudspeaker. We solve this problem by connecting the MOSFETs to a higher supply voltage—in this case 24 V.

Since the high-side MOSFET (T8) always sees the low-side MOSFET (T9) as a load, the voltage U_{GS} from the driver stage will never be high enough to drive T8 fully on. As a result, the output voltage will never rise above 15 V and the rest of the power will be dissipated by the MOSFETs as heat. This is not how class-D is supposed to work. The circuitry around T7, which bootstraps the gate of T8, remedies this situation. When the output of the T5/T6 buffer stage is high (15 V), T7 is driven into conduction and cuts off T8. Capacitor C9 is then charged through D2 and T9 (which is conducting because its gate is connected directly to the buffer stage) to a level close to 24 V. When the output of the T5/T6 buffer stage goes low (0 V), T7 and T9 are cut off. This puts the 24 V supply voltage in series with the voltage on C9, resulting in a level of approximately 45 V relative to ground. This voltage is sufficient to drive T8 fully on, so the circuit works the way it should. At the output there is a PWM signal with an amplitude of 24 V, while T8 and T9 remain nice and cool.

The network D1/R12 is included to control the dead time. It causes the turn-on and turn-off times of T9 to be different because the gate capacitance is charged through R11 but discharged through D1, which is much faster.

An LC filter is placed at the output to suppress the 250 kHz square-wave signal. The output from the filter is a clean audio signal that can be feed to the loudspeaker through a cable. The filter is dimensioned to have its 3-dB corner frequency at approximately 37 kHz. The network R13/C13 prevents undesirable oscillation when no loudspeaker is connected.

The power supply section is very simple. A 7815 provides a regulated 15 V supply voltage for the 555 stage. Due to the simple design, the amplifier does not have any real protection circuit, but the circuitry around T19 provides a switch-on delay to prevent audible switch-on pops.

A good choice for powering the amplifier is a low-cost switch-mode power supply with a regulated 24 V output voltage. Such power supplies are available from electronics distributors at reasonable prices. RFI choke L3 is included in the power supply section to block any noise from the switch-mode power supply, so that it does not reach the amplifier stages.

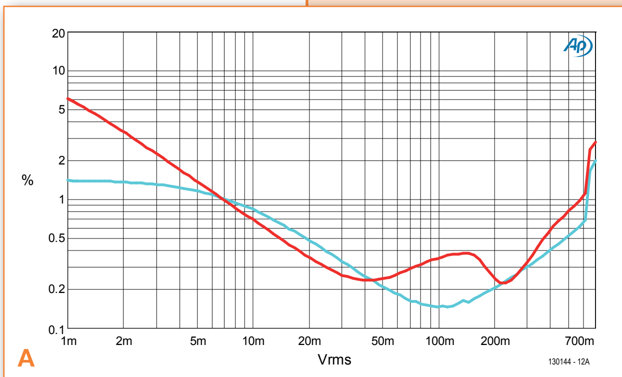
Construction

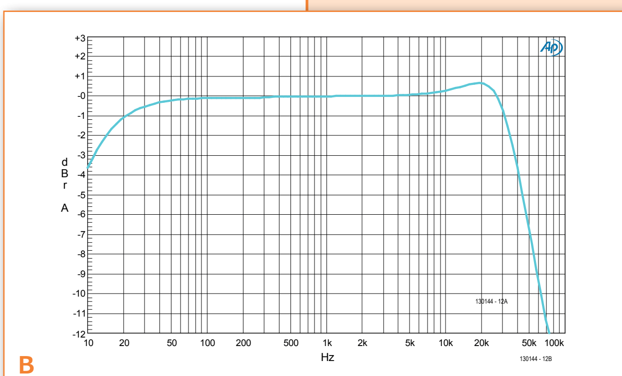
Figure 2 shows the circuit board layout designed for this amplifier. As already mentioned, this is a stereo version that only requires the connection of an external power source.

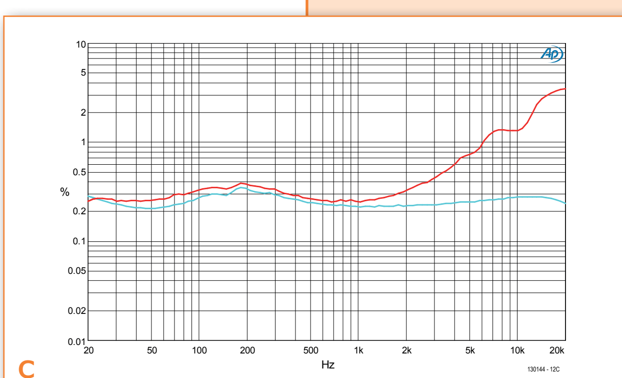
All components are leaded (through-hole) types to make assembly easy. There is room in the middle to fit a small aluminum plate that provides extra cooling for the output transistors. This is not absolutely necessary, since the transistors remain fairly cool.

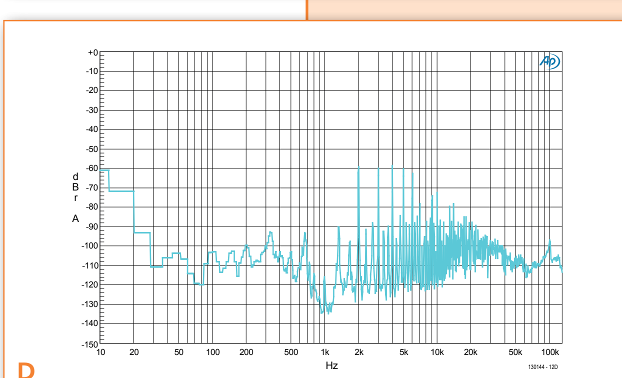
A few details deserve mention. You should use the best possible components for the frequen-

A Few Measurements









Due to the simple design and the absence of overall negative feedback, you shouldn't count on especially low distortion figures—but all things considered, the results are much better than expected. During the measurement session, it was interesting to see that the distortion components from the different stages of the amplifier partially cancel each other at some output amplitude levels.

Plot A shows this effect (THD versus input level, 1 kHz, *THD+N* with *B* = 22 kHz). The *blue* curve shows the distortion at the output of the 555 (pin 3) after filtering out the oscillator frequency. The *red* curve shows the distortion at the output of the amplifier. At input levels between 7 mV and 40 mV, the distortion at the amplifier output is lower than the distortion at the 555 output. This may be due to the dead time in the output stage (similar to the distortion of a standard class-B output stage), or it may be due to the output filter.

Plot B shows the frequency characteristic of the amplifier at 1 W into 8 Ω. The lower corner frequency is 11 Hz, and at the other end the –3 dB point is at 38 kHz. There is a small bump (0.66 dB) in the vicinity of 19 kHz. The slight overshoot can be reduced by decreasing the value of filter capacitors C12 and C26 to 390 nF, but this also reduces the attenuation of the modulation frequency by some 4 dB.

Plot C shows the total harmonic distortion plus noise versus frequency with an 8-Ω load. The red curve shows the distortion ahead of the output filter of the amplifier, while the blue curve shows the distortion at the output (after the filter). It is evident that the filter does its job properly and effectively suppresses the intermodulation products in the higher frequency region. Both measurements were made using the class-D measurement filter described in Elektor in 2011 [3], with the bandwidth of the analyzer limited to 80 kHz.

Plot D shows the Fourier spectrum of a 1 kHz signal (1 W / 8 Ω). The five harmonics responsible for the majority of the distortion (*THD+N* = 0.23%) are at a level of –60 dB. Intermodulation products are also visible with a spacing of 326 Hz. They result from the difference between the clock frequencies of the two modulators. However, these products are under –85 dB. They disappear if one of the channels is switched off. To avoid these intermodulation products, the clock frequencies would have to be at least 40 kHz apart.

cy-determining capacitors C2 and C16. Polypropylene and silver-mica are both good choices. To keep the temperature coefficients of the current sources T2 and T11 as low as possible, the (rectangular) LEDs D9 and D10 should be fitted in close contact with the corresponding transistors (T2 and T11). They are therefore located close together on the PCB. Murata pot cores and polypropylene capacitors are recommended for the output filters (L1/C12 and L2/C26).

In the prototype we used ceramic insulators for the MOSFETs, but you can also use insulators made from other materials (mica, Kapton, etc.) because the power dissipation is low. First mount the MOSFETs firmly in the right positions on the aluminum plate, and then bend the leads so they fit precisely in the corresponding holes in the PCB without exerting any lateral force on the MOSFETs. The aluminum plate can be mounted on the PCB using two small brackets. This also ensures that the plate is connected to circuit ground. Mount the heat sink on the board before soldering the leads of the MOSFETs to the board. Fit a small heat sink on the 15 V voltage regulator (IC3). A Fischer type SK 12 SA 32 (30 K/W) is suitable, but you can also use a small piece of aluminum. Make sure that the heat sink is not jammed against C33.

Adjustment

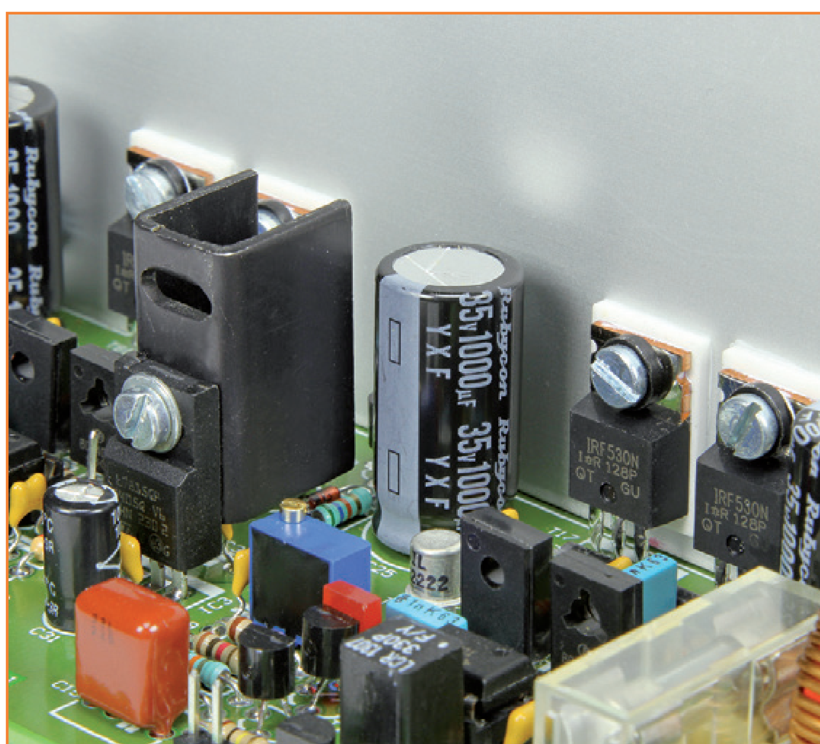
Short the input to ground and then adjust multiturn potentiometer P1 to obtain the best possible symmetry of the output signal. The voltage measured at the output (behind L1) should then be exactly half the supply voltage, i.e. 12 V.

That's it—now the amplifier is ready to use. Connect it to a pair of loudspeakers and an audio signal source, sit back and enjoy the pleasant sound of this amplifier, which in many ways resembles the sound of a tube amplifier.

(130144-I)

Figure 3. This side view clearly shows the position of the aluminum plate.

Figure 4. In the prototype the MOSFETs were mounted on the aluminum plate heat sink using screws with 3-mm insulating washers and ceramic insulators.



Internet Links

[1] www.elektor-magazine.com/130144

[2] www.elektor-labs.com/130144

[3] www.elektor-magazine.com/100540

Mixing Electronics and Mechanics with FLOWCODE 6

Electronic system design software

as explained by the developer

By **Paul Newill**
(Applications Engineer,
Matrix Multimedia Ltd., UK)

Flowcode 6 is the latest version of the flowcharting software from Matrix Ltd. As well as improved functionality over the previous versions, v6 includes many new features. This article provides an overview of one of the key new features, 3D modeling and electro-mechanical system simulation.

Drag and drop programming

Flowcode is a graphical based programming language, where flowcharts are the method of creating the program structure rather than the use of a text based programming language. Not only does this provide a method of programming which is free of syntax errors, it also allows for a simple drag and drop of icons. Flowchart functions are configured by adjusting their parameter properties. For example, within the 'properties panel' of a 'while loop' the user can choose the loop conditions, such as the number of loop executions or the break condition of the loop, such as when a variable reaches a certain value. Within Flowcode v6 we have also developed a large library of standard components, utilized through component macros, such as LEDs, switches, LCD screens, various communications modules and more. Component macros vastly simplify the programming of otherwise complex systems. For example, by dragging a single component macro into a flowchart the user can initialize an LCD display—typically a whole section of C or assembler code would be required to achieve this.

The first 'stepper' steps

One of the included components available with Flowcode v6 is the stepper motor. A stepper motor is typically used in applications such as an XY plotter or a 3D printer due to its ability to achieve very precise movements, with the rotational step angle of the motor being typically around 1.3 to 10 degrees. In this article we will proceed to progress through examples of increasing complexity, but observe the simplicity with which this can be achieved.
In the first example I will demonstrate the simplicity with which the stepper motor can be added

to the design. In Flowcode there are two panels; the Dashboard and the System Panel. The Dashboard provides a 2D view, so objects suited to this are placed here. Objects such as LEDs, switches and keypads are typically placed on the Dashboard. The System Panel offers a 3D view, and is typically more suited to objects such as motors, servos or solenoid valves. Once the stepper motor was placed on the system panel I added a primitive shape to the panel. The shape of the primitive was then modified to represent a long bar which was placed on the shaft of the motor.
The flowchart used to rotate the bar can be seen in **Figure 1**, where it can be seen that only four functions are required. The first component macro enables the stepper motor. The flowchart then proceeds to the next command—a loop which is configured to execute 50 times. The stepper motor was configured to have a total of 100 steps per revolution. Therefore, after 50 iterations the motor will have rotated through 180°. Within the loop there are the final two commands. The first is to increment the step of the motor, while the second introduces a small delay. This delay is used to slow the simulation down and allow the user to see the animation. If the program was to be downloaded to a chip, this delay would be essential to stop the motor stalling and vibrating as it tried to spin too fast.
Figure 2a shows the 3D component for the stepper motor. **Figures 2b-d** show the bar attached to the shaft and the simulation at the start, midpoint and end where it can be seen that the motor rotated 180° as desired.

Linear movements

The second example will demonstrate how Flowcode can be used to convert a rotational movement from the stepper motor into a linear movement. As previously mentioned, stepper motors are used in applications such as XY plotters or 3D printers. In these applications, the rotational movement from a stepper motor must be converted from rotational to linear in order to move the *XYZ* axes. This can be achieved by attaching a threaded bar to the motor shaft, and a threaded nut is fixed to the axis. As the threaded bar spins, the axis platform will move accordingly. In this example, the stepper motor properties were adjusted to simulate a linear movement each time the *Increment Step* component macro was called from the main program. This linear movement was then assigned to the unique handle of a shape within the 3D design environment. The program is too large to include a screenshot within this article, but the file is included within the Flowcode example files to allow you to look over the flowchart yourself. **Figure 3** shows the 3D model constructed to demonstrate two axes of linear movement. As the stepper motors rotate, the gold objects move along the rails in a linear motion. This example works by utilizing a flag within Flowcode to determine the direction of travel. If the flag is set to 0, the motor will step and rotate the shaft anti-clockwise. If the flag is set to 1, the direction of travel will be the opposite. The difficulty encountered in this program was determining when the moving object had reached the limit of the axes and the simulation

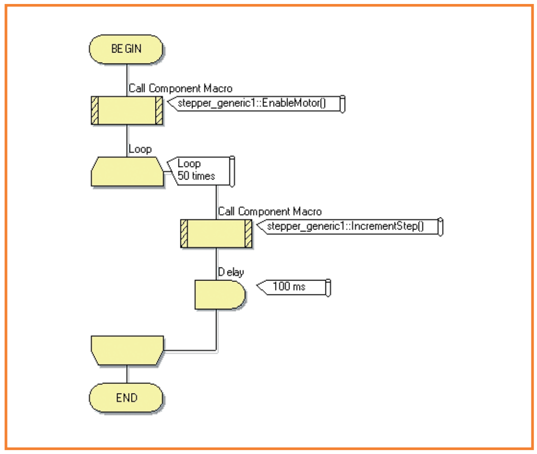


Figure 1. Example 1: flowchart for stepper motor simulation.

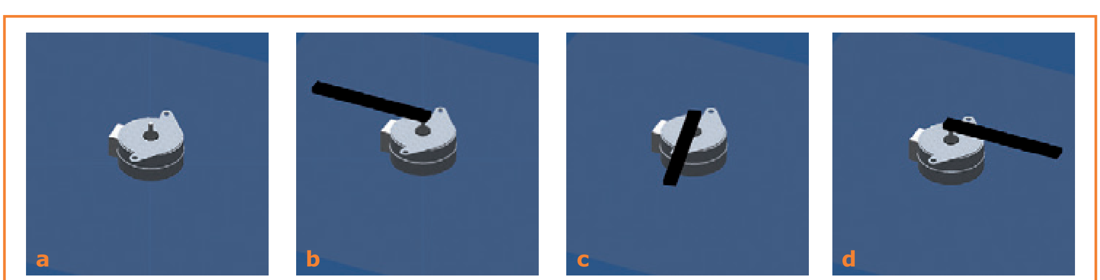


Figure 2. Example 1: stepper Motor 3D model and simulation screenshots (a-d).

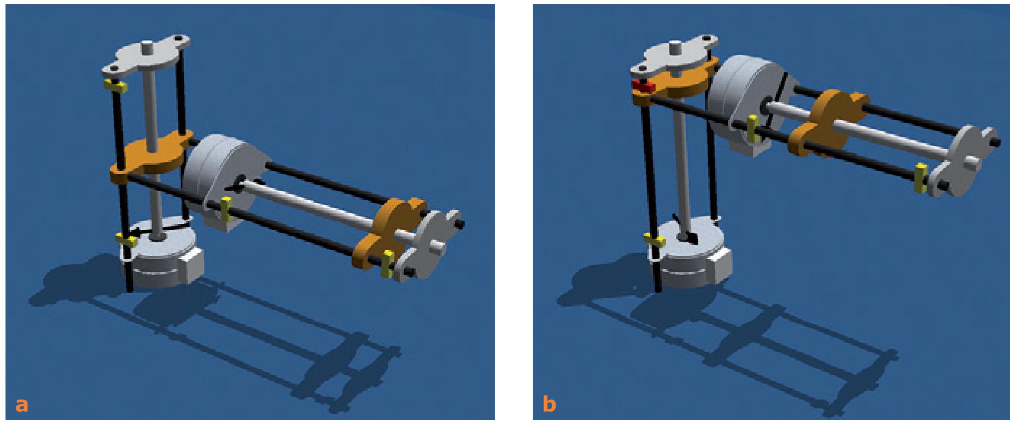


Figure 3.
Example 2: simulating linear movement from stepper motors within Flowcode 6.

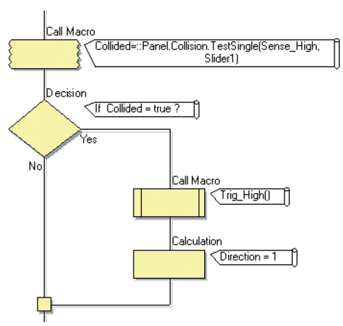


Figure 4.
Example 2: collision detection macro.

must subsequently reverse the direction of travel. Since these examples are simulation only, we implemented commands known in Flowcode as "SIM Command". A SIM command is one that will only execute within Flowcode and if the design was compiled to a microcontroller the SIM commands would not be sent. In this example, a push-to-make switch component could be configured within Flowcode to operate in the same way, however, SIM commands were used here to demonstrate their simplicity and flexibility as there will be times when you only require SIM commands. There are many more SIM commands. In **Figure 3b**, it can be seen that the top yellow marker has changed to red. This has occurred due to Flowcode detecting a collision between two objects, determined by the section of code shown in **Figure 4**. When a collision is detected, the 'yes' branch of the decision statement executes. In this branch, the first icon shows that a macro named 'Trig_High' executes. Within this macro the author has created a flowchart that temporarily changes the color of the yellow marker to red, to notify the user that the collision has occurred. The second command within this branch then changes the value of the direction flag to 1, reversing the direction of travel.

3D models

The final example provided within this article demonstrates a progressive step from those previously detailed. Here an XY plotter was simulated and also built from actual parts, to demonstrate the complexity of systems which can be created within Flowcode v6. In this example the 3D model of the plotter was first created in SolidWorks—**Figure 5**. The model was then separated into five sections and each was exported from SolidWorks as an STL file. These five files can then be simply imported into the Flowcode environment by drag and drop onto the system panel once imported, each of the five 3D models were configured as required.

The specification of the plotter was that the program was required to read in a set of XYZ coordinates from an array, and draw a shape determined by the coordinates. The given coordinates saved in the array were in the unit of millimeters, therefore in order to reach the coordinate position the distance between the current point and the new coordinates must be calculated. This value was then scaled by the thread pitch of the threaded bar, and the step size of the motor. In our plotter the stepper motors have 200 steps per 360° revolution. The pitch of the threaded bar was 1.25 mm, and therefore each motor must through rotate 160 steps to achieve 1mm

of linear movement. For diagonal movement a function was also created to allow both motors to rapidly alternate movement. The gradient of the diagonal line was dependent on the value of a scaling factor, calculated by the Bresenham's line algorithm.
The final coordinate to consider was the Z axis. Despite the plotter being only 2D, the Z axis was required to lift and lower the pen in order to draw the required shapes. This was performed with a flag; when the flag was set to 1 the pen would lower and allow drawing and when the coordinate value was a 0 the pen would raise up. A solenoid valve was used to raise and lower the pen as required.

Use 'GetCoords' in the real world

For the simulation the XY plotter requires several of the 3D models to be combined using the *group* function. This is particularly useful for systems where individual parts move, but are linked, like in the x-axis where both the pen and pen holder must move together as the x-axis moves. However, they must also be able to move independently to allow the solenoid valve to lift and drop the pen. Simulation commands are used to provide the ability to simulate, and therefore mirror, the movement of the pen around the drawing area in the XY plotter machine. This movement of axes is performed by linking the 3D objects within the System Panel to each stepper motor, and achieved by configuring the properties of each stepper motor. However, this still does not provide the ability to draw a shape within the simulation which mirrors that which the hardware will draw. For this we need to implement additional SIM commands. Here, the command GetCoords is used to regularly check the XYZ position of the 3D object representing the pen. By knowing the position of the pen, a second SIM command can be used to draw a circle point at the exact XY location. Doing this for each incremental step of the motor produces a pseudo line, constructed from a series of dots. Since this is a complex system, the design utilizes more than 20 flowcharts, as well as the use of hardware and software interrupts and many simulation commands which allow the same flowcharts to be used for both simulation and programming to the microcontroller.

From software to hardware

The hardware of the XY plotter requires a microcontroller board, here the ECIO40 which is attached to the EB061 to provide break out ports A-E. PORTA is connected to a motor board, from which the subsequent stepper motor is connected to the X-axis to provide horizontal movement. The second motor board for the Y axis is connected to PORTD. PORTB is connected to a numerical keypad, which provides a user interface. This also allows manual movement of the X

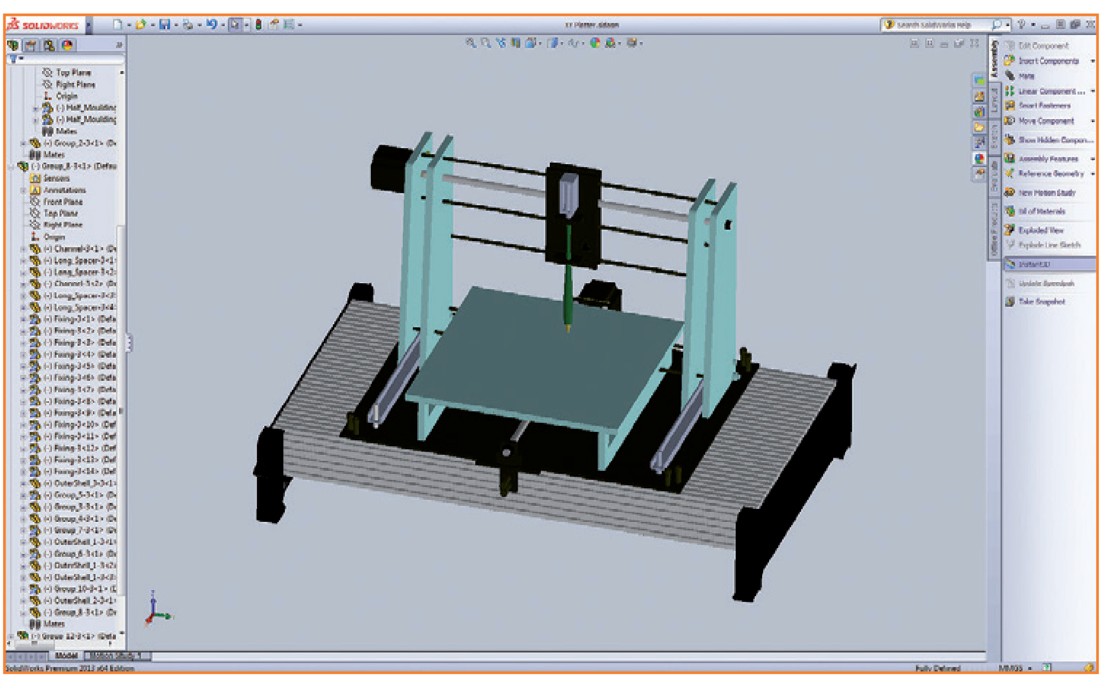


Figure 5.
Example 3: XY Plotter in SolidWorks.

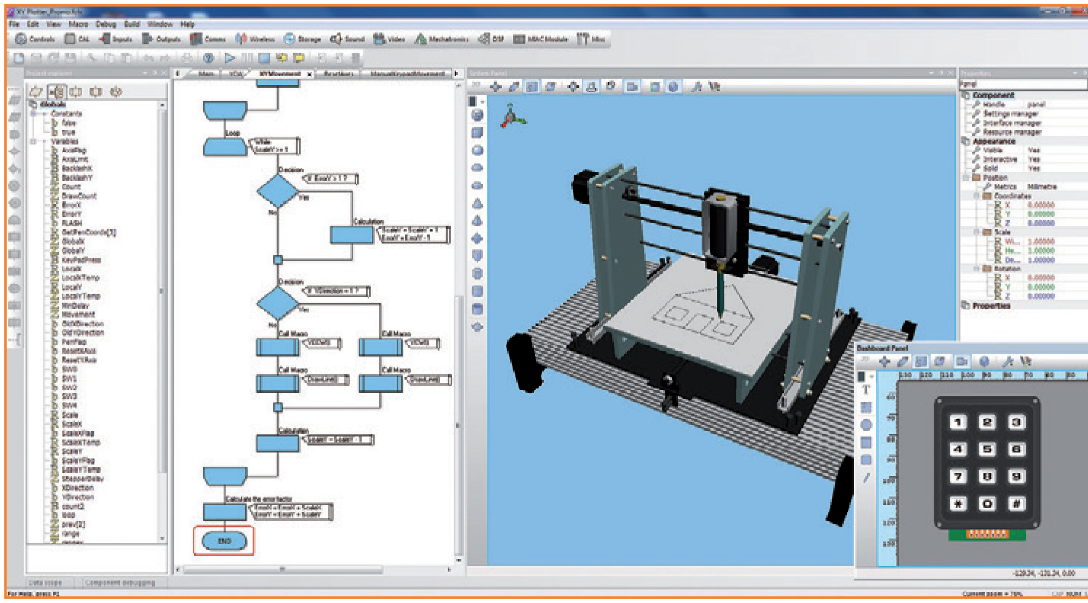


Figure 6.
Example 3: 3D model of XY plotter within Flowcode v6.

and Y axes if and when required. PORTC is connected to a terminal Eblock board, which allows the axis limit switches to be connected. These are connected to the microcontroller as hardware interrupts, which when switched will immediately halt the plotter. These are required if the coordinates provided exceed the drawing area of the plotter, which could cause damage to hardware. The final Eblock is a power board used to provide the required 7.5 volts for the solenoid valve, which is used to lift and drop the pen.

To conclude

The following figures show both the simulation of the XY plotter within Flowcode, and the constructed hardware. In **Figure 6** we can see that the simulation of the XY plotter is capable of drawing a shape, constructed purely from XY coordinates. To conclude, **Figure 7** pictures the electro-mechanical hardware of the XY plotter.

(130391)

Additional Information

Elektor International Media is official distributor of Flowcode 6 and E-blocks products. More articles and sales information can be found at www.elektor.com/flowcode

Follow Matrix on Twitter: @MatrixFlowcode
Check out the Matrix blog: www.matrixltd.com/blog

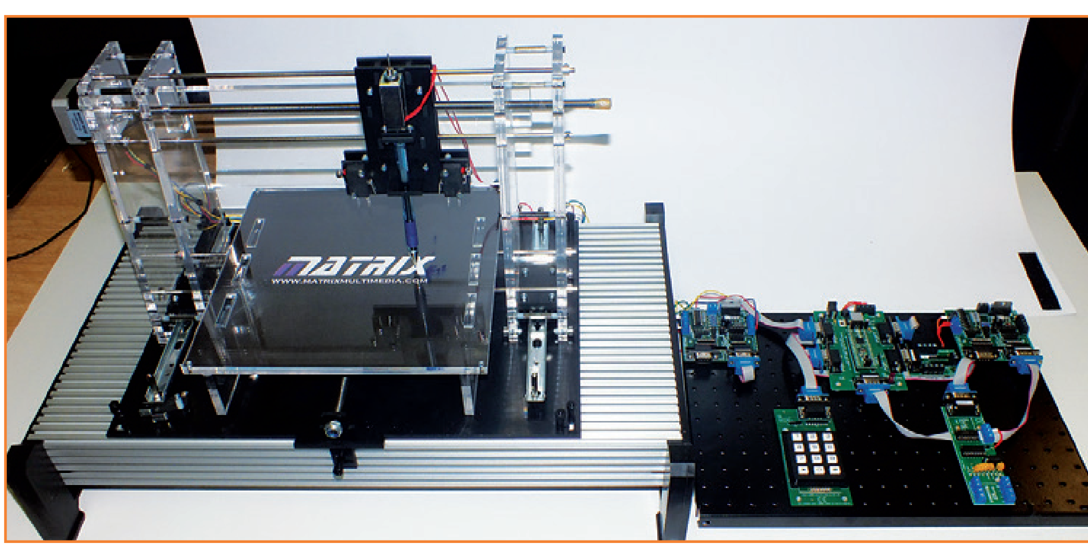


Figure 7.
Example 3: XY Plotter model.

He should have used a LabSat!

GPS simulator - Record once, replay endlessly

- Recreate real world conditions
- Single or dual channel
- GPS, GLONASS, Galileo, BeiDou, QZSS and SBAS
- Signal simulation software available
- Free library of worldwide recordings

LabSat3
record • replay • simulate

Try LabSat **FREE** for 14 days now **www.labsat.co.uk/trial14**

APEX EXPO IPC

MEETINGS & EDUCATION › March 23-27, 2014
CONFERENCE & EXHIBITION › March 25-27, 2014
MANDALAY BAY RESORT AND CONVENTION CENTER
LAS VEGAS, NEVADA

NEW IDEAS...
FOR NEW HORIZONS

- **TRUST** in the **expertise of IPC** – recognized worldwide as the leading standards organization for electronics manufacturing.
- **EXPLORE** **emerging technologies** and the latest innovations that will shape the future of our industry.
- **COMPARE** **equipment, materials and supplies** from more than **400** of our industry's **top companies**, in one location.
- **LEARN** about the **latest research and processes** that will **advance your career** and your company.
- **BE INSPIRED** with **new ideas and approaches** that will turn your vision into tomorrow's realities.

"Attending a CID class, combined with the experience of technical classes, really made my trip to IPC APEX EXPO worth it."

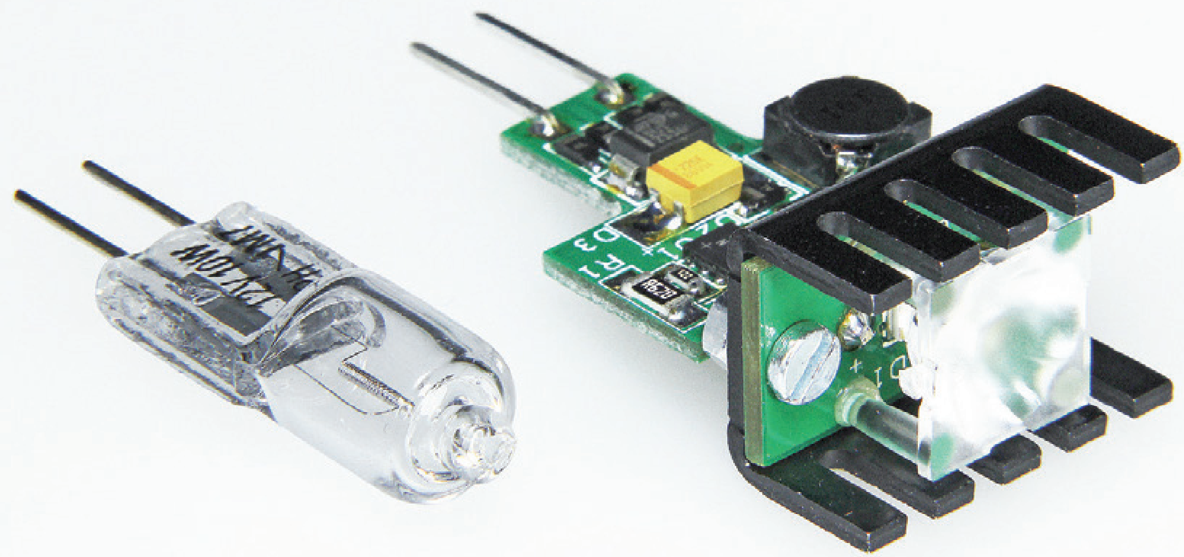
Brent Huggler
EE-CAD Designer
Bourns Inc., Automotive Division

REGISTER TODAY
www.IPCAPEXEXPO.org

DESIGN | PRINTED BOARDS | ELECTRONICS ASSEMBLY | TEST

LED's Replace That Halogen Lamp

Economical alternative for the GU4



By **Fons Janssen**
(Maxim Integrated,
Netherlands)

There are numerous LED alternatives available these days for most types of incandescent and halogen lamps. But as an electronics hobbyist you can of course design and build something like this yourself, as this article illustrates. The LED lamp presented here can replace a GU4 model 10-watt halogen lamp, uses only one-tenth of the power and is barely bigger than the original.

LED lighting has experienced a great ascent, particularly since several government commissions have banned the good old incandescent lamp. Since 100% compatible LED retrofit lamps are very complex, because of all the optical, thermal, electrical and commercial requirements, the design of such a lamp is normally reserved for large development teams. This however, does not have to hold us back from designing such an LED amp ourselves!

The real Elektor reader does not shy away from a few compromises to nevertheless build a working lamp which can also save a lot of energy.

LED alternative for a 12-V capsule lamp

The LED lamp proposed here is an alternative for a so-called halogen capsule lamp, also known as the GU4 (the 4 refers to the distance of 4 mm between the pins). The headline photograph shows both lamps side by side. In practice the light output of the LED lamp at 1 watt is comparable to the light output of a 10-watt halogen capsule lamp. This is a saving of 90%!

By selecting 12 V as the power supply voltage, we don't have to worry much about electrical safety. That is because the 12 VAC transformer provides sufficient isolation from the powerline so that it is safe to touch the circuit without having to place trust in an earth-leakage circuit breaker. As can be seen from the photo, the dimensions of the LED variant are somewhat larger than the original, with the consequence that it is not possible to replace every GU4 in any light fixture. There are, however, also applications where the additional space requirement is not a problem. A nice example is the so-called 'starry sky', where a matrix of lamps is spread across a ceiling. Our alternative fits exactly in the socket and, at most, will stick out a little further than the original. Because the lamp is not enclosed it is very easy for it to dissipate its heat. Because an LED is a point source it is not pleasant to look at it directly. A special lens solves this problem and provides a nice beam of light.

Construction

Which challenges remain? The design of the lamp must be simple, robust and frugal. The individual components have to be readily available and the whole assembly cannot be too expensive.

As can be seen from the photo, this lamp shines from simplicity. There is no need for a separate enclosure; two circuit boards are connected together using standard right-angle header pins. The LED board is clamped firmly to a TO-220 heatsink using two M2 bolts. The small lens is attached with a couple of drops of glue to the LED board. The main connection pins use the same header pins as those used to interconnect the boards.

The LED that has been used here is the so-called *Rebel* from Philips Lumileds. This is readily obtainable and available in various colors and color temperatures. By limiting the power to 1 watt we kill two birds with one stone. Firstly, it is possible to use any type of Rebel-LED with that a similar power rating. The second advantage is that the heatsink can remain quite small and finally the LED will operate more efficiently since it doesn't become so hot. That is because LEDs do not like heat and operate more efficiently at lower temperatures.

Electronics schematic

Figure 1 shows the schematic for the circuit for the drive electronics. Diodes D1a through D1d form a bridge rectifier for the incoming 12 VAC voltage. The rectified voltage is filtered a little by C1. A tantalum capacitor is used because a ceramic type was found to produce an annoying 100 Hz (120 Hz) audible hum. D3 was added at a later stage of the project, after a few prototypes without this transient over-voltage protection went up in smoke. The LED driver IC, a MAX16820 van Maxim (IC1), has a 'buck' architecture and uses an external switching FET (T1). The IC attempts to regulate the voltage across shunt resistor R1 to 200 mV, so that the current through this resistor, the coil and the LED equals 200 mV/620 mΩ ≈ 320 mA. The forward voltage drop across the LED is about 3 V, so that the

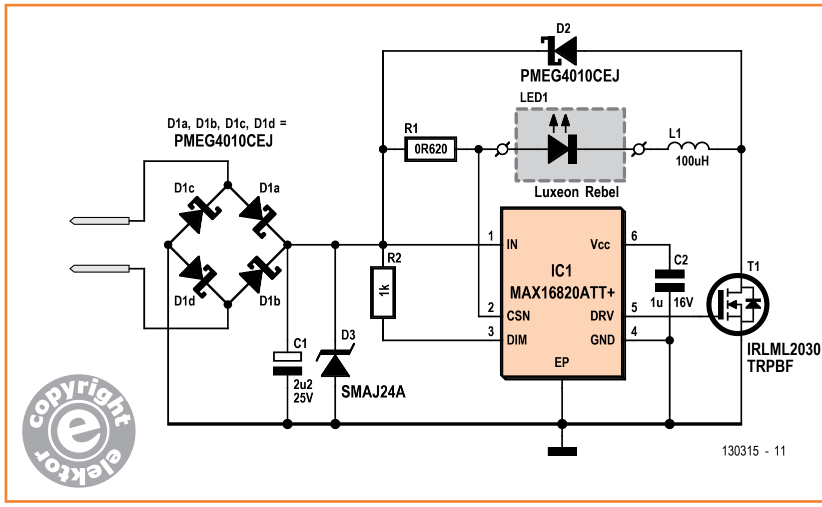


Figure 1.
Schematic diagram for the control electronics.

power in the LED amounts to about 1 W (3 V x 0.32 A). The control itself is quite simple. When the voltage across the shunt resistor is smaller than 190 mV FET T1 will be turned on. The current will increase linearly at a rate of

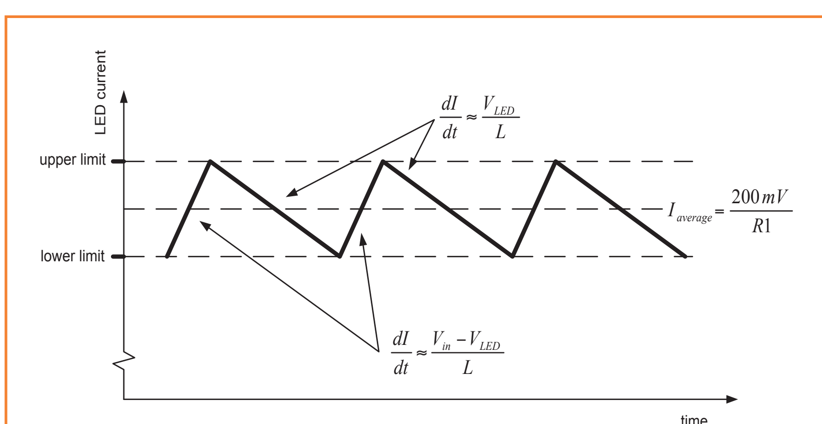
$$dI\ /\ dt \approx (V_{in} - V_{LED})\ /\ L$$

As soon as the voltage has reached a value of 210 mV the FET is turned off again, with the result that the current will decrease at a rate of

$$dI\ /\ dt \approx V_{LED}\ /\ L$$

noting that the current is returned to the input via D2. As soon as the bottom threshold of 190 mV is reached again, the FET is again turned on and the cycle repeats. The result is a sawtooth shaped current as shown in **Figure 2**. The switching frequency depends on the input voltage. Since the input voltage is not constant, the switching

Figure 2.
The sawtooth shaped current through the LED.



Component List

Resistors
R1 = 0.620Ω (SMD1206)
R2 = 1kΩ (SMD0603)

Capacitors
C1 = 2.2µF 25V (SMD3528)
C2 = 1µF 16V (SMD0603)

Inductors
L1 = 100µH (e.g. LPS5030-104MLB)

Semiconductors
D1a,b,c,d,D2 = PMEG4010CEJ (SOD323F)
D3 = SMAJ24A (SMA)
LED1 = Luxeon Rebel LED
T1 = IRLML2030TRPBF (SOT23)
U1 = MAX16820ATT+ (6 TDFN–EP)

Miscellaneous
PL1 = right angled pinheader (4 per lamp)
Lens: RS Components # 697-4288
Starry Night set: e.g. Conrad Electronics #'s 570590–89, 570591–89, 570592–89 or 570593–89

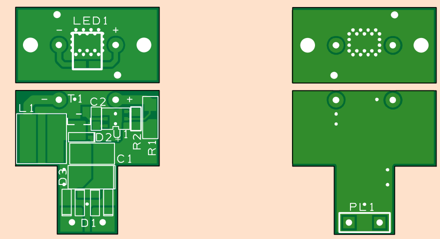


Figure 3. The printed circuit board design comprises two parts. Because of space constraints D1a through D1d are indicated with a single 'D1'.

frequency will vary with the input voltage. For a detailed explanation of this LED driver refer to the article 'Get a Grip on LED Drivers' in Elektor April 2009 [1].

The design of the circuit boards

The printed circuit boards were designed using the well-known CAD program DesignSpark. The only through-hole components are the header pins—all the other parts are SMD types, which allows the boards to be as compact as possible. The design is shown in **Figure 3**. The DesignSpark design files are available as a free download [2].

To make the cooling of the LED as good as is possible, there are a large number of thermal vias around the LED. The purpose of these is to conduct the heat from the LED to the other side of the board, which in turn is in contact with the heatsink. Use the LED board as a template to locate the holes in the heatsink. The standard hole in a TO-220 heatsink corresponds with one of the LED terminals. The other three holes can be drilled using a 2-mm drill.

The solder mask on the bottom of the LED-board covers all the electrical connections to prevent short circuits. The header pins therefore have to be soldered only on the top side of the circuit board; make sure that the header pins do not short to the heatsink. It is therefore very important that the holes are exactly in the right place. On both sides of the LED are two more small holes, which are for the alignment pins of the lens.

The mounting of most SMD components using tweezers and a fine soldering iron should not be a problem if you have some experience with soldering. The IC and the LED are a little more troublesome and require a hot-air workstation or a reflow oven.

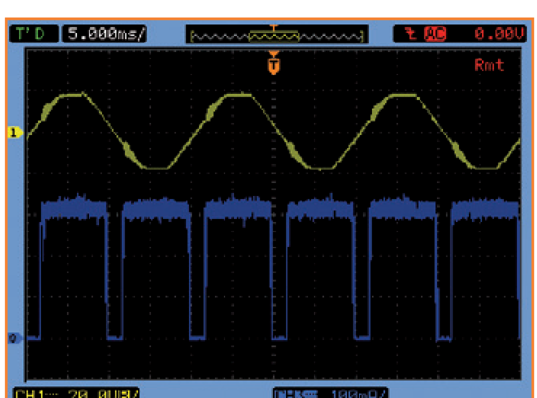


Figure 4. Oscilloscope printout showing the input voltage (top) and the LED current (bottom).

Measurements

Figure 4 shows the LED current together with the input voltage. As can be seen, the current is regulated nicely at 320 mA for most of the time. However, the driver circuit cannot operate when the input voltage drops below 4.5 V. The consequence of this is that the LED current is interrupted around the zero-crossings of the line voltage. This is however not visible because

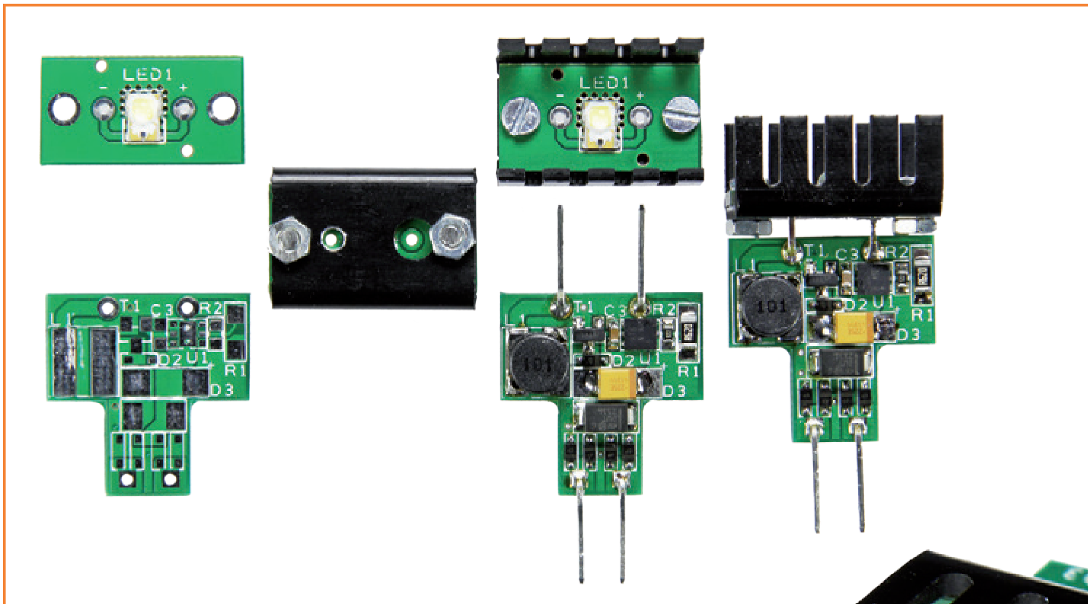
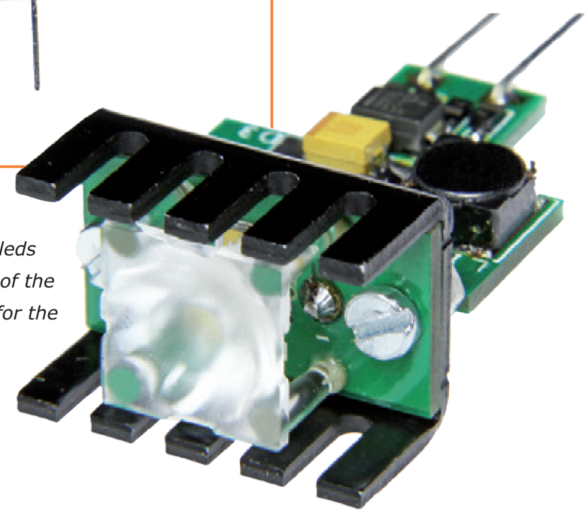


Figure 5.
Here you can see the various stages of building the lamp: the individual circuit boards (but with LED mounted), the holes in the bottom of the heatsink, the assembled boards with mounted heatsink, the boards soldered together and finally a detail of the lens in front of the LED.



the human eye cannot sense this frequency of 100 Hz (120 Hz). As a result the filter capacitor can remain small. The LED is on for about 80% of the time, so that the effective power consumption is 0.8 x 320 mA x 3 V = 0.77 W. The efficiency of the driver circuit, including the losses in the bridge rectifier, amounts to about 77%, so that the lamp has a total power consumption of about 1 W. By changing the value of R1 it is possible to increase or decrease the LED current to your requirements. Just make sure that the temperature of the LED does not become too high.

(130315)

With thanks to Philips Lumileds for help with the mounting of the LEDs on the circuit boards for the prototypes.

Internet Links

[1] www.elektor-magazine.com/080888

[2] www.elektor-magazine.nl/130315

Electronic transformers

A source of headaches with 12-V LED lighting is the compatibility with electronic halogen transformers. These transformers come in many variants, but have as a common requirement that they need a substantial load connected to their output. If that is not so (and this is the case with LED lamps), they will protest. The consequence is that the LED lamp will not turn on at all or starts to flash annoyingly. The solution to this requires quite a complex circuit that is outside the scope of this application. This lamp can therefore only be powered from a standard 50-Hz (60-Hz) line transformer. By the way, an old laptop power supply is also a good alternative. In principle the diodes D1a through D1d could be omitted, because such a power supply has a DC output. However we recommend that you leave them as they will then operate as reverse-polarity protection, for it is easy to make the mistake of reversing the lamp in its socket. Keep in mind that the LED power when operating a DC input will increase by about 25% because there are no zero-crossings and the LED is on continuously.

Model Train Dual Radio Control

Using low-power radio devices

By **Walter Trojan**
(Germany)

Here's the problem—two grandchildren each with a remote controlled train set but both IR remote controllers use the same channel! Short of banishing one to play in another room the author came to the rescue with the help of radio modules.

The Royal Express [1] is a cheap model train set manufactured in China and traded from Hong Kong by Golden Bright Manufacturer Ltd. It comes complete with steam locomotive, tender and wagons with trees, station, houses and rocks—in fact everything you need for a simple layout. Train control is via an infra-red remote controller which allows you to make the train go forward or backward, make noise, smoke and lighting. Above all the set runs from batteries so there is no risk from dangerous voltages. The set is advertised as suitable for children of ages 4 to 8. Above all it is relatively inexpensive so if anything goes wrong during the hacking process it will not be too much of a loss although there may be tears shed.

From IR to RF

For such a low cost kit it's probably a bit unfair to expect anything too sophisticated in the equipment design. The main problem for the author is that the IR remote controller units use the same "channel". With two layouts in the same room (or two trains on the same layout) signals from the remote controllers interfere with one another so the trains cannot be independently controlled.
First thoughts were to check out the remote controller to see if was possible to switch channels. With the covers off I couldn't identify any obvious wire links or pads on the PCB that could be bridged to change the channel. All of the train sets must work on just the one channel. Without a circuit diagram it was going to be difficult if not impossible to make the necessary modifications to the transmitter and receiver.
It just so happened that I had been experimenting with some wireless transceiver modules type RFM12. The planned home control system can be put on hold; harmony in the household gets higher priority. First it was important to find out

if there was room for additional circuitry. Under the roof of the engine tender there was space enough for a small PCB, more than enough for the radio receiver in fact.

So the solution was clear: an RF data link could be used to transfer control information. At the receiver end we just need to translate the received commands into the signals that would be produced by the standard IR receiver that came with the set. These will then be passed on to the train controller electronics. For the wireless link, RFM12 modules can be used which operate in the 433 MHz ISM radio band. In countries where 433 MHz is not allocated to license-exempt SRD (short-range radio), 868 MHz or 915 MHz modules should be used instead.

The first task was to work out the IR transmitter signals corresponding to the individual commands sent by the remote controller. Hooking up a logic analyzer to the IR receiver output it was possible to capture the control pulse patterns. The control message consists of a sequence of five pulses. The commands are coded by varying the pulsewidth and space ratios. The message pulse sequences are given in **Table 1**.

In place of the IR messages the new RF link will send command bytes that the receiver microcontroller decodes and translates into the pulse sequences recognized by the train's built-in controller. A secure RF messaging protocol had already been developed by the author (see the RFM12 Library article in this magazine) and is outlined in **Table 2**.

Some characters (such as 0xAA) are RFM12 control characters and can't be used in the message strings of version 1 of the protocol (in the mean time release 2.0 of the protocol has been developed which gives better transmission speed thanks to the use of Hamming-Code [2]). A transmission rate of 4800 Baud is more than adequate for this application.

Transmitter and receiver: almost the same hardware

The transmitter schematic in **Figure 1** is a combination of a 20-pin Microchip microcontroller type PIC18F14K22 (IC2) and a small transceiver module type RFM12B-433-D (MOD1) from Hope RF [3]. The module supplies a crystal-derived clock to the microcontroller pin 2 (RA5/Osc1). Data transfer between the chips is taken care of by

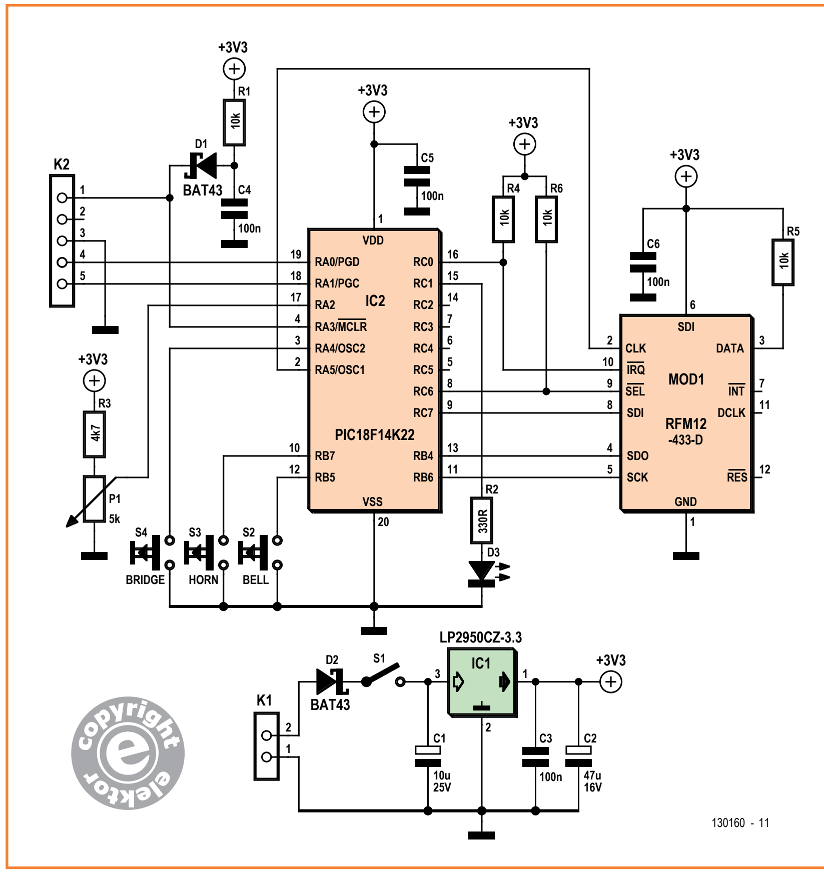


Figure 1. The transmitter circuit.

Table 1. Functions, IR messages and RF commands

Function	IR Message	RF Command
Slow forward	I L I P I P I P I E	0x51
Fast forward	I P I L I P I P I E	0x52
Stop	I P I P I P I L I E	0x59
Slow reverse	I L I L I P I P I E	0x53
Fast reverse	I P I P I L I P I E	0x54
Sound1 Horn	I L I P I L I P I E	0x55
Sound2 Bell	I P I L I L I P I E	0x56
Sound3 Bridge	I L I L I L I P I E	0x57

Key: I = Pulse, active Low, 660 µs duration
P = short pause, High, 1130 µs duration
L = long pause, High, 2270 µs duration
E = End, High, 5 ms

Table 2. The RF communication protocol

Byte	Function	Value
0	Target device address	0x22
1	Transmitter address	0x11
2	Message length	0x08
3	Command	0x51…0x59
4…6	Reserve bytes	0x55
7	CRC16 high	0xXX
8	CRC16 low	0xXX

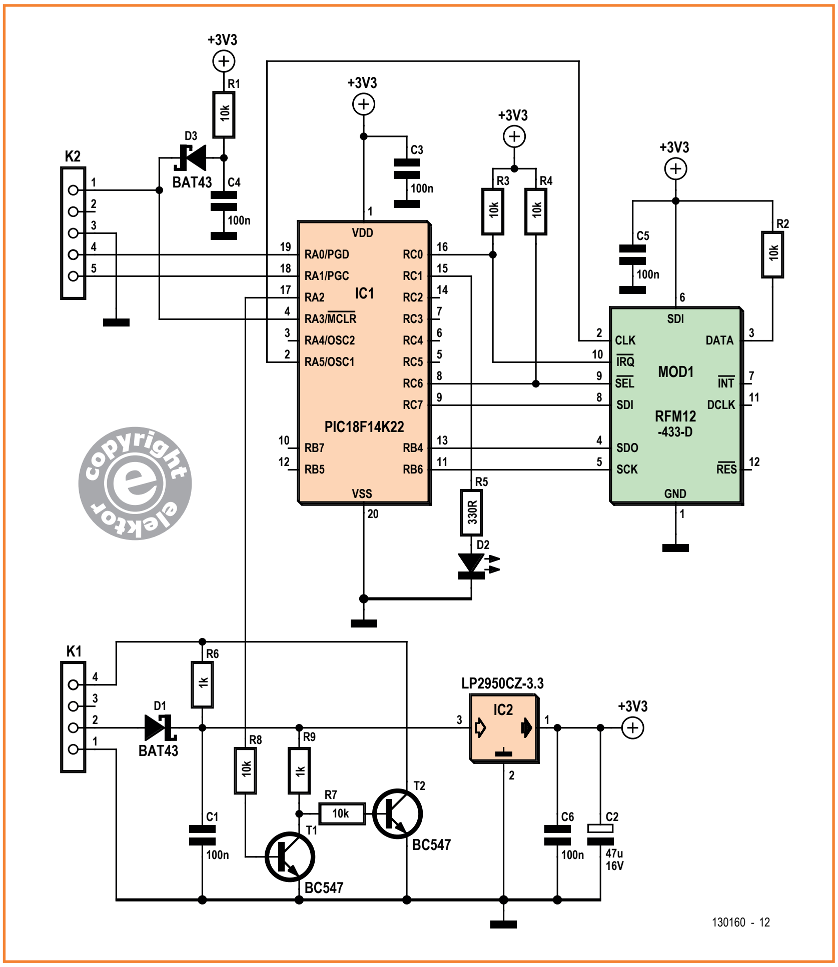


Figure 2.
Beware of confusion:
The receiver circuit is almost identical to the transmitter's.

an SPI interface connecting to the SDI, SDO and SCK of the RFM12 board. Generation of the SPI signals is taken care of by firmware. Signal nSEL/RC6 activates the transmitter chip and the signal nIRQ/RC0 indicates when the transmitter buffer is empty. Further tests on the PIC SPI interface indicated that the RFM12 board can handle a serial clock frequency up to 2.5 MHz. A quarter-wave length of wire is used for the antenna. Pull-up resistors R4, R5, R6 ensure the correct quiescent level on the Active-Low signals nSel and nIRQ.

Table 3. Relationship between control knob position and speed	
Value	**Command**
<22	Fast reverse
23 to 43	Slow reverse
44 to 82	Stop
83 to 104	Slow forward
>104	Fast forward

The controller is slightly modified from the original unit. A pot is now used in place of the five pushbuttons to provide slow/fast forwards and slow/fast reverse plus stop. The stop position is the control knob at mid-travel. Turning clockwise is slow forward and fully clockwise is fast forward. Counter clockwise from the central position achieves the same speeds in the reverse direction. Compared to the original controller it is more intuitive and the young train enthusiasts actually prefer to use it.

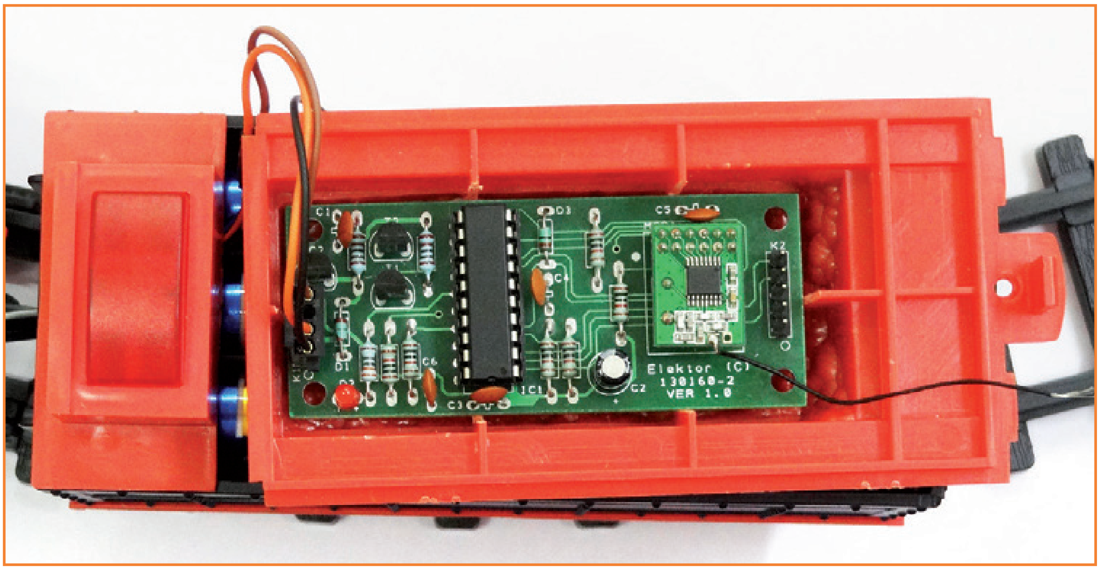
From the circuit function the analog input RA2 sees an input voltage varying in the range from zero to half supply voltage. The measured value is converted into an 8-bit digital value in the range of 0 to 128 by the A/D converter. This translates to 5 speed ranges (this is not an analog control function) given in **Table 3**.

Also on the controller are pushbuttons for the sound generator: Bell (S2), Horn (S3) and 'Bridge' (S4) (sounds like the train is passing over a bridge). These signals connect to inputs RB5, RB7 and RA4 (with pull-up resistors) on the microcontroller. The LED connected to RC1 flashes each time a command message is sent.

The double-sided transmitter PCB has been designed in the Elektor Lab using the DesignSpark [4] software. Component fitting should be quite easy; no SMD components are used in the design. The RF module is available with either SMD mounting pads or a DIP version with two rows of pin headers (at 2 mm grid spacing). This version can be soldered directly to the PCB. Note that the antenna, made up of a 17 cm (6.5-inch) length of copper wire (for 433 MHz), is soldered on the topside of the board by the SMDs (see **Figure 1**). For 868 MHz, the antenna length is 8.5 cms (3.3 inches)

A 9 V battery connected at K1 provides power to the circuit and low-drop regulator LP2950CZ-3.3 provides the on-board 3.3 V, ensuring that every last drop of juice gets used up from the battery. Diode D2 in the supply protects the circuit from accidental reverse polarity connection. The switch type for S1 is not important and depends on the type of housing you use for the transmitter. The transmitter circuit's current drain on the 9 V battery pack is 4.5 mA quiescent, rising to 11.8 mA when transmitting.

The PIC controller can be ordered (just like the PCB) ready programmed from [5]. It is also possible to program it yourself if you have a pro-



grammer. Free software downloads for this project from the Elektor site include the ready-assembled firmware and Pascal source file. Firmware for this project can be compiled using versions 5.60 or 6.01 of Pascal Pro from MikroElektronika [6]. The fully functional compiler is free to use for programs less than 2 KB in size. The microcontroller connector K2 hooks up to a Microchip PICKit3 programmer [7] for in-circuit programming.

The receiver circuit (**Figure 2**) corresponds quite closely to the transmitter circuit. Not only is there the same PIC18F14K22, RFM12B-433 combination, they are also identically connected. The microcontroller built into the train is controlled by active-low pulses from the receiver unit via a two stage transistor driver consisting of T1 and T2. The output control pulse sequences are given in **Table 1**.

Power is supplied by the battery pack (2 x 3 AA cells) in the train. The battery supply attaches to connector K1 through protection diode D1 and voltage regulator IC2. The 3.3 V output powers the circuit. LED D2 is a status indicator, it gives a short flash every 2 s while no data is received and extends to double the length when a message is received. The PIC is programmed in exactly the

COMPONENT LIST, Receiver

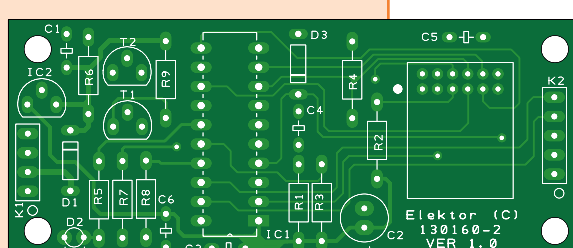
Resistors
R1–R4,R7,R8 = 10kΩ
R6,R9 = 1kΩ
R5 = 330Ω

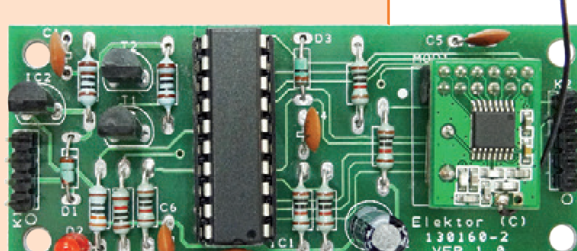
Capacitors
C1,C3–C6 = 100nF
C2 = 47µF 16V

Semiconductors
D1,D3 = BAT43
D2 = LED, 3mm, red
T1,T2 = BC547
IC1 = PIC18F1422-I/P, programmed, Elektor Store # 130160-42, [5]

Miscellaneous
MOD1 = RFM12B-433-D, 3.3V version (Hope RF), or 868 MHz 3.3V version (country-specific)
K1 = 4-pin pinheader
K2 = 5-pin pinheader
PCB # 130160-2 [5]

www.elektorpcbservice.com





same way as the transmitter unit using PICKit3 connected to K2.
The receiver circuit's current drain on the 4.5 V battery pack is 19.5 mA quiescent, rising to 21.2 mA during message reception.

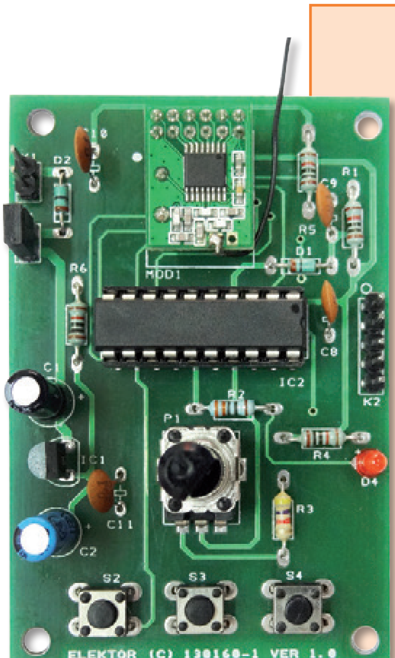
Firmware

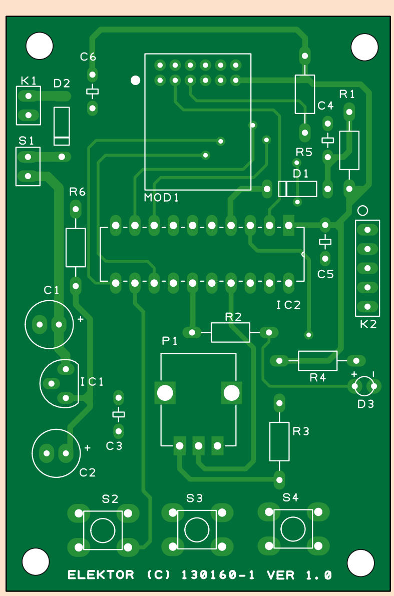
The transmitter firmware continually reads the value of the speed/direction pot and pushbuttons. When a change is detected the corresponding command is sent to the train receiver. After the PIC microcontroller and the RFM12-433 module have been initialized the firmware enters an endless loop where it reads the input voltage level from the speed control pot and generates a command. Note that 868-MHz modules may require different initializing codes. When this command is identical to the previous one nothing is sent out. It will only be sent if it is not the same. The microcontroller also takes into consideration the current status of the train: suppose you were to spin the pot quickly from say maximum forward to maximum reverse. The controller first sends out a Stop command and then a Slow Reverse command followed by Maximum Reverse. This puts less strain on the mechanism and pauses between the commands help to give a more realistic motion. In the second part of the endless loop the sound generator pushbuttons are polled. When the train is running the corresponding sound command is sent to the train.
Similarly the receiver firmware executes an endless loop after everything has been initialised. In the loop it checks if a new message has been received. When an error-free message is received the command byte is interpreted to select the corresponding message defining the pulse/pause sequence sent to the train's built-in controller. For security the process is repeated three times.
Now with the new controller working, peace has broken out in the playroom. The only source of conflict now is whose turn it is to use grandpa's controller.

(130160)

Internet Links

[1] Manufacturer: www.golden-bright.com/
[2] http://en.wikipedia.org/wiki/Hamming_code
[3] www.hoperf.com/rf/fsk_module/RFM12B.htm
[4] www.designspark.com/eng/page/designspark-pcb-home-page
[5] www.elektor-magazine.com/130160
[6] www.mikroe.com
[7] www.microchip.com/pickit3





COMPONENT LIST, Transmitter

Resistors
R1,R4,R5,R6 = 10kΩ
R2 = 330Ω
R3 = 4.7kΩ
P1 = 5kΩ trimpot

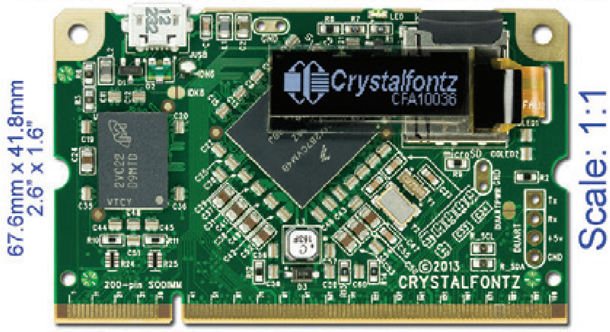
Capacitors
C1 = 10µF 25V
C2 = 47µF 16V
C3–C6 = 100nF

Semiconductors
D1,D2 = BAT43
D3 = LED, 3mm, red
IC1 = PIC18F1422-I/P, programmed, Elektor Store # 130160-41, [5]
IC2 = LP2950CZ-3.3/NOPB

Miscellaneous
MOD1 = RFM12B-433, 3.3V version (Hope RF), 915 MHz or 868 MHz 3.3V version depending on area.
K1 = 2-pin pinheader
K2 = 5-pin pinheader
S1 = 2-pin pinheader and/or slide switch, 1 make contact
S2,S3,S4 = pushbutton
PCB # 130160-1 [5]

www.elektorpcbservice.com

Throw off the 8-bit ball and chain!



CFA10036 ARM9 + Linux SOM

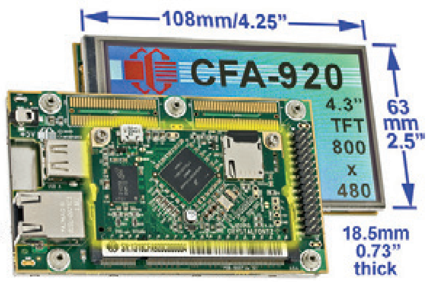
- fast: 454MHz ARM9
- deep: 128/256MB
- wide: 91/126 GPIO
- µSD: 4GB to 64GB
- USB/UART/SPI/I²C
- debug/status OLED
- Linux mainline kernel
- 6xADC/8xPWM/CAN
- $71@Q1 to $48@Q500

Get your project to market fast: lay down a standard SODIMM connector, snap in a CFA10036 System-On-Module and you instantly have access to tons of GPIO and the power of Linux. Leave the 8-bit dark ages behind.

CFA920-TS

This tiny, touch-enabled embedded Linux PC is powered by the CFA10036

- 800x480 color TFT
- resistive touch screen
- 10/100 Ethernet + USB
- 24 GPIO on 0.1"
- 79 more GPIO on 1mm
- 6-channel accel+gyro
- $140@Q1 to $105@Q500



Crystalfontz
www.crystalfontz.com +1 509 892 1200

USB

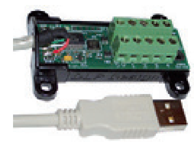
Add USB to your next project. It's easier than you might think!

DLP-USB1232H: USB 2.0 UART/FIFO

HIGH-SPEED 480Mb/s

- Multipurpose: **7** interfaces
- Royalty-free, robust USB drivers
- No in-depth knowledge of USB required
- Standard 18-pin DIP interface; 0.6x1.26-inch footprint

DLP-IO8-G
8-Channel Data Acquisition



Only $29.95!

- 8 I/Os: Digital I/O
  Analog In
  Temperature
- USB Port Powered
- Single-Byte Commands

DLP-IOR4
4-Channel Relay Cable

DLP-TH1b
Temp/Humidity Cable

DLP-RFID1
HF RFID Reader/Writer

DLP-FPGA
USB-to-Xilinx FPGA Module

DLP Design

www.dlpdesign.com

elektor
PCB Service

powered by Eurocircuits

25% Discount on new Elektor PCBs

Benefit now: Elektor PCB Service offers a permanent 90-day launch discount on new Elektor PCBs!

Check www.elektor.com/pcb for an overview of all Elektor PCBs

Arduino on Course (6)

Report on the Mobile Weather Station experiment

Arduino Verkstad ("Workshop") was tasked to create a set of educational experiments aimed at introducing Highschool students to electronics. The primary goal was to make it affordable, yet engaging and challenging enough to trigger their curiosity. Included in the set of experiments is The Mobile Weather Station, a cross-curricular experiment combining natural sciences, physics and technology.

By **David Cuartielles, Clara Leivas,** and **Tien Pham**
(Arduino Verkstad)

For the mobile weather station, we wanted students to be able to actively gather data from different environments—in the city, through the parks, and along the riverbanks. Each environment might provide a different set of data for the students to use for comparison.
We decided to use tethered balloons so that students could control the placement and gather data at higher altitudes. This also gave us the opportunity to include a time-lapse camera enabling the kids to correlate and visualize their data to the different environments.

Decisions, decisions

We set about gathering the required electronics—a temperature/humidity sensor, a microphone, a lightweight spy camera, and a CO_2 sensor. To our knowledge, there were no ready-built wind speed sensors that would fit our requirements. Most anemometers require a stable surface or object to attach to in order to get their readings, but because ours would hang from a balloon, we had to figure out a different method. That, along with a few other unexpected issues, complicated what we thought would be a relatively simple project.
In order to keep costs low, we decided to use a cluster of balloons rather than a single large weather balloon. We purchased a consumer-size helium tank along with regular party balloons. Searching online, we were able to estimate that it would take about 25 or so balloons to lift the Arduino, a shield, the sensors and a lightweight frame or container.
Next up was the anemometer. To measure wind speed accurately wasn't our goal so much as to be able to compare the speeds. Deciding to only measure rotations per second, we thought to use a light dependent resistor (LDR) and an LED to create a sensor. The idea was to place the LDR opposite the LED with a paper disk in between the two. The disk would have a slot cut out and as the blades would turn and intermittently allow light to pass through to trigger the LDR.

Problem solving

As part of our first attempt we tried placing the LED very close to the LDR hoping the magnitude of the readings would obviate the need for shielding. As suspected, this wasn't the case, so we decided to find a dark box to house the electronics.
This presented another set of problems regarding mechanic assembly, and so we set out minds on using an infrared (IR) sensor and a black and white paper disk instead. This looked more hopeful as it required just one electronic component, meaning less room for mechanical error.
We tested the assembly and got it to work. The next part was a bit trickier—how to actually catch the wind in order to spin the disk. Because our aim was to build a lightweight weather station, we couldn't exactly use its weight to stabilize it so that an anemometer could spin while having a stationary point as reference.
Again, to make the assembly reproducible anywhere by students, we decided to try papercraft. We made a simple turbine blade out of paper and tape (**Figure 1**). We mounted it horizontally so that regardless of wind direction, it would spin. In order to keep the balloons stable, we clustered the balloons at the four corners hoping that the weather station wouldn't spin.

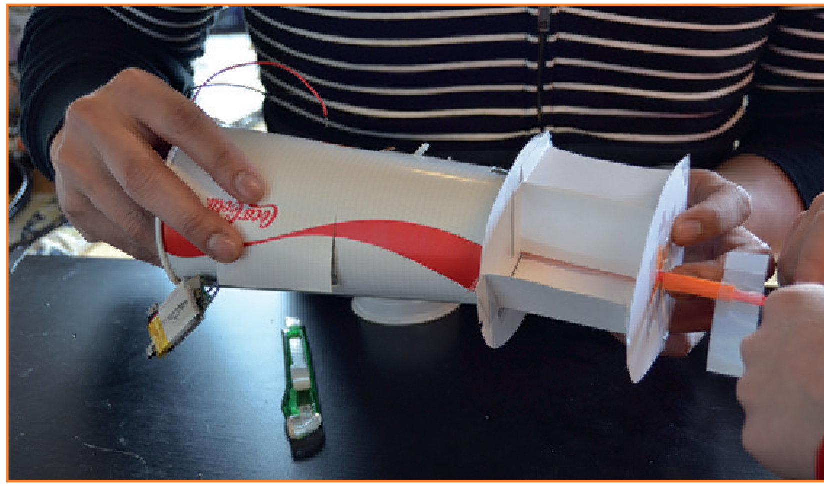


Figure 1. A paper-and-tape crafted model of a turbine for use as an anemometer. Hopefully!

Blowing in the wind

Our paper mockup proved that it might work. In any case, we got the IR sensor and paper disk to work so we were quite happy. Time was now upon us to test the proof of concept—we only had the temperature/humidity sensor, IR sensor, and spy camera.
We first tested the temperature/humidity sensor and camera indoors. We walked over to the University building nearby, inflated the balloons and were quite happy to realize we didn't have to use that many balloons (**Figure 2**). Up it went—we did our high-fives as we loaded the results from the SD card. Now it was time to test the makeshift anemometer.
The West Harbor in Malmö, Sweden, is very windy and we were hoping that it would yield good results. So, out we went with the balloons. As soon as the sliding doors opened the balloons got sucked out and the tether pulled taut. The balloons started to spin wildly and tangled their strings. Not only that, but the winds were so strong that the balloons and weather station blew almost horizontally, seemingly close to the point of breaking the tether string.

Figure 2. About a dozen balloons appeared enough to lift the Arduino Mobile Weather Station assembly, rather than 25 as initially calculated.

Figure 3. Initially impressed with the Rokkaku kite.

It was only then that we realized what should have been obvious. The design of the anemometer wouldn't work because the balloons would never be upright if tethered. At best, it would be at a slight angle. Perhaps it was because of the traditional design of free-floating weather balloons that we didn't consider this, but having a tether changed the dynamics completely.
Defeated, we tried several designs, one of which was to mount the turbine vertically. The problem was getting it to align to the wind so we tried adding a tail similar to that of weather vanes. That shifted the weight of the unit and simply was too unpredictable as it oscillated in the wind.

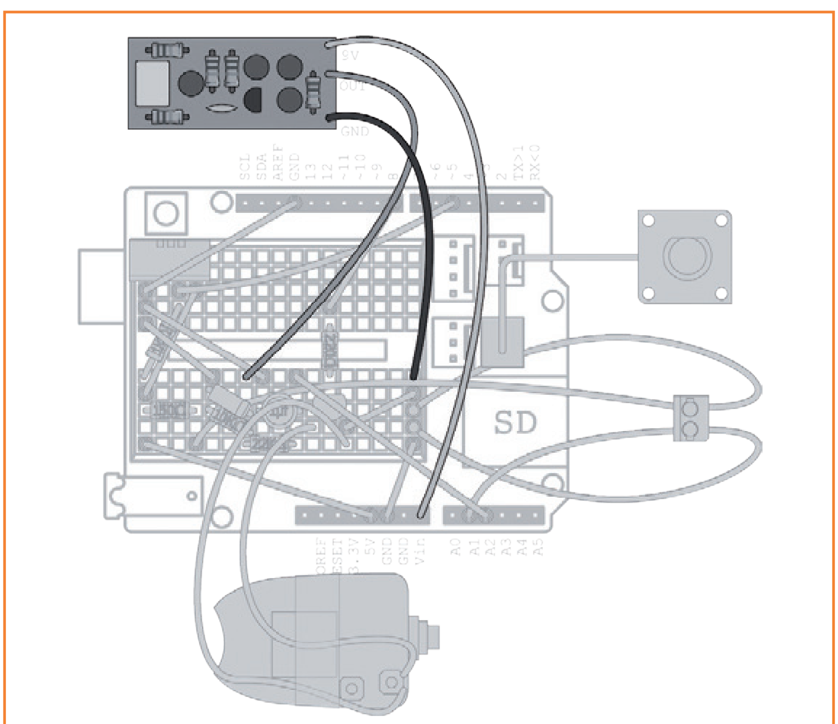


Figure 4. Connection diagram intended for Highschool kids working at the "connect-wire-x-to-wire-y" level.

Bidirectional turbine

Finally, we decided to design and 3D-print a turbine that would blow in both directions. This was achieved by twisting the shape of the blades along its spinning axis. Since the test school we were working with already had a 3D printer, they could at least print the turbine out at a reasonable cost. It was a compromise, but we couldn't think of anything better at the time (and time was pressing).
The person in charge of the mechanics (that would be me, David) didn't have much experience with 3D printing—in fact, it would be his first time. After a few tries, a rough, workable prototype was printed.
We tested it again and were glad to find that, yes, the anemometer worked at almost every angle. At this point, we decided to also laser-cut a frame out of MDF (plywood) so that it could be mounted properly, and construction would be consistent. This would help to enable the sensors, especially the IR sensor, to work well. (In terms of cost, it would be relatively inexpensive for us to use the communal laser cutter to produce lightweight kits that we could send off. Again, a compromise, but still affordable.)
Our joy was short lived as we again went to test it outside (no balloons, just outdoors). Somehow, the IR sensors were receiving a lot of interference being outside. The only solution we could find was to spray-paint the plastic casing black. Eventually, we would have to print the casing using only black plastic since other colors allowed interference. The IR sensor proved to be just as temperamental as the LDR. Moving the sensor a few millimeters closer or farther from the disk would prevent the sensor from reading values correctly. In fact, it seemed that the IR sensor we were using had a small working range tolerance, but we finally managed to get it working.

Goodbye balloons

We found that 3D printers, while being great for prototyping, weren't necessarily the most accurate (perhaps it was simply a matter of inexperience) or smooth. Small points of friction where the print was bumpy meant it now required more wind, thus raising the speed threshold. Another compromise, but one we again decided to take. We thought we were nearing the finish line but then remembered the issue regarding the balloons. The idea of a kite came up and we figured it would be best to try it that way. So, it was off to research kites.

Our first foray into kite making was a box kite. Someone mentioned an anecdote about a box kite capable of producing enough pull to lift a human, so we decided that would be our first model. After building a rather sad looking kite using dowels and trash bags, we managed to get it flying... for about ten seconds before it crashed back down. Adding wings as stabilizers helped, but the area we were in had unpredictable winds and the kite came crashing down—hours of work gone with a few broken dowels.

Eventually, we happened upon a kite design called ROKKAKU—a six-sided kite originating from Japan that was said to be the most stable single-string kite available. We built a mockup with bamboo and plastic sheeting, and got it up and flying (**Figure 3**). We were quite happy with the result, it being the first kite that any of us had ever built. We were set to bring the whole thing together for a test run . . . when it was then decided that all of these elements would be a bit too unpredictable for our purposes. We needed something a bit more foolproof and so we compromised yet again by deciding to hang the mobile weather station from a long stick with a curved hook. This actually turned out be a good and interesting decision because it felt a bit more interactive. Students could direct the weather station into the trees to see if greenery would affect readings, or hover it over a bridge directly above the water.

Satisfied, we went back to working with the electronics. We added the sound sensor and it worked perfectly. The CO_2 sensor was another matter. While it did work, there were complications. It requires warming up and demands quite a lot of care on the source code, and therefore it gets hard to explain to the kids participating in the workshop how the sensor works. **Figure 4** gives an idea of how everything gets connected up to the Arduino.

While we aren't 100% completely satisfied with the way the experiment turned out, we are satisfied with the learning curve. We messed up and failed many times, but eventually, we got there. Sometimes, there is success in failure.

The completed Mobile Weather Station assembly is pictured in **Figure 5** as an exploded view. More of these drawings showing the electronic side of things are available for free downloading from [1].

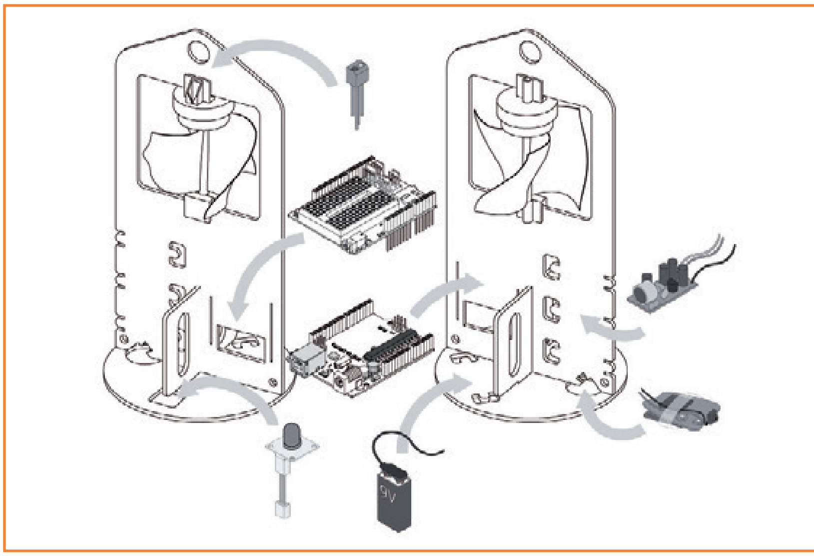


Figure 5. Final assembly of the Mobile Weather Station hardware. Temperature, humidity, altitude and CO_2 level are measured, processed by Arduino and written to an SD card. The system also takes pictures from above. Note the 3-D printed turbine incorporated in the PCB.

Running the show

We tried the weather station hanging from balloons, as well as from a kite. Both attempts at getting meaningful pictures out of the experiment turned out bad. Therefore we decided to hang the station from a long pole with a staff extension (**Figure 6**).

The first time this weather station was put into practice was at the "Etopia Kids" Tech-Camp in Zaragoza (Spain) that took place in June and July 2013. During this event, kids get exposed to different interactive technologies during a period of 5 days. They learn how to count in binary, how to program Arduino boards, get robots to move, and how to read environmental data and show it on a display.

The kids first assemble the station and then go for a hike to capture data at different locations. Temperature, humidity, sound level and wind speed are captured at a constant pace together with an image of the location from above. The participants then return to the computer lab, and using the 'Processing' software application, they map the information with the images and proceed to create a small presentation for the other groups explaining what they learned about the different locations. The Processing "sketch" developed for the project is also available at [1].

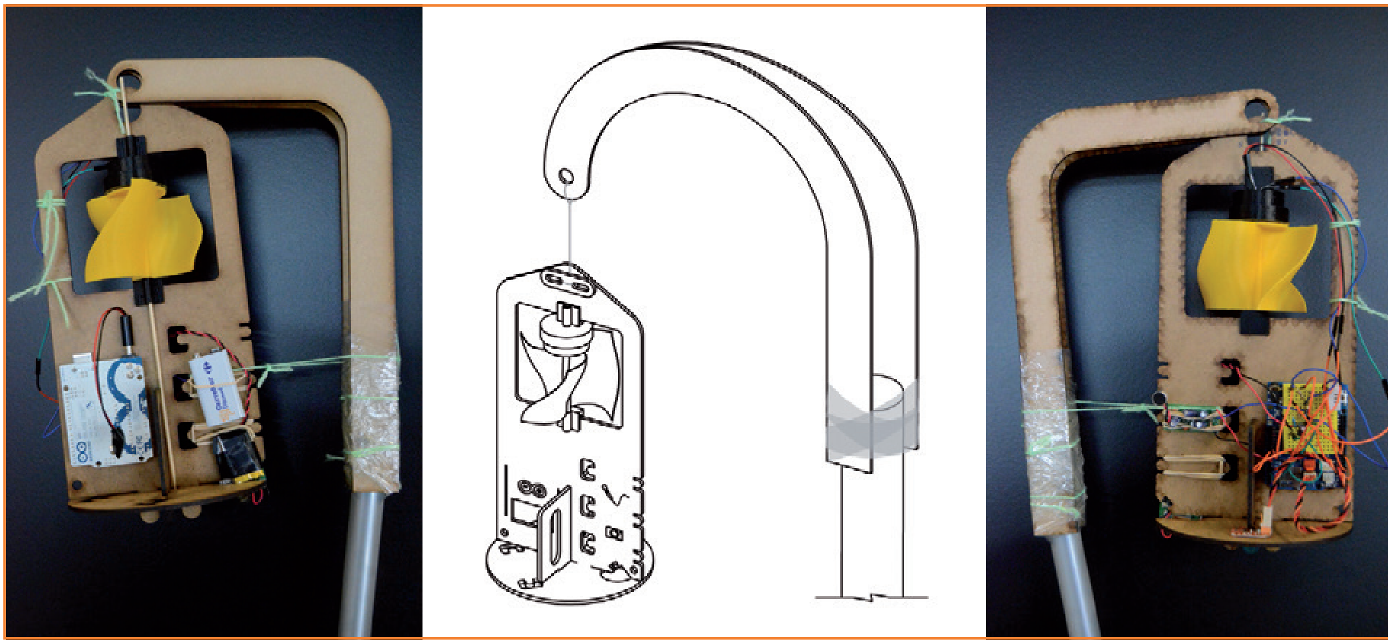


Figure 6. After fruitless attempts with balloons and kites, it was decided to hang the Arduino Weather Station from a long pole. Here's what the teachers had in mind… and an example of how one kid built it. The turbine parts were produced in different colors.

Lessons learned

As the time of writing we have been running the experiment with kids for three days. It will be done a total of 15 times over the following three weeks and 150 kids will get to try the weather stations. One thing we have seen is that there are far too many cables on the design. For a successful continuation of this or even the expansion to more sites, the current Arduino shield design needs to have the components soldered to a board. Also, the camera we hacked for taking the pictures turned to be as weak as one could expect for the price we paid. We had to resolder the wires that "hack" the button on it several times.

We'll definitely develop this experiment further and if you have any feedback, ideas on how to improve it, or know of similar successful experiments that we can refer to, please feel free to let us know at experiment.design@arduino.cc.

The illustrated story

The set of tech drawings presented to students to assist in the assembly of the Arduino Mobile Weather Station is available for free downloading from the Elektor website [1]. One set of drawings illustrates the connection of the external electronics to the Arduino main board ('Balloon-Experiment-*xx*.png), the other, the assembly of the complete unit, including the pole it is hung from (MDF_WeatherStation_Instructions-*xx*.png).

(130043)

[1] Arduino software Processing sketch (.zip file); electrical and assembly drawings (.zip file): www.elektor-magazine.com/130043 (free download).

About Arduino Verkstad

Arduino Verkstad ("workshop" in Swedish) is a newly formed team—each of us specializes in different areas, some of which overlap.

Not everyone was directly involved in the process, but nearly everyone was able to contribute a bit of advice and guidance to the main developers, Tien and Clara.

Active ESD Protection

For microcontrollers and more

This circuit is designed to be used when an application requires a greater degree of electrostatic discharge (ESD) protection than that provided by an IC on its I/O pins. Although arrays of ESD clamping devices are available, they rarely offer precise limiting of voltage overshoots and undershoots.

By **Peter Krüger** (Germany)

Normally when dealing with a microcontroller or other digital circuit the connections on the device are protected against electrostatic discharge. Nevertheless engineers are 4ever taking special precautions when handling such devices to avoid the risks of ESD: the lab will have an anti-static covering on the floor, and nylon clothes and shoes with soles made of insulating material are avoided. And, in case that is not enough, it is normal to wear an anti-static wrist band when moving devices from their anti-static bags to the anti-static bench surface. But what exactly do we mean when we talk about ESD?

The 'human body model' and others

The first model for static discharge, mentioned as early as the nineteenth century, was the 'human body model' (HBM). This takes as its starting point a voltage of up to 40 kV, a body capacitance of a few hundred picofarads and a (skin) resistance of 1.5 kΩ. We find that even with a static voltage of only 10 kV, as might easily be acquired by walking across an artificial fiber carpet in shoes with synthetic soles, it is possible to discharge through a fingertip at peak currents of up to 20 amps! The discharge also happens in a very short period, perhaps measured in nanoseconds.

The HBM was adopted in the electronics industry in the 1970s with the introduction of sensitive JFET devices in space applications. The components were tested using a simple RC circuit like the one shown in **Figure 1**. The discharge current depends only on the resistance in the circuit, and the damped discharge curve is largely free of oscillation and is accurately reproducible.

There are also other models that deal with discharge through a sensitive component, for example when a low-resistance electrical connection is made between two devices (the 'machine model', or MM), or when a static charge present on the device itself is discharged (the 'charged device model', or CDM). Good introductions to this subject can be found at [1] and [2].

ESD clamp circuits

Figure 2 shows the typical protection circuitry provided on a microcontroller's I/O port. This example is from an ATmega; other microcontrollers and logic devices use similar arrangements. Two bipolar protection diodes conduct discharge currents that could cause undershoots or over-

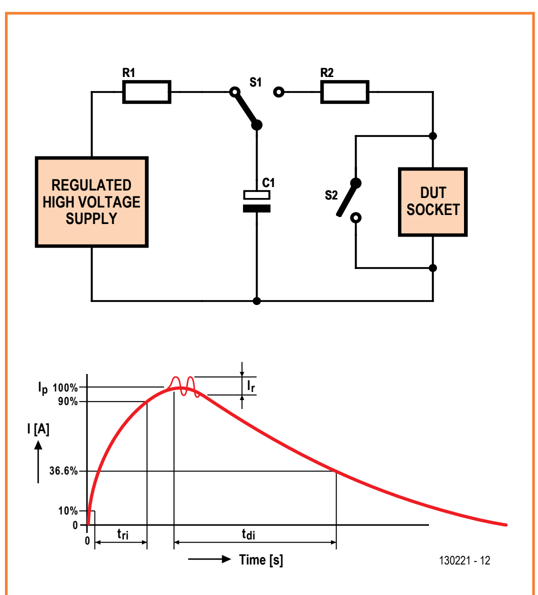


Figure 1.
Standard test circuit and current waveform for the human body model.

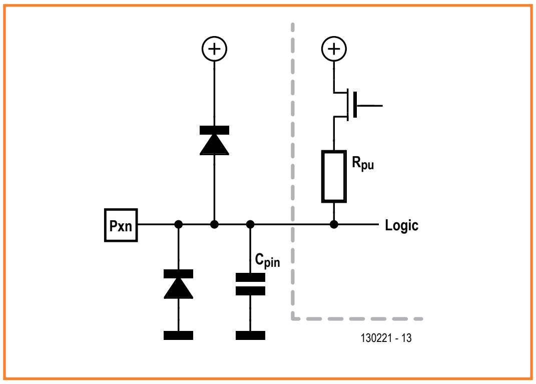


Figure 2.
Typical ESD protection circuit, as found in an Atmel microcontroller.

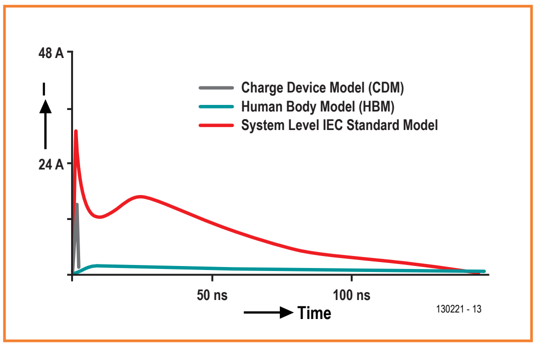
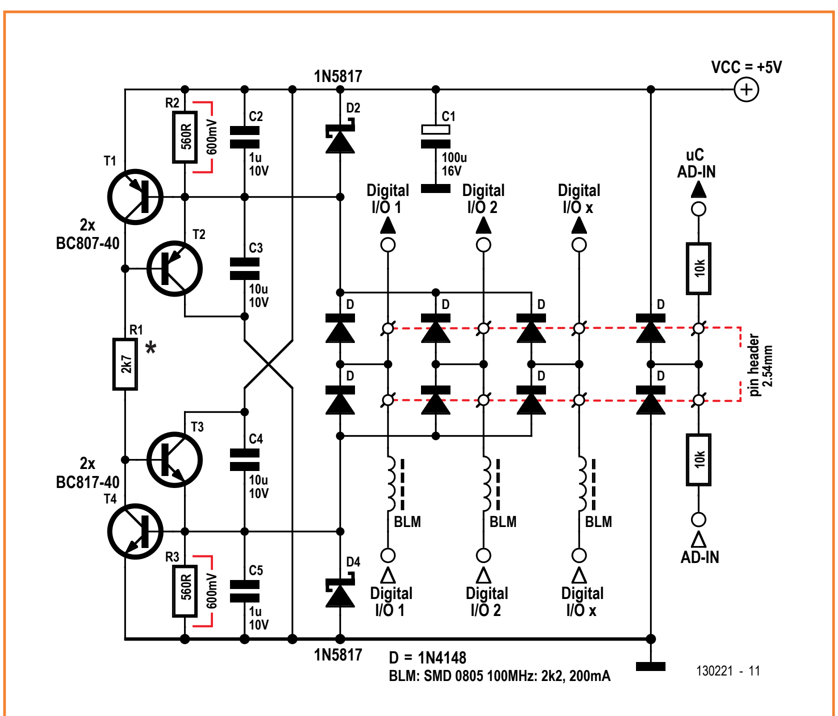


Figure 3.
Current waveform under the system-level model.



shoots to one of the supply rails, either V_{CC} or ground. However, the diodes take about 6 ns before they conduct fully. Since ESD transients can sometimes be considerably shorter than this it is possible that the CMOS circuit structures will be damaged long before the diodes spring into action. The parasitic capacitance of the pin is around 6 pF, and this is quickly charged up by the energy in the electrostatic discharge. Unfortunately we cannot increase this capacitance without increasing the impedance of the pin, which is not desirable.

Standard ESD protection circuits like this one are designed to meet the particular requirements set by the ESD association [3]. However, it is becoming apparent that the traditional models are not appropriate for modern applications. Recent efforts have been directed toward developing a new 'system level model' (SLM) which takes into account the different aspects of the older models. This model employs two stored charges that are discharged in different ways, creating a high-amplitude current pulse that decays very quickly plus a low-amplitude pulse that dies away more slowly. The energy transferred in a discharge under the SLM can be very much higher than that in the traditional models (**Figure 3**).

It is readily apparent that the conventional I/O pin circuitry on the IC is not sufficient to provide ESD protection under this model. Also, the continuing industry pressure to make smaller and more complex structures makes it very difficult for design engineers even to maintain current levels of ESD protection, let alone improve on them. In other words: the silicon area needed to provide ESD protection in accordance with the SLM is simply not available!

For this reason external ESD clamp circuits (see [4], for example) are becoming more relevant. If a component provides only a low level of ESD protection (or even none at all) it is possible to add such a circuit at the points most at risk. The clamp circuits usually use so-called transient suppression diodes (transils or tranzorbs) which, like Zener diodes, start to conduct at a specified

Figure 4. This protection circuit clamps voltage transients outside defined upper and lower thresholds.

threshold voltage. However, unlike Zener diodes, they react quickly and can withstand much higher current transients. There are many variations on the circuit design, but none has exceptional performance and none offers precise clamping of voltage undershoots and overshoots.

State of the art ESD clamping

If we are in the lucky position of not having to worry about the last cent of materials cost or the last square millimeter of board area we can easily create a 'state of the art' active ESD protection circuit from discrete components (**Figure 4**).

The transistor circuit forms a kind of regulated voltage divider. The current through the two resistors R2 and R3 is such that the voltages across them are just enough that transistors T1 and T4 start to conduct and T2 and T3 are just short of saturation. So we have one base-emitter voltage (about 600 mV) across each of these two resistors, which means in turn that the emitters of T2 and T3 are 600 mV below VCC and above ground respectively. The circuit as shown is suitable for a 5 V supply; R1 can be changed to suit supplies of 3.3 V or 2.7 V if needed.

What is the point of this complexity? If the I/O pin is high (at +5 V) the upper 1N4148 switching diode will conduct fully as its cathode is at only 4.4 V. If a positive voltage transient should occur it will be conducted by the 1N4148, without switching delay, to the positive rail by 1N5817 Schottky diode D2, which acts quickly and has a low forward voltage. The same thing happens with polarities reversed when a negative voltage transient (below ground) occurs. Hence the digital inputs and outputs are protected against voltage excursions outside the range of the supply rails. In addition, voltage peaks are limited by the use of suppression inductors. The Murata BLM series inductor [5] presents a relatively high impedance to signals in the 100 MHz range and so can significantly reduce the level of transients.

Although the approach we have described works well with digital levels, it is not suitable for use with signals destined for the analog-to-digital converter (ADC) on a microcontroller. In this case a reverse-biased diode between the signal and each supply rail is required to clamp overshoots and undershoots, with a pair of 10 kΩ series resistors to limit the transient current.

The series-connected capacitors C2 and C3 present a low-impedance path for transients between VCC and ground, and hence spikes on the supply rails will also be conducted away.

(130221)

Internet Links

[1] www.teseq.de/de/de/service_support/technical_information/01_Transient_immunity_testing_e.pdf

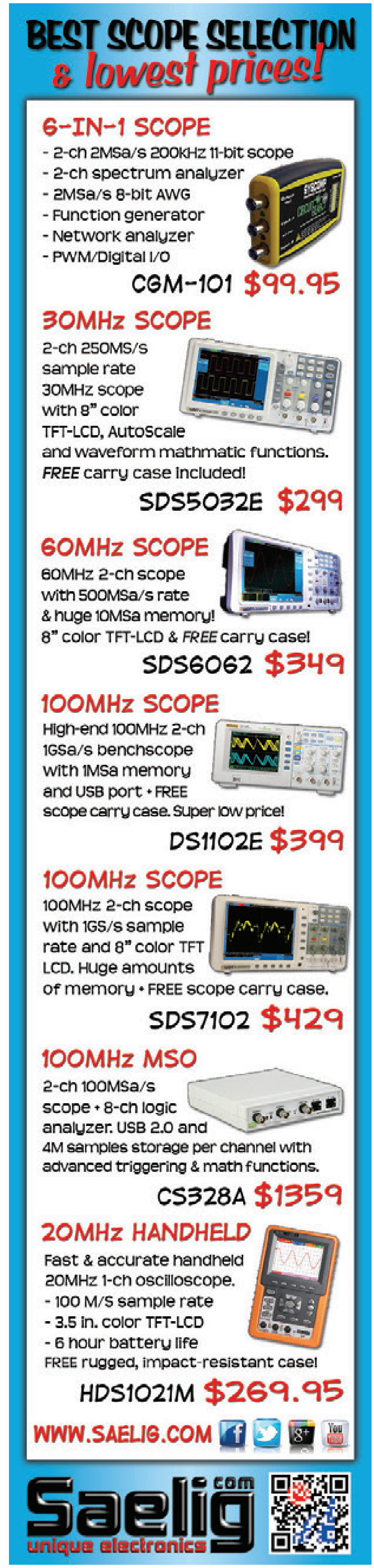
[2] www.ti.com/lit/sg/sszb130b/sszb130b.pdf

[3] http://www.esda.org

[4] www.semtech.com/circuit-protection/esd-protection/

[5] www.murata.com/products/emc/basic/feature/bl_intro.html

Advertisement

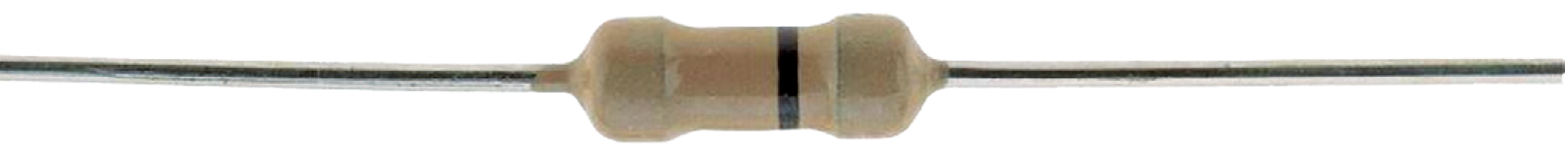


That Zero-ohm Resistor

Weird Component # 2

By **Wisse Hettinga**

One of my favorite topics to discuss with the technicians at Elektor Labs is the existence of the zero-ohm resistor. This subject defies the imagination of an electronic engineer; why does the zero-ohm resistor exist and why is it called a 'resistor' if its value is zero?

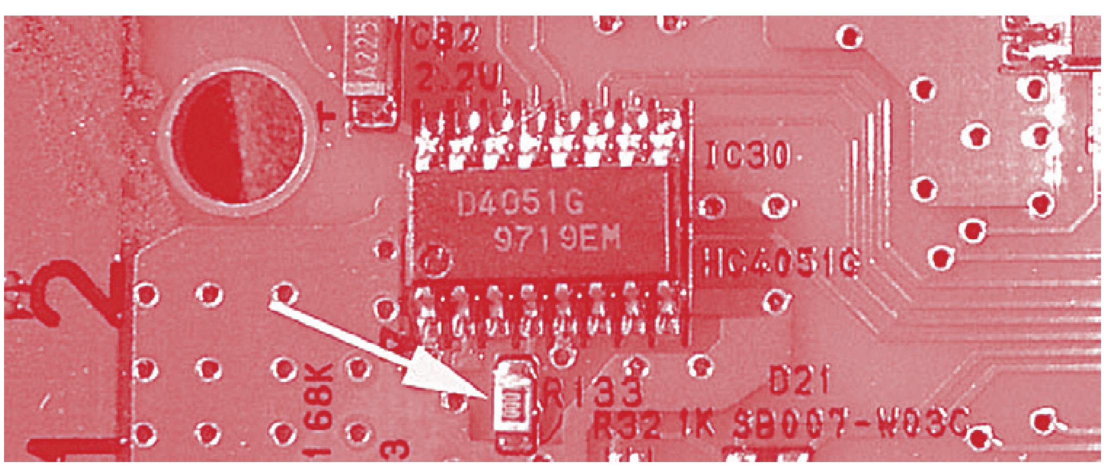


Could it also be called a 'conductor'? Why don't we just call it a jumper? The discussion with my colleagues becomes tricky when we start talking about the tolerance of a zero-ohm resistor. Usually, the tolerance of a resistor is defined using ±. Theoretically this could mean there could be such a thing as a resistor with a value less than zero ohms!
The zero-ohm resistor is quite literally borderline—only few electronic components have the pleasure of being in such an existentialist position. At this point my colleagues usually declare me insane and return to their work, but you won't get rid of me easily.

In practice the zero-ohm resistor has proven to be a very useful component. What makes it so complicated is the fact that we confuse it with a piece of copper wire or a jumper. Which is correct—electronically speaking. But because of its casing this part is often used as a fixed jumper or wire bridge, to set a fixed value on a controller pin, or simply because the designer of the PCB saw no other way of solving the connection between two points. The designer will never admit to this however—it would be too shameful. What matters is that the component and the board can join the assembly line.

My colleagues have had a moment to catch their breath. Maybe I should ask them about capacitors? The ones with infinite capacity!

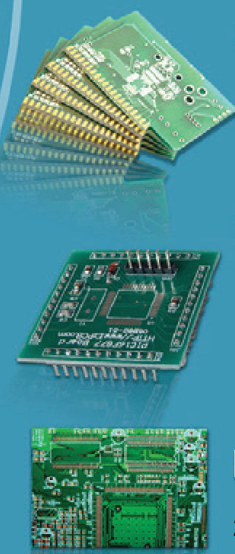
(130396)



www.ezpcb.com

Professional PCB & PCBA Supplier

New Website Is Online!



Better Design
More Powerful
Easier to Use

EzPCB

Instant Quote & Pay

1 to 40 Layers
Prototype to Production
Amateur to Professional

order now

Prototype start at $10/ea

2L 4"x4" each

Free Shipping!

http://www.ezpcb.com

The Convenient All-in-One Solution
for Custom-Designed Front Panels & Enclosures

You design it
to your specifications using our FREE CAD software, Front Panel Designer

We machine it
and ship to you a professionally finished product, no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

FRONT PANEL EXPRESS
FrontPanelExpress.com
1(800)FPE-9060

Create Complex Electronic Systems in Minutes Using Flowcode 6

FLOWCODE6

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.

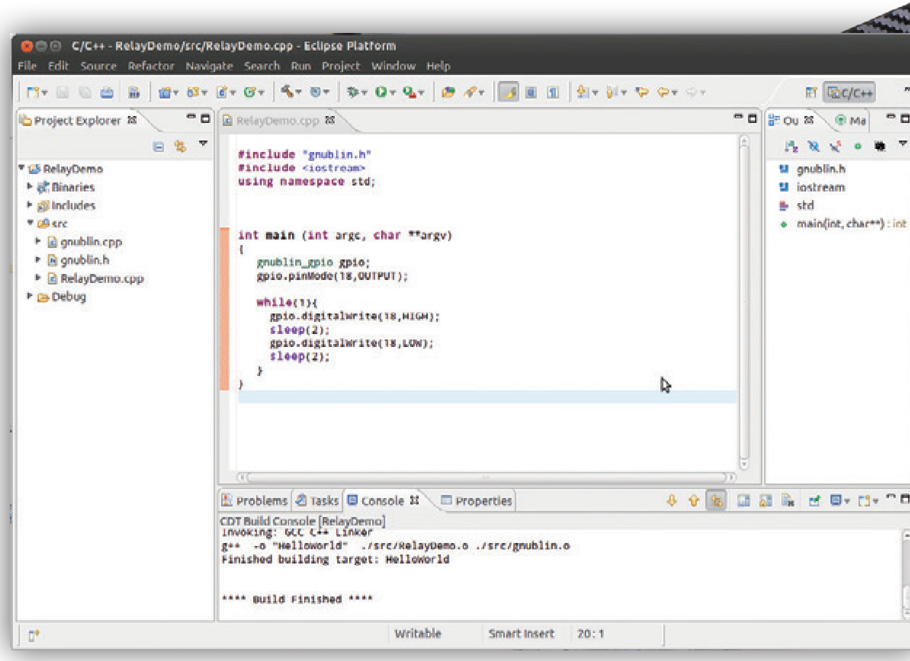
New in Version 6:
- Component Library Expansion;
- Improved Simulation;
- New Test Features;
- 3D Design Environment;
- Third Party Instrument Support;
- Dashboard HMI Components;
- And More!

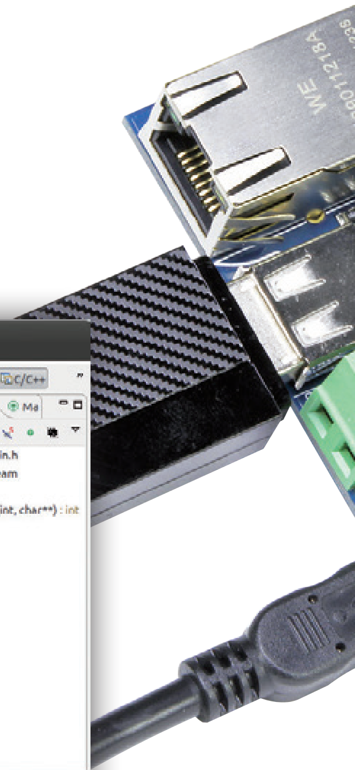
Further Information and Ordering at www.elektor.com/flowcode

Developing With Embedded Linux

using C++ and Eclipse

By **Benedikt Sauter**
(Germany) [1]





Things have been moving on since the start of our series on Embedded Linux. A second version of the Gnublin/Elektor Linux board has been produced, and we have made improvements to the file system used on the memory card. It is now time to get back to the fundamentals, and in this article we turn to a topic of interest to all users: developing your own applications.

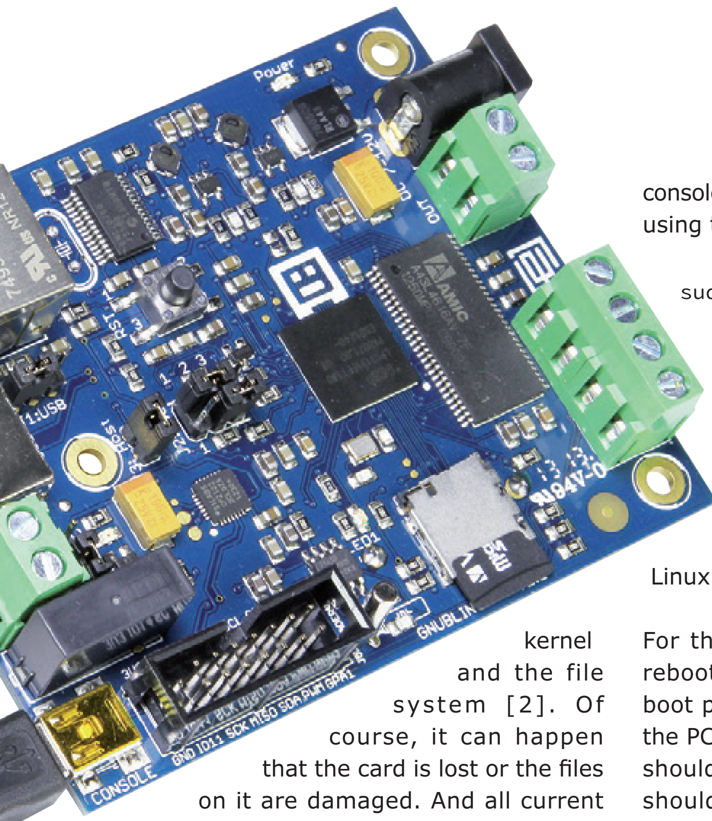
First of all we will quickly review the basic architecture of the embedded Linux operating system. The bootloader copies the kernel from the SD card to the processor's working memory and jumps to it. The kernel includes all the basic operating system 'firmware' functions including the scheduler, drivers and so on. Meanwhile the file system hierarchy is used to store all the applications, user data, log files and the like.

At start-up a number of different programs are loaded from the file system, and executed. If we want to include our own application among these we need to store it in a suitable place in the directory hierarchy and integrate it into the start-up sequence. When copying the application to the SD card it is necessary to take great care to ensure that the structure of the file system is not otherwise disturbed.

Copying the application to the card can be done using a Linux PC and an SD card reader (**Figure 1**). Alternatively, a PC can be connected to a running Linux board over the network and the file transferred that way.

Developing for Linux with Linux

The Elektor Linux board is shipped with an SD card which already contains the bootloader, the



kernel and the file system [2]. Of course, it can happen that the card is lost or the files on it are damaged. And all current users of the first version of the Elektor Linux board will find it worthwhile to create a new SD card: the new file system is much less prone to damage, for example when power is interrupted (see text box).

We have developed a graphical tool, 'Gnublin Installer', to help create new SD cards. The text box describes how to use it.

Gnublin Installer is a program that runs only on Linux PCs. Even if you do not need to use this program we still recommend using a Linux PC for the development of embedded Linux applications. A good choice is the Ubuntu distribution, as there is plenty of help for beginners available on the Internet for it, and the Gnublin distribution for the board was itself developed using Ubuntu.

We described how to install Ubuntu on a PC in the third part of our Linux series [3]. The article also showed how to use the VirtualBox image, specially prepared for *Elektor* readers by the author and which can be downloaded from the Elektor website [3]. This lets Linux run in a virtual machine, avoiding the need for a full installation of the operating system.

A first test

As we have mentioned before, there are a couple of steps you can take to test your embedded Linux board (possibly including a new memory card). Connect the board to a PC using a USB cable (**Figure 2**) and then apply power: the board will automatically start to boot.

Now, on the Linux PC, bring up a command line console and start the 'picocom' terminal program using the following command:

```
sudo picocom -b 115200 /dev/ttyUSB0
```

This will establish a connection between the PC and the board via the USB cable and the CP2102 USB bridge chip. If a Windows PC is being used in conjunction with VirtualBox, it must be installed in such a way that the signals from the CP2102 are visible 'on the inside' in the Linux guest operating system.

For the initial test the board should first be rebooted (by pressing the reset button): the boot process can be followed on the screen of the PC. When the boot process is complete you should see the Gnublin prompt, at which you should enter 'root'. Beginners can now start to familiarize themselves with the basic Linux commands such as 'cd', 'mkdir' and 'cat'. For a

Figure 1.
A reader for microSD cards.

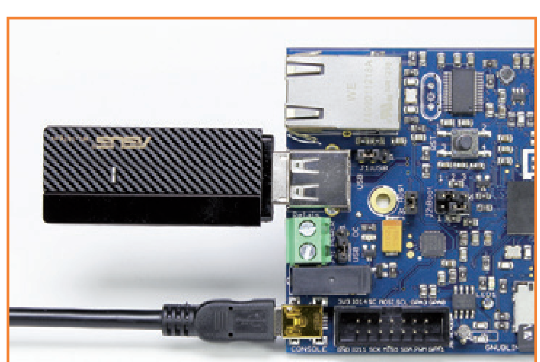


Figure 2.
Connecting to the PC over USB and WLAN.

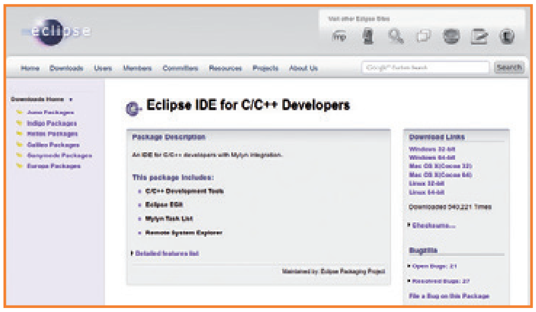


Figure 3.
Downloading the Eclipse development environment.

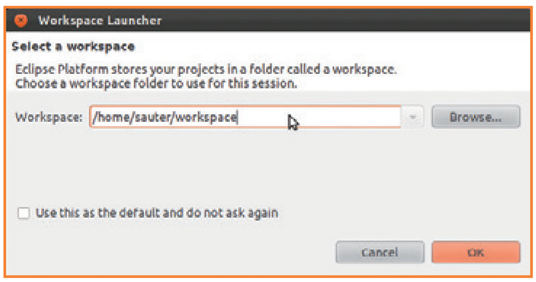


Figure 4.
Here we select where our project will be stored.

little instant gratification you can switch the LED on the board on and off, as described in part 2 of our Linux series [4].

When you have finished experimenting, it is important to power down the board properly using the 'halt' command. Now, when the board is disconnected from the Linux PC, the terminal program will close down automatically and the PC's normal prompt will appear.

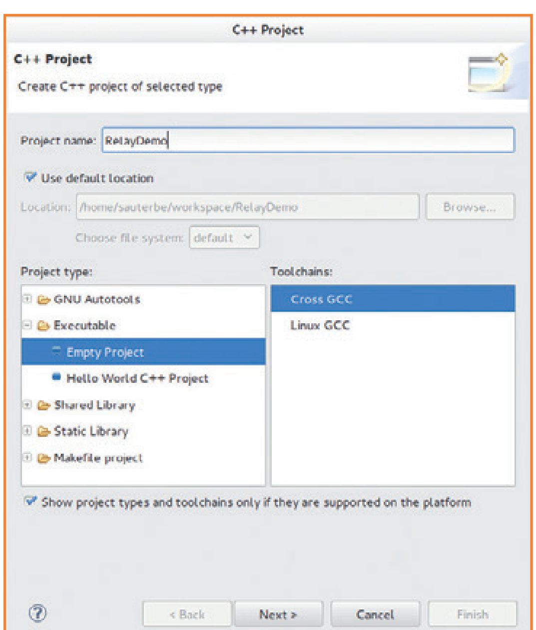


Figure 5.
Our first C++ project.

The development environment

In order to develop our own applications we need to install a compiler and a development environment on the PC. First open a terminal window on the PC and enter the following command:

```
wget http://gnublin.org/downloads/eldk-
eglibc-i686-arm-toolchain-qte-5.2.1.tar.
bz2
```

Now unpack the archive you have downloaded into the file system on the development computer:

```
sudo tar xjf eldk-eglibc-i686-arm-
toolchain-qte-5.2.1.tar.bz2 -C /
```

This command extracts the contents of the tar archive and installs the cross-compiler in the directory /opt/eldk-5.2.1/, a process which takes a little time to complete. The command requires super-user privileges as the archive is unpacked into '/', the root of the file system. If the ready-made virtual machine image is used, the required password is 'elektor'.

The next step is to download the 'Juno' version of Eclipse [5]. Only this version (and not the most recent 'Kepler' version of Eclipse) includes the 'cdt tools', an extension to allow the use of Eclipse for C and C++ development. We will use the Linux 32-bit or 64-bit version (see **Figure 3**). When the download is complete a .tar.gz file will be found in Ubuntu's 'Downloads' folder. Unpack this archive into your home directory using the following commands:

```
cd ~/Downloads

tar -vxzf eclipse-cpp-juno-SR2-linux-gtk.
tar.gz -C ~/
```

A small change to the file '.bashrc' in your home directory will make it easier to launch Eclipse from a terminal window. Launch the 'nano' text editor:

```
nano ~/.bashrc
```

and add the following new line to the file:

```
export PATH=$PATH:~/eclipse
```

Then save the file using Ctrl-O and exit the editor with Ctrl-X. You should now close the terminal window that you are using and open a new one. From now on you can launch Eclipse from the terminal by simply typing:

```
eclipse
```

On start-up you will immediately be asked for a workspace path where a new project will be created. The default path suggested by the system is normally a good choice (**Figure 4**).

Our first project

We can now start on our first small application. In Eclipse's main menu click on File -> New -> C++ Project and then select Executable -> Hello World C++ Program and 'Cross GCC' (see **Figure 5**). We will call our first project 'RelayDemo'.
Now open the window shown in **Figure 6** where we can set the path to the cross-compiler:

```
Prefix: arm-linux-gnueabi-
Path: /opt/eldk-5.2.1/armv5te/
sysroots/i686-eldk-linux/usr/bin/
armv5te-linux-gnueabi
```

The next step is to ensure that the Gnublin C++ library is linked into our project. This library provides an easy way to communicate with the peripherals on the Linux board and expansion boards [2]. The best approach is to download the Git repository that contains a copy of the software archive: it is then very easy to download any updates later.
First install the Git version control system:

```
sudo apt-get install git
```

Now switch to a directory where the source code of the C++ API can be kept, and download the source code from the Internet using the following command:

```
git clone https://github.com/
embeddedprojects/gnublin-api
```

The two files 'gnublin.cpp' and 'gnublin.h' have to be imported into the Eclipse project, which involves copying them into the project directory. To do this, in Eclipse's 'Project Explorer' on the left click with the right mouse button on the project's src directory, and then select the 'Import' menu item, followed by 'General' and 'File System'. In the window that now appears (see **Figure 7**) first select the directory from which the files are to be imported: in our case this means the directory 'gnublin-api'. A list of the files in this directory will now appear, from which you should select the files 'gnublin.cpp' and 'gnublin.h'.
To update the API at a later date simply switch to the 'gnublin-api' directory in a terminal window and type the command:

```
git pull
```

Network connection

Now enter the code from **Listing 1** in the main program source code window. The program can be compiled by clicking on the small hammer icon. The status window below will show whether compilation has been successful or not: with luck, no warnings or errors will be displayed there.
The file 'RelayDemo' will now be found in the directory ~/workspace/RelayDemo/Debug. This is an executable file which needs to be transferred to the Linux board. One way to do this would be

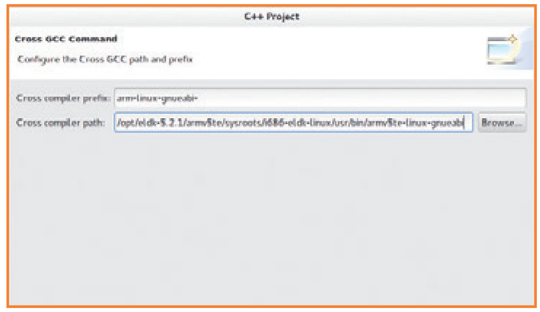


Figure 6.
Selecting the cross compiler for our embedded Linux system.

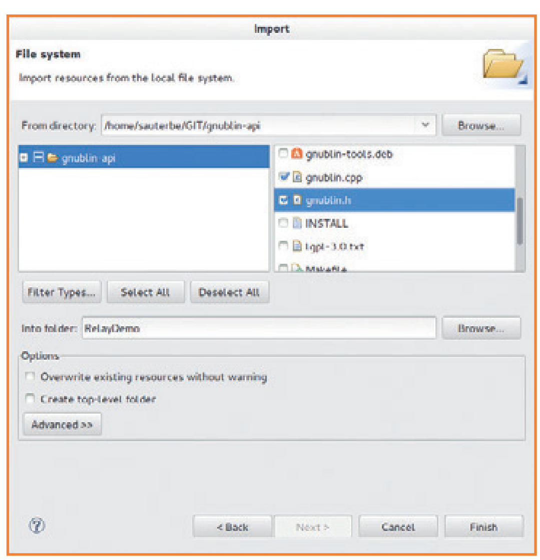


Figure 7.
The files 'gnublin.cpp' and 'gnublin.h' contain functions that greatly simplify application development.

Listing 1: Relay Demo

```
#include "gnublin.h"

int main (int argc, char **argv)
{
   gnublin_gpio gpio;


   gpio.pinMode(18,OUTPUT);


   while(1){

     gpio.digitalWrite(18,HIGH);

     sleep(2);

     gpio.digitalWrite(18,LOW);

     sleep(2);

   }
}
```

to put the SD card into a card reader attached to the PC and copy the file across. However, especially with larger programs that will undergo a lot of tests and changes, this approach will soon become tedious. It is much more convenient to transfer the program using a network connection.

First we have to set up the network connection at the Linux board end. This can be done using its RJ45 wired connection, as described in the previous article in this series [6]; further information can be found at [7].
Even simpler is to use a wireless network (WLAN), using one of the WLAN sticks recommended for use with the Linux board such as the Asus WL-167g V3 (see Figure 2).
First connect to the board using picocom as before, and on the board execute the following command:

`gnublin-wlan -s` **`networkname`** `-k`
`wlanpassword` `-t dhcp`

The name of the network and the password must not include any spaces or special characters.
If it is desired to bring up the network connection automatically when the board is powered up the above command can be added to the file '/etc/rc.local'. Open the file using nano and add the command before the line that reads 'exit 0' (see **Figure 8**).
To test whether the board is connected to the Internet, use a 'ping' command as described previously:

```
ping www.google.com
```

If the gnublin-wlan tool does not work (this was the case, for example, with the *Elektor* guest WLAN which uses WPA security), further help can be found at [8].

Open an SSH console

When the board is connected to the network, its IP address can be determined using the 'ifconfig' command.
Now we want to use the network to establish a connection from the PC to the board, using SSH. First we have to set the board's password. On the Linux board type

```
passwd
```

and then enter the same password twice. Once this is done we can connect from the development PC to the board using the following command (replacing 192.168.0.190 with the IP address of your board):

```
ssh root@192.168.0.190
```

The first time you do this, type 'yes' to confirm the board's fingerprint. Then enter the board's password that was set up above; you should now see a console prompt at which you can enter commands for the board.

How to transfer a program

First, on the development computer, switch to the directory where the compiled program is located

```
cd ~/workspace/RelayDemo/Debug
```

and then copy the executable file from there to the Linux board:

```
scp RelayDemo root@192.168.0.190:/root
```

This will again require you to enter the SSH password you chose previously. The file will now

Create your own SD card

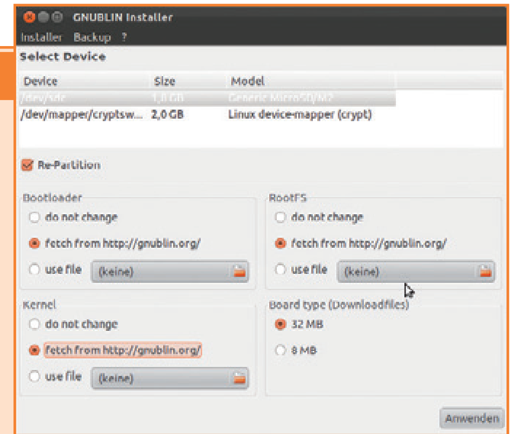
The easiest way to create a new SD card is with the ‘Gnublin Installer’ tool. You will need a Linux PC on which this tool can be installed, an Internet connection (to download the required image files) and an SD card reader. It is possible to use a virtual machine instead of the Linux PC, as long as it uses an Ubuntu or Debian distribution.

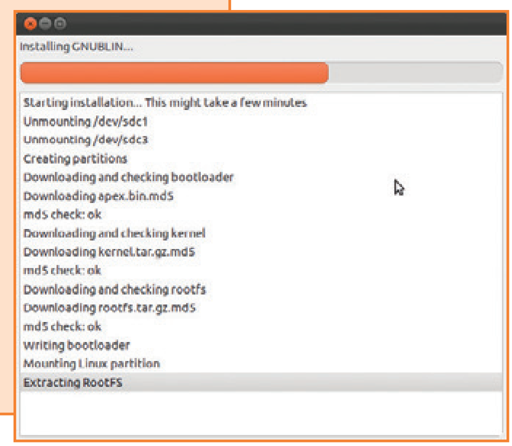
First go to the website at [11] and download the relevant package (amd64 version for 64-bit systems, i386 for 32-bit systems). Double-clicking on the package file should trigger the installation process. Once the installer is installed it can then be run from the console with super-user privileges (having ensured that the right SD card is inserted in the reader):

```
sudo gnublin-installer
```

Now select the SD card from the list under ‘Select Device’ and select ‘fetch from http://gnublin.org’ for all the options below. Later you can select your own images here.

Now click on ‘Apply’: the process will take a little while, depending on the writing speed of your SD card. The log window shows what is happening in detail.





appear within the file system on the Linux board. Connect to the board again using a console and switch to the directory into which we copied the file:

```
cd /root
```

The program can now be run with:

```
./RelayDemo
```

You should be able to hear the relay on the Linux board clicking every second.

It can be inconvenient to have to enter the SSH password every time you want to copy a file across to the board, and it is possible to avoid this by exchanging keys between the development computer and the Linux board. More about how this is done can be found at [9].

Auto start

There are two ways to have an application start up automatically when power is applied to the board: the simpler but less elegant way, and the more complicated but neater way.

The simple method is to add the command to launch the program to the file ‘/etc/rc.local’ before the line that reads ‘exit 0’ (as we did above for the gnublin-wlan tool). Since the program should be started in the background, you need to add an ampersand symbol to the end of the command:

```
/root/RelayDemo &
```

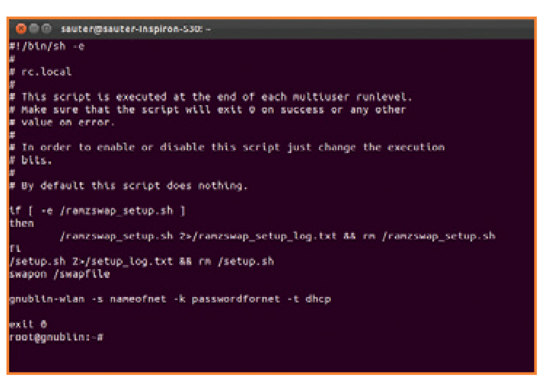


Figure 8.
You can add commands to the file ‘/etc/rc.local’ to cause programs to be run when the board starts up.

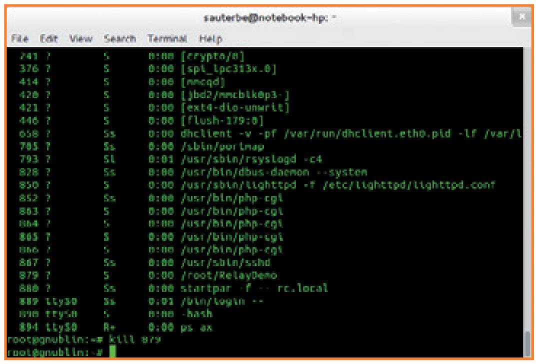


Figure 9.
Applications running in the background can be stopped if you know their process ID.

It is important to give the full absolute path to the program. To stop the program after it has been launched, you need to determine its process ID. If, while the program is running in the background, you type the command

```
ps ax
```

at the console you will see a list of processes including, somewhere, the RelayDemo program. There is a number at the beginning of the line which is the process ID. The command

```
kill ProcessID
```

will stop the program (see **Figure 9**).
The more sophisticated approach to having a program start up automatically is to use a start/stop script. These allow a program to be started and stopped conveniently from the command line. The technique is described in more detail in the wiki [10].

(130298)

Internet Links

[1] sauter@embedded-projects.net
[2] www.elektor.com/gnublin
[3] www.elektor.com/120180
[4] www.elektor.com/120146
[5] www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/junosr2
[6] www.elektor.com/130214
[7] http://en.gnublin.org/index.php/GNUBLIN-LAN
[8] http://en.gnublin.org/index.php/WLAN
[9] http://en.gnublin.org/index.php/Ssh_administration
[10] http://en.gnublin.org/index.php/Application_autostart
[11] http://en.gnublin.org/index.php/Gnublin_Installer

No more broken file systems!

A Linux application should start up, even after a power cut, as reliably as a program stored in flash memory in any embedded system. In the first version of the Elektor Linux board we used the ‘ext2’ file system on the SD card. Unfortunately this file system is not very robust against damage in the event of unexpected power loss. The new version of the Elektor Linux board uses the ‘ext4’ file system which, if there is a power cut or if the board is simply unplugged without being shut down properly, is able to restart as normal.

Users with older memory cards are therefore encouraged to create a new SD card image using the Gnublin Installer (see text box). This works with all Elektor Linux boards, regardless of whether they have 8 MB or 32 MB of memory.

Simple Current Pulse Generator

An aid for checking wide-range current measurements

Time-dependent *voltage* measurements are easy to take, with just a test probe and an oscilloscope. These days signals from DC up to a few tens of MHz can be viewed and analyzed with this setup with no difficulty. Making *current* measurements like this is a bit more difficult, however. A pulse generator is very useful for verifying measurements made with shunts or current clamps. This article describes a simple current pulse generator and shows how you can use it to test and make comparisons between various shunt setups.

By **Martin Ossmann**
(Germany)

A simple pulse generator

For checking your own measurement equipment you need a pulse generator that can deliver powerful current pulses with small rise times. To this end we employ the MOSFET gate driver MCP1407 in the circuit shown in **Figure 1**.

This enables us to turn a normal TTL square wave generator into a current generator. The driver (in DIL form factor) delivers a current of 500 mA continuously, with short pulses of up to 2 A creating no problems. The specified rise time is 20 ns, with an internal resistance of around 2.5 Ω. Next we will see how relatively wide-range current measurements are taken.

Measuring with shunts

First we will investigate how we determine current using the classic shunt method. The shunt is constructed from five 1 Ω metal film resistors connected in parallel. To deliver the small signal to the scope without noise effects we use a 50 Ω coaxial cable that is terminated at each end with a characteristic impedance of 50 Ω (**Figure 2**). In the process the shunted signal is attenuated by a factor of 2.

The signal generator is now used to produce a pulse of 1 A into a resistance of 10 Ω. To achieve this we adjust the power supply to around 12 V, so that the voltage pulse shown on the oscilloscope has an amplitude of 10 V. In series with our load resistor is the shunt of 0.2 Ω (**Figure 2** and **Figure 3**). At 1 A a figure of 200 mV is dropped, whereby a 100 mV square wave pulse should be seen on the scope.

In **Figure 4** the voltage flow is shown in blue and the shunt signal in red. The rise time of the voltage pulse is clearly less than 20 ns. The overshoot is relatively minor.

The corresponding shunt signal is not square

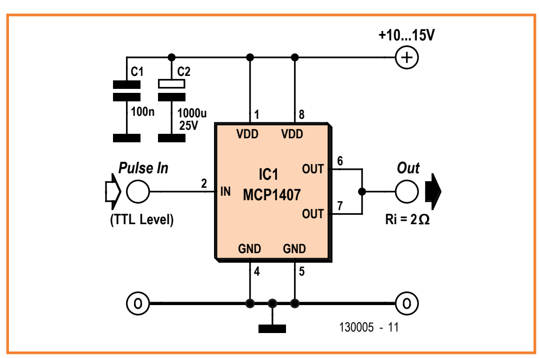


Figure 1.
Pulse amplifier for current pulses.

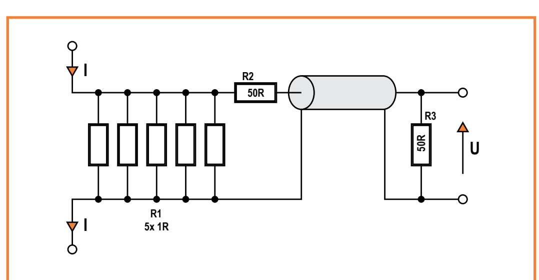


Figure 2.
Circuit for current measurement.

wave in form, as expected. It overshoots only above 200 mV and also displays clear RF disturbance. The overshoots arise because the rapid

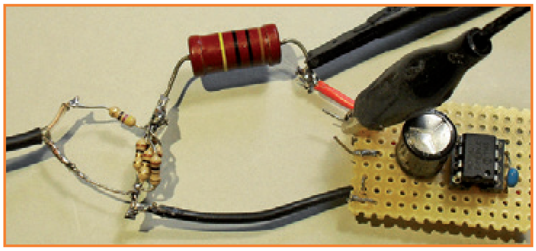


Figure 3.
Current measurement with shunt, as in Figure 2.

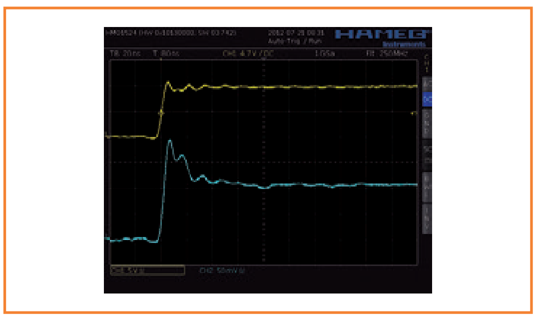


Figure 4.
Signal measurements from the circuit in Figure 3.

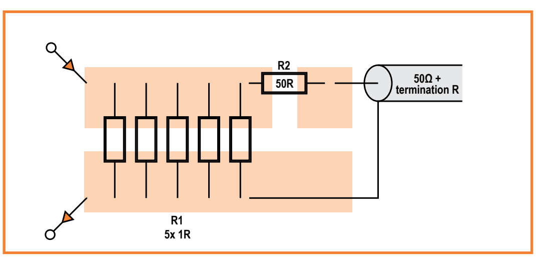


Figure 5.
Layout for low inductive effects.

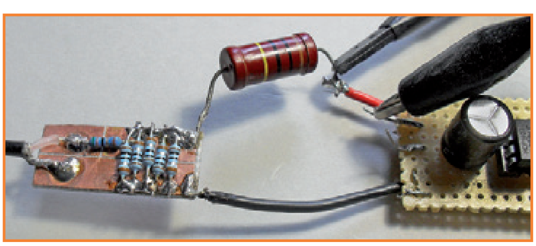


Figure 6.
Practical arrangement of the circuit in Figure 5.

Figure 7.
Signal flows of the setup in Figure 6.

current change brings about a voltage in the loop formed by the 50 Ω resistance and the 50 Ω cable connections.

It's obvious that we need to reduce the inductance in this setup. For this we make two narrow gaps on the copper side of a scrap of printed circuit board, as shown in **Figure 5**. The resistors are soldered direct to the surface (**Figure 6**).

The result shown in **Figure 7** illustrates how the measurement is now improved significantly.

Next we'll do something about the RF interference by feeding the 50 Ω cable twice through a clamp-on (clamshell) ferrite choke. You can see this on the left side of **Figure 8**.

In this way we achieve a definite attenuation of RF disturbance. The curve shape in **Figure 9** is now acceptable.

Using clamp-on ferrite chokes effectively

At this stage it's worthwhile explaining why a clamp-on choke helps in this situation. This is important, because in this way you will recognize when these chokes can improve the accuracy of measurements.

Take the state of affairs in **Figure 10**. A voltage source produces a 10 V voltage pulse with a rapid rise time. The voltage generates in resistor R1 (10 Ω) a current that amounts to 1 A in steady state. This current flow is shown in red. The voltage pulse is measured using a 1 MΩ test probe and oscilloscope. The screened cable connection is shown at the bottom of Figure 10. Resistor R2 is the current measurement shunt; the voltage reaches this over a screened 50 Ω cable, also connected to the oscilloscope. A so-called ground loop (green) is formed by the two cable screens. This creates a problem in that the red circuit and the green ground loop share one piece of wire (shown blue) in common. The ohmic resistance is often relatively low and hence not the culprit, but even 1 cm of wire has a typical inductance of 10 nH (roughly). The voltage pulse has a rise time of around 5 ns. Ideally the current should then climb too from 0 to 1 A in around 5 ns. This meteoric current change induces a voltage of

$$U = L\,(\Delta I\,/\,\Delta T) = 2\ \text{V}$$

in the cable's low inductance. The induced voltage generates a current flow in the green loop. In turn this current generates voltages in the screening shield of the cables, which augment the original signal. With rapid pulses these voltages can be significant, particularly when compared with the minute voltages across shunt R2.

Figure 11 illustrates the situation with a ferrite choke. The two conductors of the screened cable carrying the signal current are wound in the same direction through the choke. This creates a small inductance in the screening. At higher frequencies this represents significant resistance. In this way the ground loop is virtually broken, with the voltage induced in the blue wire falling on account of this inductance. Since the core of the cable is wound through the choke exactly as the screening is, the induced voltage is created in both conductors, meaning that the differential voltage (our current measurement signal) does not change. Effectively the ferrite operates as a common-mode choke.

Measurements taken with an oscilloscope frequently employ several test probes with ground connections, so it generally makes sense to use a clamp-on ferrite choke. The measurements will then be more accurate, if RF voltages (ground bounce) exist between the various (nominally) ground connections.

Low-inductance resistors

In this article we have discussed tests using a pulse voltage source (10 V) with a resistor, in order to produce square wave current pulses. However, there are further pitfalls lurking in the details. Figure 8 shows a setup with a 10 Ω/50 watt power resistor. The relevant voltage and current flows are illustrated in **Figure 12**.

Here the current takes about 500 ns to achieve a final value of 1 A in steady state. With such a slow rise time it is naturally impossible to characterize current measurements. What's the reason for this? Well, the power resistor is wire-wound and thus has an inductance of around 2.5 µH. This parasitic inductance inflicts the slow rise time. To achieve faster rise time it is absolutely essential to employ low-inductance resistors (for example from Caddock). The circuit layout must also be designed for low inductance, using the smallest circuit elements possible. Over time you gain a feel for this problem, by the way.

The scenarios described show how you can use the current generator to build and check shunts that work reliably for current measurement over wide frequency ranges. Once tested, they can be put into service, for example for developing switch-mode power supplies.

(130005)

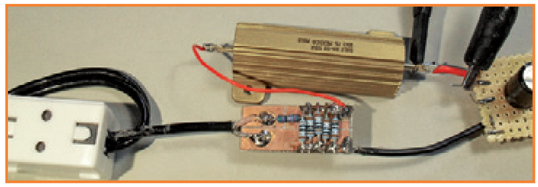


Figure 8.
Fitting a ferrite choke onto the 50 Ω cable.

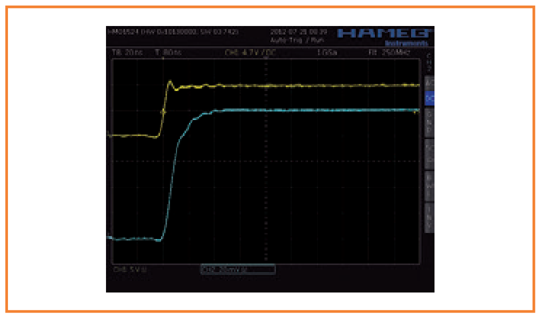


Figure 9.
Signal with ferrite choke installed.

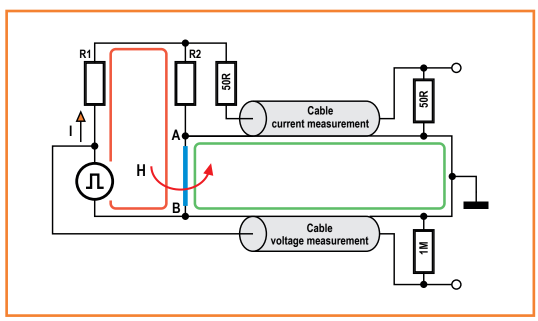


Figure 10.
Inductive coupling in the ground loop.

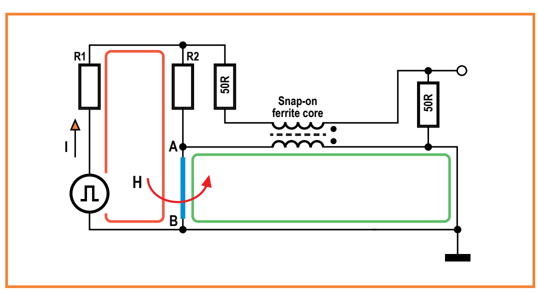
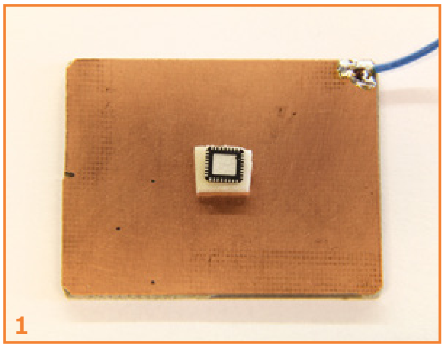
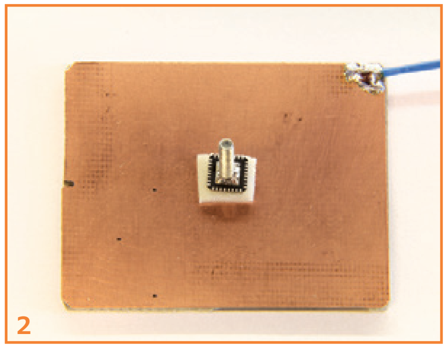
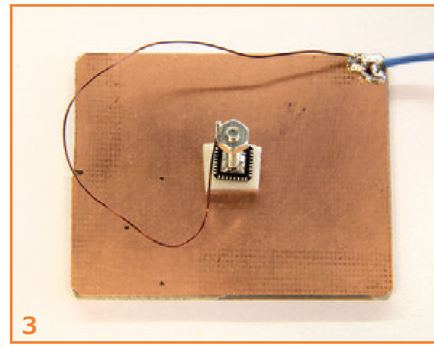


Figure 11.
Effect of ferrite choke.

Figure 12.
Signals in a resistor with parasitic induction.


1


2


3

Soldering LFCSP ICs by Hand

By **W.T. Knoeff,**
PA3EHN (Netherlands)

Lead Frame Chip Scale Package (LFCSP) integrated circuits are devices with dimensions that are apparently determined by the required number of connections. Some of them are only a few square millimeters in size. For the connections to the outside world, there are small contact pads on the bottom along the four sides, each measuring 0.6 x 0.25 mm with a spacing of 0.4 mm. A small ground plane is located in the middle. Soldering these ICs to a PCB by hand can be very difficult. In addition, with these devices you have to be especially careful to avoid electrostatic discharges, which can have fatal consequences.

In this article we describe an alternative mounting method for these ICs. It consists of first soldering tiny copper wires to the individual contact pads, and then mounting the IC on a regular circuit board.

The reason for all of this is that the author wanted to build an RF signal generator using the AD9913 DDS from Analog Devices (to be described in the March 2014 edition), but he did not have any way to properly solder this tiny LFCSP IC (with 32 contact pads on the bottom in a 5 x 5 mm square array around a ground plane) to a printed circuit board. All sorts of things could go wrong during soldering, such as a short somewhere under the IC. The ground plane is also practically inaccessible; how are you supposed to solder that?

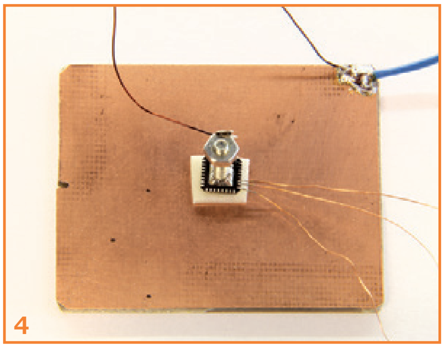
Faced with this problem, the author came up with the idea of first soldering pieces of wire to the contact pads to turn them into leads that could be soldered to the PCB in the usual manner. Another advantage of this method is that you don't need any closely spaced, super-thin and easily damaged tracks on the actual PCB.

What do you need for this operation? A few simple things: some double-sided adhesive foam tape, a length of coaxial TV cable with a braided shield, super glue, a 2-mm (or number 2) screw made from solderable material, a sturdy magnifier lamp and a soldering iron with a fine tip. Of course, you also need a steady hand and a good dose of patience.

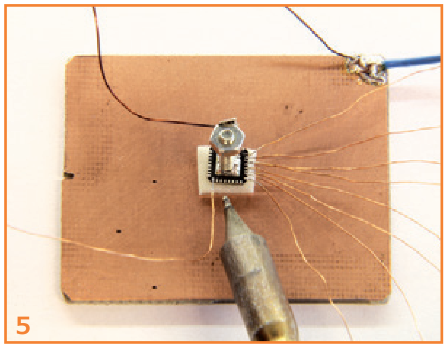
To avoid problems with electrostatic discharge, you should use a properly grounded soldering iron (if necessary, you can improvise a ground connection to the outer surface of the iron) and provide good grounding for your worktop and your own body.

Use a piece of virgin PCB material as a work surface, and ground it as well. Stick a small piece of double-sided adhesive foam tape onto the PCB material and then stick the IC upside-down onto the tape, with its contact pads facing up. The adhesive is strong enough to hold the IC securely during the upcoming soldering work, even if it gets really hot for a short time. If necessary, use a magnifier lamp to help you work precisely (**Photo 1**).

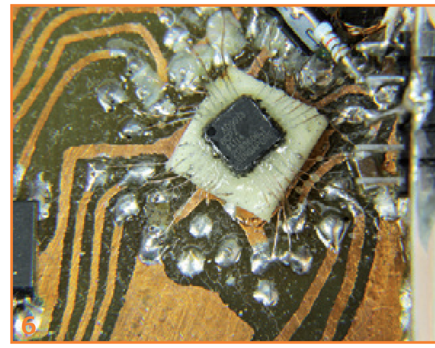
Start by tinning the central ground plane. With the soldering iron in one hand and a length of thin wire solder in the other hand, apply a bit of solder to the ground plane. Saw off the head of



4



5



6

the 2-mm screw, tin the end of the remaining threaded section, and solder it vertically onto the IC in the middle of the ground plane (**Photo 2**). Later this screw will be used to mount the IC on the PCB.
Now ground the IC by connecting a small wire between a nut on the screw and the grounded copper sheet of your work surface (**Photo 3**). Solder the wire to the nut before threading it onto the screw, as otherwise the screw might come loose from the ground plane.
The next step is to tin all of the contact pads. You can rotate your work surface as necessary to enable good access to all four sides with your soldering iron and solder.

Now comes the most difficult part: soldering new leads onto the tinned pads. For the leads, you can use copper wires (bare or tinned) taken from the shield braid of a piece of coax cable. These wires have roughly the same diameter as the contact pads (you might have to strip several different types of coax cable, since the diameter of the braid wires can vary considerably). This may sound like rather fiddly work, but it's actually fairly easy once you get the hang of it. First tin the tip of a wire with solder paste. While holding the soldering iron in one hand, place the end of the tinned wire (with the solder paste) on one of the tinned contact pads with your other hand. Touch the tip of the soldering iron alongside the junction of the wire and the contact pad for half a second—just long enough to let the solder on the pad and the wire melt together. Wiggle the wire a bit afterwards to check that it is firmly attached. **Photo 4** shows an IC with several leads soldered this way.
After you have soldered leads onto all the pads on one side, you can secure the ends of the leads to the work surface with electrician's tape. Then turn the work surface (PCB material) by 90 degrees and continue with the next set of leads (**Photo 5**; the electrician's tape is missing here). When all four sides are done, you can use a small needle to scratch off any extra solder paste around the pads. Then carefully loosen the IC from the double-sided adhesive foam tape, together with the leads.
To reinforce this arrangement, slide a piece of double-sided adhesive foam tape with a 2-mm hole in the middle over the ground plane screw and press it against the bottom of the IC with its new leads. Squirt a bit of super glue between the IC and the foam tape to form a permanent bond between the tape and the IC. That makes it less likely that the wires will come loose later on when you mount the IC on the board. The elasticity of the foam tape also cushions any bumps on the bottom of the IC due to the solder on the pads and around the screw. The other side of the adhesive foam tape comes in handy later on when you mount the IC on the board.
All in all a rather complicated task, but it's worth the effort.

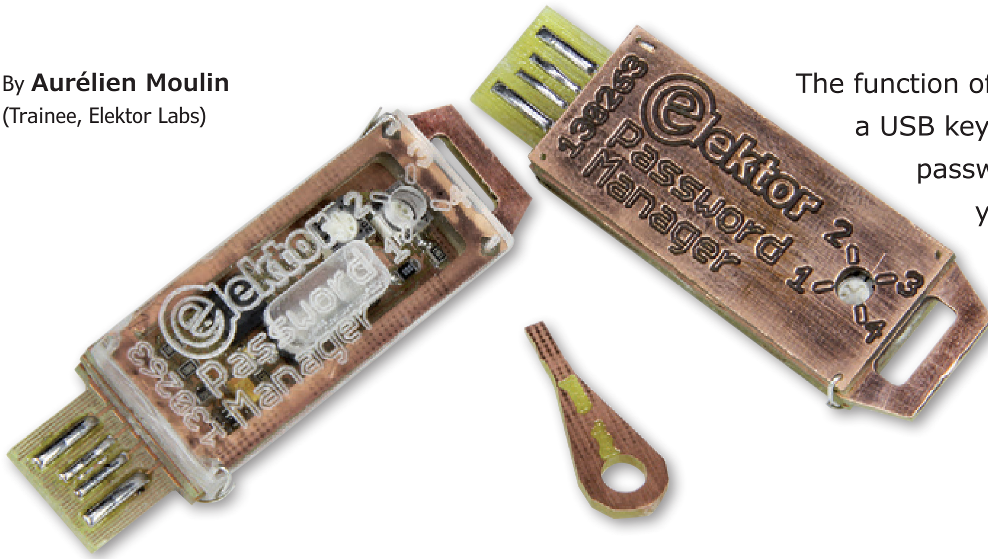
The PCB pads for the leads should be arranged in a square around the IC. If the IC has lots of leads, the pads can be staggered in a double row. Drill a 2.5-mm (0.1 inch) hole in the middle of the IC position for the mounting screw. Apply thermal paste and insert a copper strip in the hole to connect the ground surfaces on the top and bottom sides of the board.
Then fit the IC on the board and screw it tight (but not too tight) with a small nut. The screw connects the IC to the ground plane of the PCB and provides adequate heat dissipation.
Finally, solder all the wire leads to the pads on the PCB (**Photo 6**).
The DDS signal generator designed around this IC will be described in the next edition of Elektor.

(120246-I)

USB Key Passport

Secure Password Manager

By **Aurélien Moulin**
(Trainee, Elektor Labs)



The function of this passport in the form of a USB key is to automatically type a password (up to 30 characters) for you. A selector lets you choose which of the four passwords stored on the key will be entered just as if it came from a keyboard.

For about a year now, Elektor has been setting aside each month two unpublished articles (on paper that is) especially for its members, who can download them from the www.elektor-magazine.com website. Project #10 [1] was a **Keyboard Simulator** in the form of a USB key, using an ATtiny85 that types a pre-recorded password automatically when a certain key combination is entered.

This project is based on the superb V-USB library developed by obdev.at [6] for adding the USB V1 standard (referred to as 'low speed') to all types of AVR microcontrollers. Here it is being used to implement a USB HID (Human Interface Device) keyboard.
The interest aroused by this project gave us some ideas for improvements that were so persuasive we are offering here a new and considerably improved version.

The principle of storing a password on a USB key and being able to enter it automatically is appealing, but for it to be practical, we need to be able to put several passwords on the same key. By the same token, it must also be possible to modify them easily without having to reprogram the key's microprocessor. So it was also necessary to develop a protocol for communicating with the key and an application to read and modify the passwords.

One thing leads to another...

For anyone familiar with the original project, the update also provided an opportunity to optimize cost and quality. This consists in reducing the size (SMD) and abandoning the metal USB A plug, which proved not to be mechanically very robust. The new key's USB connector is formed directly by the copper tracks of its own PCB. While I was about it, I changed the microcontroller to an ATtiny45 (Flash: 4 KB, RAM: 256 Bytes, EEPROM: 256 Bytes).
Unable to find a case that was really small and yet sturdy enough, I had the idea of creating one out of PCB elements, stacking them up onto the PCB proper, as shown in the photos. The various layers of PCB forming the case are held together by tinned copper wire threaded through the fixing holes and then soldered at each end.
We'll come back to the actual construction once we've taken a closer look at the (new) circuit diagram and its software (**Figure 1**).

Ergonomics

On the first version of this project, to tell the USB key (which acts as if it were a keyboard) to send the single stored password, we used a

combination of the Num Lock, Scroll Lock, and Caps Lock keys. Example sequence:
1. Num Lock → 2. Num Lock + Caps Lock → 3. Num Lock, following which the key sends the password to the PC as if it had been typed in from the keyboard.
For the 4-password version, I had the idea of selecting the password to be entered directly on the key, using a 4-position selector. Since the only pin available on the PIC ($\overline{RST}$) is also, quite by chance, an analog/digital converter (ADC0), by way of a selector I've added preset P1, configured as a voltage divider. Depending on the position of P1 wiper – at the start of its travel, at 30 %, at 60 %, or at the end of its travel – one of the four passwords will be entered by the USB key when it is inserted, just as if this password came from a keyboard. However, the existing function to call up the passwords by a key combination is still possible.

To enable ADC0, we must disable the PIC's reset function by setting its RSTDISABLE fuse to 1.
P2 is the calibration potentiometer. The aim is to obtain a good measuring range for the 10-bit ADC with three well-defined thresholds, despite component tolerances. Since ordinary potentiometers have a tolerance of around 20 %, while R5 is 1 %, and as a more accurate potentiometer would be too expensive, P2 is used to compensate for the variations in P1. The values are set in the firmware, and the hardware is adapted to them. I haven't changed anything else from the original circuit. Let's come back now to the board (**Figure 2**).

Production of this single-sided PCB has been possible thanks to the Elektor PCB prototype [2], a superb digitally-controlled engraver specializing in PCBs (see also my article on this subject in the November 2013 issue [3]).
To make production easier, I've brought all the different elements together on a single board, including the key itself and even the locking hole. The only thing missing is the wrist-strap or key-ring.

The tongue that acts as the USB plug is long enough to be able to use it even with the most hard-to-reach USB ports (where the computer case stops you plugging in a normal key).
In the hollowed-out part of the two rectangles of PCB material (**Figure 3**) forming the center

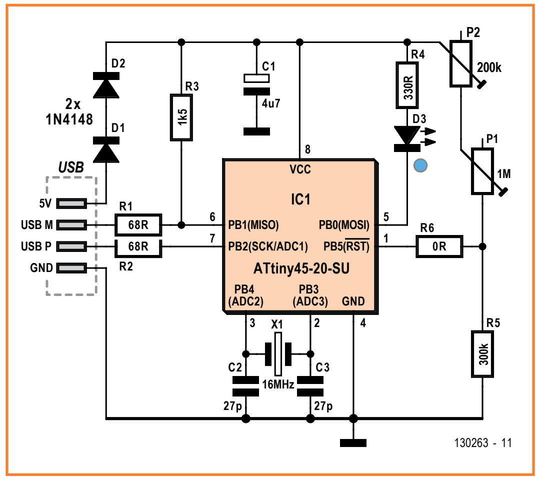


Figure 1.
Schematic of the USB Passport, with preset resistor P1 taking the place of a selector switch.

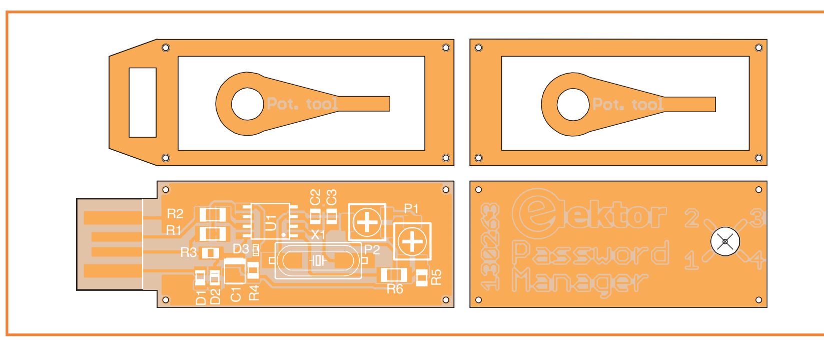
COMPONENT LIST

Resistors
(1%)
R1,R2 = 68 Ω
R3 = 1.5kΩ
R4 = 330Ω
R5 = 300kΩ
R6 = 0Ω
P1 = 1MΩ adjustable
P2 = 200kΩ adjustable

Capacitors
C1 = 4.7uF 25V
C2, C3 = 27pF 100V 5%, ceramic

Semiconductors
D1,D2 = 1N4148
D3 = LED, blue, SMD
IC1 = ATtiny45-20SU, programmed

Miscellaneous
X1 = 16MHz quartz crystal



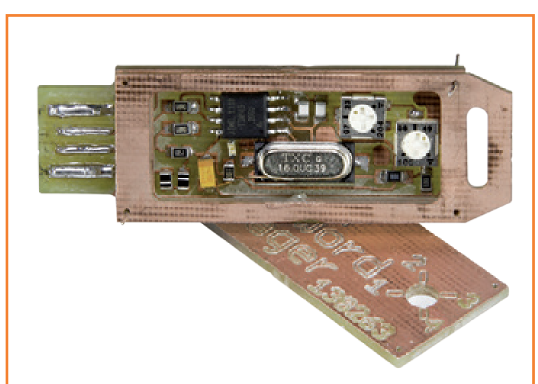
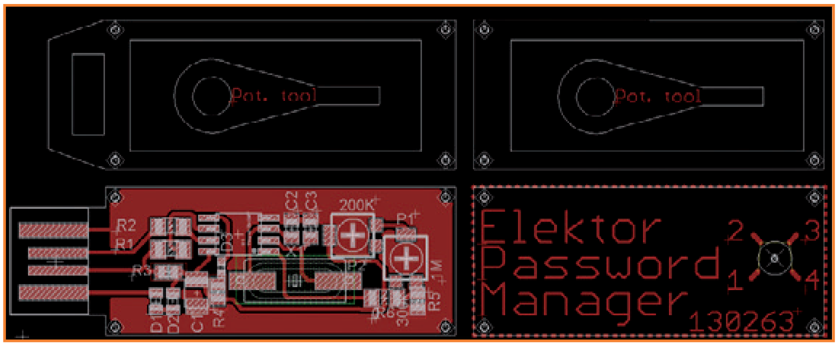


Figure 2.
The (rather unusual) design for the USB Passport PCB.

Figure 3.
The PCB elements transform into a case.



of the case, I've added some tools in the shape of screwdrivers that will let you adjust the password-selector potentiometer. All you have to do is taper the end slightly so it fits in P1 slot.

While I was at it, I've tried to make up for the not very intuitive (but otherwise excellent) functions of the PCB MODULE software that controls the CNC by improving the documentation [4]. Until I set about it, no-one had so far tried to make the most of cutting-out arbitrary shapes. After several tests, I've set up a tutorial to guide users from creating the PCB in Eagle right up to the cutting-out part using PCB MODULE.

This experience has shown me that a satisfactory compromise has to be found between the number, position, and size of the essential breakout areas. There must be sufficient to withstand the torsion caused by the milling cutter while machining, but not too many, so that stripping-out remains easy and finishing fast (**Figure 4**).

Software side

This project includes two complementary elements: the ATtiny 45/85 firmware and the software for managing the passwords on the PC. These communicate via an asynchronous protocol based on the state of the three lock keys: Num Lock, Scroll Lock) and Caps Lock. The great advantage of this protocol is the absence of drivers, other than the generic HID driver for recognizing the USB key and its ATtiny as a keyboard. In order to be reliable, the communication speed must be slow. In fact, it does not exceed about 10 baud, which is acceptable given the small amount of data to be transferred and how infrequent the reprogramming operation is.

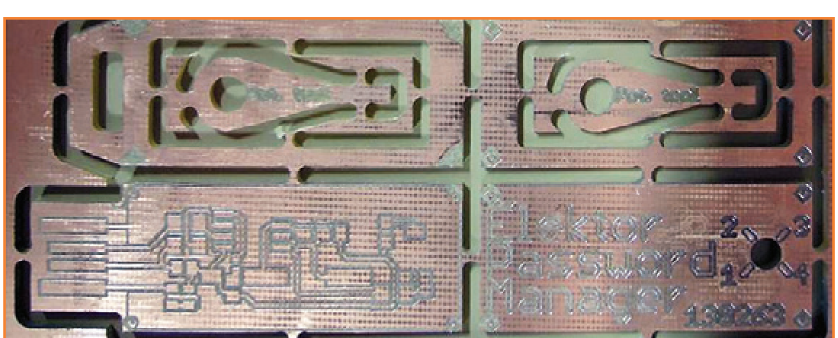


Figure 4.
All the elements of the circuit and case brought together on the same sheet.

The communication between the USB key and the host PC is entirely based on sampling and memorizing the states (current and previous) of the lock key status LEDs. The default mode is read mode, in which the password manager waits to receive a valid combination. As soon as this is the case, it enters the corresponding password and goes back into standby.

The *write* mode can be accessed using a special combination sent by the host software. In this mode, bytes are reconstituted from the bits received then written into the internal EEPROM. The program detects the ends of strings and manages branching in the memory, as well as *write* mode exit; once the four passwords have been received, it returns to *read* mode. In this new version, the program performs this reading and writing of the passwords in the background. So it is still possible to use the key without the potentiometer.

The dump from the ATtiny45 EEPROM shows the four demonstration passwords Password1, Password2, Password3, and Password4, along with their string terminators: \0 (0x00 in memory) (**Figure 5**).

Firmware

In order to determine the password the user has selected via P1, the IC1 firmware reads the value delivered by ADC0 and compares it with the predefined reference thresholds. Then, as soon as communication has been established with the host PC, the characters of the corresponding password are entered as if they came from a keyboard. It's simple, and yet this modest modification to the original version gave me terrific headaches. During the early testing, certain passwords stored in the EEPROM were erased after reading (!), even in the absence of any write operation. On certain prototypes, the error only occurred once the circuit was fitted into its case (stray capacitance? EMC problem?) Neither adding a ground plane nor grounding the crystal case cured the problem, and my measurements revealed no faults. A mystery…

My suspicions then turned to the analog/digital converter, which runs at its own frequency (125 kHz). In order to space the operations of initializing and reading the ADC out in time from the rest of the program, and in particular the reading of the EEPROM, I inserted some 10 ms delays into the software and… the problem of erased passwords disappeared as if by magic.

Host software

The Elektor Password Manager software written in C# [5] takes care of the interaction with the Password Manager key. This makes it possible to read and write passwords into the microcontroller EEPROM. Its interface is sparse (**Figure 6**): four text fields, for reading and writing the passwords, and two buttons *Read* and *Write* to read and write the passwords.

The read operation consists simply in emulating, i.e. simulating, the keyboard codes corresponding to the various sequences of characters in the passwords. The write operation is more complex, and involves verifying the strings typed in by the user and refusing the write operation if the strings are too long or rejecting those containing invalid characters. If the string is valid, the write operation proper can start: the characters are broken down into bits, then transmitted to the key in the form of signals corresponding to the status of the Caps Lock, Scroll Lock, and Num Lock keys. A synchronizing signal frames each byte transmitted.

Watch out! Because of the constraints of keyboard regionalization (*QWERTY, AZERTY*, etc.), only alphanumeric characters are accepted. It would be easy for you to modify the code so it will accept special characters. In read, there is a similar constraint: it only works if the host system is configured to use a QWERTY keyboard (EN/US regional settings).

To encourage users to use passwords that will resist any attempt to analyze, I've added to the application a generator of random strings, whose length you can set.

An interactive guide for calibrating the P1 thresholds makes it easier for users to get started, through clear instructions and detailed diagrams. All the files for building this project using the PCB Prototyper are also available on our members' website [5].

Password security and robustness

The passwords stored can have up to 30 characters. With the 62 alphanumeric characters recognized (lowercase, uppercase, figures), the time an attacker would take to guess the password by going through all the possible combinations increases at a dizzying rate with the number of

```
:10000000506173737776F72643400FFFFFFFFFFFF6F
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:10002000506173737776F72643300FFFFFFFFFFFF50
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000506173737776F72643200FFFFFFFFFFFF31
:10005000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:10006000506173737776F72643100FFFFFFFFFFFF12
:10007000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:10008000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF80
:10009000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF70
:1000A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF60
:1000B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF50
:1000C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF40
:1000D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF30
:1000E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF20
:1000F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF10
:00000001FF
```

Figure 5.
Memory dump with the four passwords.

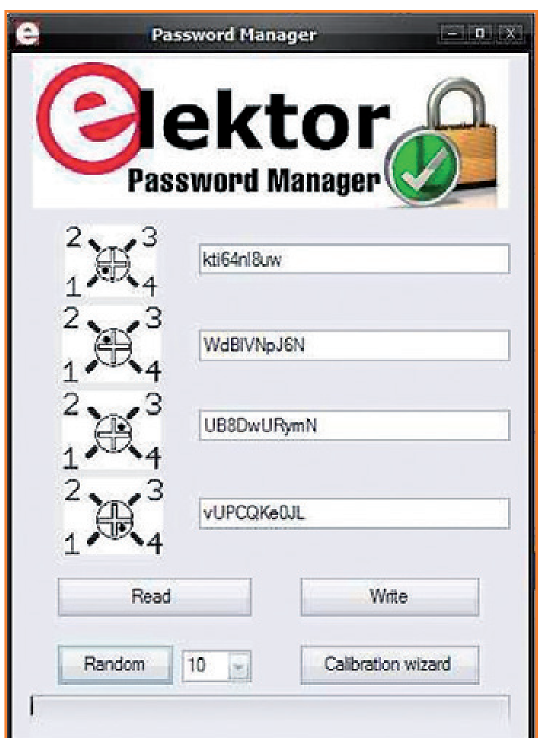


Figure 6.
The software's graphical interface

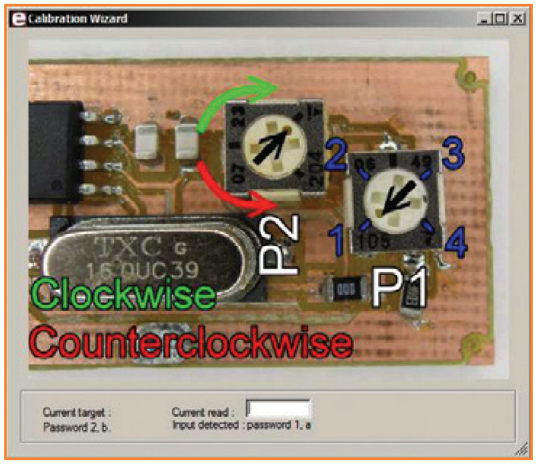


Figure 7.
Graphical interface for the calibration guide.

characters actually used in the password. Eight is a minimum, but it is recommended to use longer passwords. Now the advantage of this accessory is exactly that it allows you to use passwords that are not only long (you won't have to type them by hand each time!), but above all made up of random strings, which are proof against dictionary attacks, unlike short, easy-to-remember passwords that are too easy to decode.
If you lose your USB passport, the passwords it contains will be compromised. To increase security, here are some ideas for possible software counter-measures, which I have not tested:

- Masking or encryption of the EEPROM contents (the ASCII character codes are currently readable).
- Activation of the 'lock' fuse at the time of programming to prevent direct reading of the internal EEPROM.
- Erasure of the EEPROM for invalid combinations. It would be possible to set traps at certain positions of P1 such that if these are selected, the contents of the EEPROM would be erased. For example, position 1 = password 1; position 2 = erase all… On the one hand, this counter-measure would limit analysis and on the other, afford protection in the event of use under duress.

And lastly two hardware counter-measures: using a photosensitive detector to detect if the case is opened in order to disable the microcontroller; and encasing the circuit in epoxy resin.
That said, even these precautions are not enough to resist a determined attack. Don't forget, too, that although the key does protect you against your own memory lapses as well as against hardware keyloggers inserted between the physical keyboard and the computer, it does not protect against spyware.

(130263)

Internet Links

[1] Elektor.POST no. 10: USB Keyboard Emulator (members only): www.elektor-magazine.com and at www.elektor-labs.com/120583

[2] www.elektor.com/PCBprototyper

[3] PCB Prototyper Master Class: www.elektor-magazine.com/130128

[4] PCB MODULE software tutorial: www.elektor-magzine.com/130263

[5] Software and source code in C# from Elektor Password Manager, PCB Prototyper production files: www.elektor-magazine.com/130263

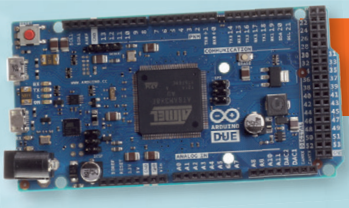
[6] www.obdev.at/products/vusb/index.html

The circuit in 10 ph(r)ases:

1. Turn on host computer and download the program [5].
2. (In the PC's regional and linguistic settings) change the system language configuration of the keyboard to EN/US
3. Run program
4. Plug key in
5. Enter 1, 2, 3, or 4 passwords then click on Write
6. Wait for the operation to end before ejecting and removing the key.
7. Select the desired password using the potentiometer
8. Select the text field where you would have typed your password manually
9. Plug the key in—the selected password is entered automatically.
10. Unplug the key.

10% OFF for GREEN and GOLD Members

Arduino
Now Available @ Elektor!

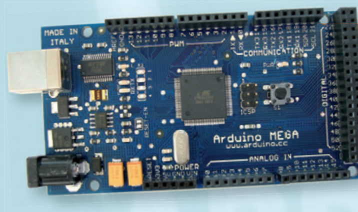


ARDUINO DUE
32-bit power thanks to an ARM processor

Features

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 12 provide PWM output)
PWM Channels	12
Analog Input Pins	12
Analog Outputs Pins	2 (DAC)
DC Current per I/O Pin	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB (all available for the user applications)
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz

£47.00 • € 52.77 • US $76.50

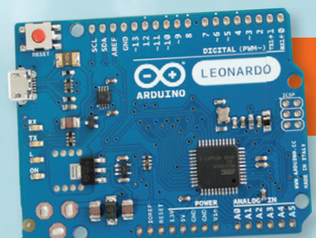


ARDUINO MEGA
Like the Uno but with more memory and I/O

Features

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB (of which 8 KB used by bootloader)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

£47.00 • € 52.77 • US $76.50

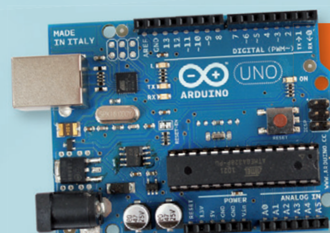


ARDUINO LEONARDO
Especially good for USB applications

Features

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20 (of which 7 provide PWM output)
PWM Channels	7
Analog Input Pins	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

£22.10 • € 24.81 • US $36.00



ARDUINO UNO
The most popular board with its ATmega328 MCU

Features

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Channels	6
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

£24.40 • € 27.35 • US $39.70

ARDUINO ETHERNET
Networking has never been easier

Features

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 4 provide PWM output)
Arduino Pins reserved	10 to 13 used for SPI 4 used for SD card 2 W5100 interrupt (when bridged)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

£48.10 • € 53.98 • US $78.30

ARDUINO YÚN
Two processors

Features AVR Arduino microcontroller

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage	5V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

Features Linux microprocessor

Processor	Atheros AR9331
Architecture	MIPS @400 MHz
Operating Voltage	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host/Device
Card Reader	Micro-SD only
RAM	64 MB DDR2
Flash Memory	16 MB

PoE compatible 802.3af card support

£62.30 • € 69.95 • US $101.40

ARDUINO

elektor

Further Information and Ordering at www.elektor.com/arduino

Nanoamps on the DMM

Measure tiny currents using an ordinary digital multimeter

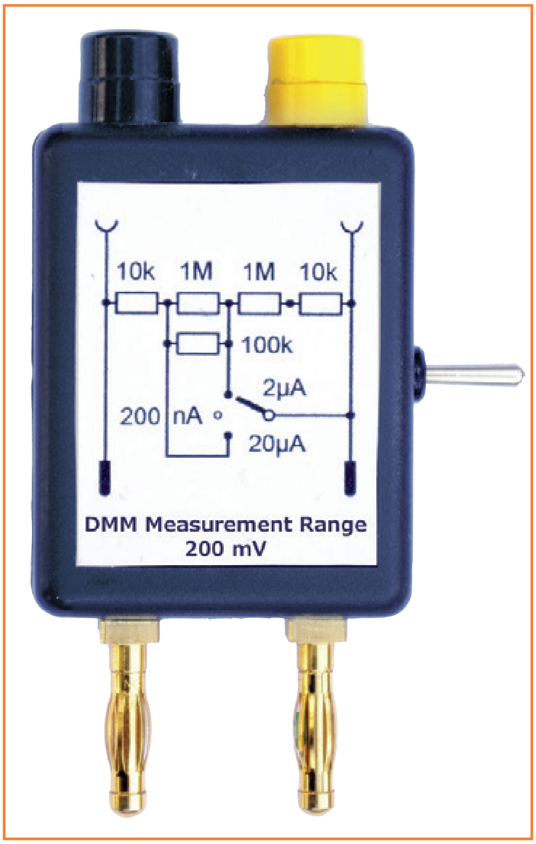


Figure 1.
The “circuit diagram” of the adapter, shown here glued to the base of the author’s prototype.

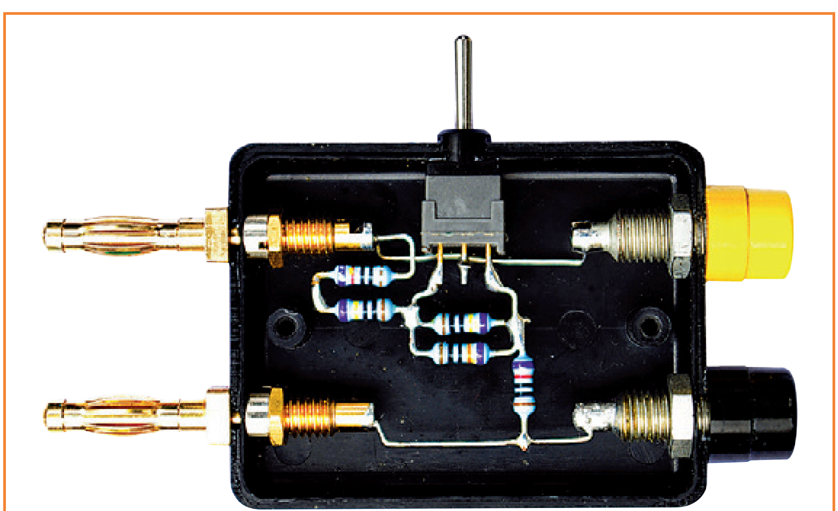


Figure 2.
Looking inside the enclosure reveals the “component mounting plan”.

By **Jo Becker,** DJ8IL (Germany)

Even the most Bargain Basement LCD digital multimeters (DMMs) these days have more than adequate accuracy and range coverage for any normal use. However, sometimes less ordinary situations arise and you’ll find yourself out of luck: for example, a cheap 3.5 digit meter will normally have a lowest current range of 200 µA full scale. What can be done?

Here is a neat trick that lets you measure tiny currents using the lowest voltage range of 200 mV full scale, and it really works! The typical internal resistance of the meter on this range is around 10 MΩ, which means that at a voltage drop of 200 mV a current of 20 nA will flow. This will allow you to measure, for example, tiny photocurrents in photodiodes. Unfortunately, however, this leaves several decades of gap between a full-scale reading of 20 nA and a full-scale reading of 200 µA.

It did not take me long to decide to fill in that gap with a home-made adapter. As the photographs show, all that is required is a small plastic enclosure, a toggle switch with a central ‘off’ position, two 4-mm banana sockets, two 4-mm banana plugs and a few metal film resistors. Together

Component List

Resistors

(0.1% metal film)
2 pcs 10kΩ
1 pc 100kΩ
2 pcs 1MΩ

Miscellaneous

Switch, 1 x change-over with center position
Plastic enclosure
2 pcs banana socket
2 pcs banana plug with threading

these make a very simple yet useful and functional adapter that can be inserted between the meter and the test probes and which provides the three missing ranges.

The photographs show how the circuit is put together. The circuit diagram, shown glued onto the assembled adapter in the author's prototype (**Figure 1**), includes five precision resistors from the E1 series. The switch selects between the 0.2 µA, 2 µA and 20 µA ranges. The clever aspect of the circuit is that it only adds insignificantly to the measurement error of the multimeter.

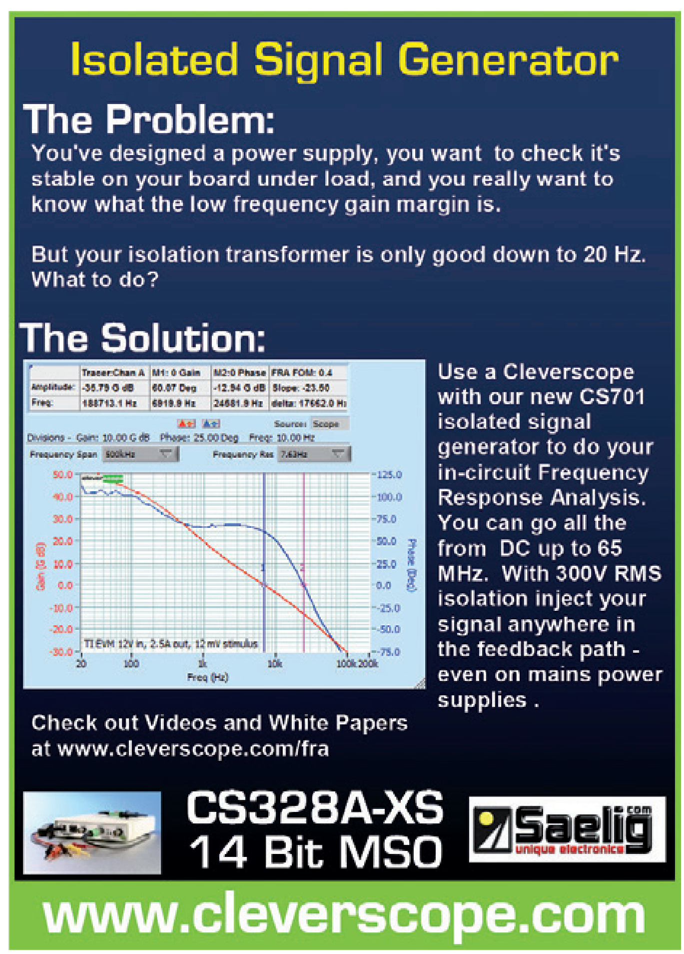
If 0.1% tolerance resistors are used the measurement error of the meter is only increased by 0.2% on the 2-µA and 20-µA ranges and by just 0.116% on the 200-nA range. You can check these surprising results for yourself: don't forget to take into account the 10 MΩ input resistance of the meter.

Construction is straightforward. The 4-mm input sockets on most multimeters have a spacing of ¾ inch (19 mm), but it is a good idea to measure and make sure!. It is then just a matter of drilling suitable holes for the two banana sockets and the two banana plugs on opposite faces of the enclosure, and a hole for the miniature switch in the middle of one of the adjacent sides.

The resistors can be soldered to one another and mounted in mid-air: **Figure 2** shows what might be described as the 'component mounting plan'. Fit the lid and the unit is ready for use.

(130159)

Advertisement



Bird Drinking Water Heater

Using a 2N3055

Many people enjoy feeding the birds in their garden in winter with seeds, peanuts and bits of fat. However, it is often forgotten that these animals also need water. And when it’s freezing, there is nothing for them to drink. This can be solved with an electronically heated water feeder, so that our feathered friends, even with temperatures below freezing, can continue to sip their water.

By **Bas Schmidt**
(Netherlands)

The water is heated with the aid of the most famous power transistor of all time, the 2N3055 (in a TO-3 package). The idea to use a transistor in a TO-3 housing as a heating element has been described in Elektor on earlier occasions, but is still a cheap and effective method. An NTC is used to measure the temperature of a small tin (which contains the water). The current for the heater is supplied by eight rechargeable batteries. The electronics is built around a dual comparator type LM393 and comprises a temperature controller and a power supply voltage monitor, which prevents the batteries from being discharged too deeply. Two LEDs are used to indicate the status of the circuit.

Water feeder construction

The water bowl is made from an empty tomato paste tin, which is mounted on top of a small box. The author originally used a wooden cigar box for this, but you can of course use a plastic box, like the one shown in the photo.
The decision was made to use a **small water bowl**, for two reasons. Firstly, the smaller the amount of water that needs to be heated, the longer the batteries will last. But the second reason is equally important: **The birds must not be able to take a bath in the water!** Because if they do that there is the possibility that their feathers will freeze and that will not end well. The tomato paste tin can contain more than enough water to last a twenty-four hour period. After all, birds only need a small amount of water per day. The cigar box was spay-painted twice with green paint to make it water-resistant. This is not really necessary when using a plastic box, but a green color blends better into the garden of course. The box contains the batteries and the printed circuit board with the electronics. The tin is mounted on the lid, the 2N3055 is bolted on the bottom of the

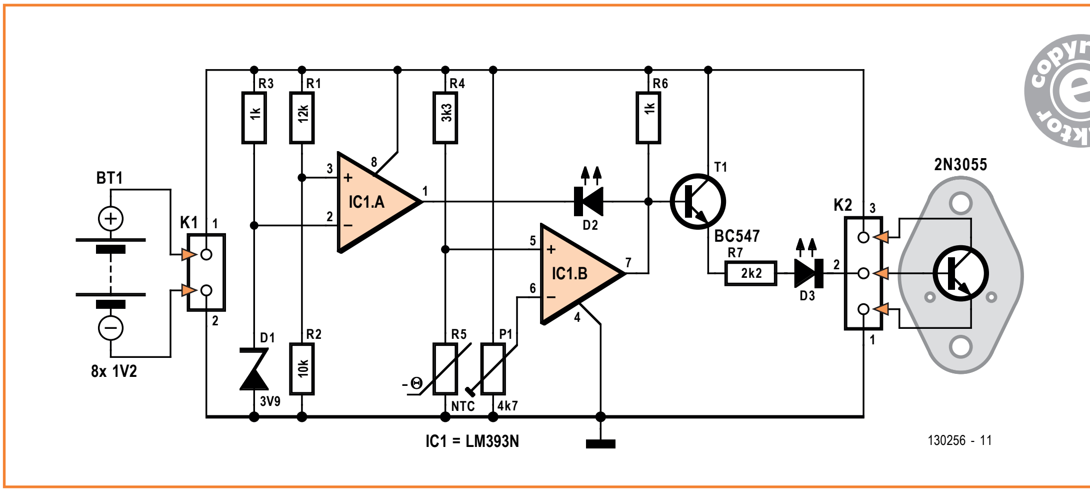


Figure 1.
The circuit consists of two parts: a low-battery indicator around IC1A and a temperature controller around IC1B.

tin and afterwards covered with a small layer of silicon adhesive (since it has to be waterproof, after all). The NTC is glued to the side of the tin with a little silicon adhesive or thermal glue.

Electronics

The schematic in **Figure 1** is a simple design and quickly explained. At the far left we see the power supply, which consists of eight series-connected NiCd or NiMH batteries. Comparator IC1A compares the power supply voltage (or more accurately: 5/11th of this voltage via divider R1/R2) with the reference voltage of 3.9 V, which is generated by zener diode D1. When the power supply voltage drops to about 7.8 V, the output of IC1A will go Low, with the result that T1 (and therefore also the 2N3055) will block and red LED D2 will light up as a warning to show that the water is no longer being heated and the batteries need to be recharged.

The second comparator in the LM393 compares the voltage across NTC R5 with a reference value that can be set with trimpot P1. The resistance value of the NTC increases when the temperature reduces. Below a certain temperature (depending on the setting of P1) the voltage across R5 will be greater than the voltage at the wiper of the trimpot, with the result that the output of the comparator will go High, which turns on T1 and the 2N3055. The water is then being heated. This is indicated by orange LED D3. Resistor R7 limits the current through the 2N3055 to value of about 100 mA, the power dissipated in the power transistor is then a little less than one watt. The magnitude of the current is strongly dependent on the current gain of the actual 2N3055 that is used. It is a good idea to measure the current consumption from the batteries and if necessary to adjust the value of R7.

You can calculate at what ambient temperature the heated water will freeze using the following method. A 2N3055 has these characteristics: $R_{th(jc)}$ = 1.5 °C/W, $R_{th(ca)}$ = 175 °C/W, but that applies for air as ‘ambient’. The thermal conductivity of air is 0.024 W/(m·K) and for water this is 0.6 W/(m·K). Water therefore conducts 0.6/0.024 = 25 times better than air. So

$$R_{th(c\text{-}a)water} = 175/25 = 7\ °C/W$$

In this case (dissipation of 0.96 W) the water will freeze at temperatures lower than about –8 °C (0.96×(1.5+7)). This turned out to be the case pretty much in practice.

The supply current is provided by eight rechargeable AA batteries. When using batteries with a capacity of 2700 mAh there is sufficient energy to heat the water feeder for about 24 hours. It is therefore recommended to work with two sets of eight batteries and a fast charger. Allow the batteries to warm up to room temperature before you start to recharge them. It is also a good idea to bring the water feeder indoors during the night, since the birds won’t be drinking from it anyway.

Printed circuit board

A small printed circuit board has been designed for the electronics by Elektor Labs, which is shown in **Figure 2**. Mounting the (leaded) components is very straightforward and should cause little difficulty. Since the number of components is

Component List

Resistors

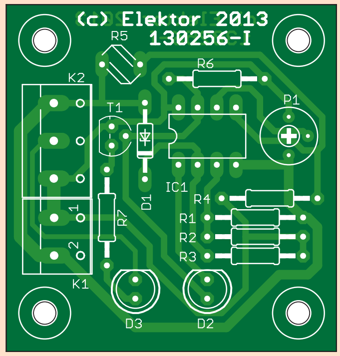
R1 = 12kΩ
R2 = 10kΩ
R4 = 3.3kΩ
R3,R6 = 1kΩ
R7 = 2.2kΩ
R5 = 4.7kΩ NTC
P1 = 4.7kΩ trimpot

Semiconductors

D1 = 3.9V 1W zener diode
D2 = LED, red, 5mm
D3 = LED orange, 5 mm
T1 = BC547B
T2 = 2N3055 (TO3 case)
IC1 = LM393N

Miscellaneous

K1 = 2-way PCB screw terminal block
K2 = 3-way PCB screw terminal block
PCB # 130256-1, see [1]



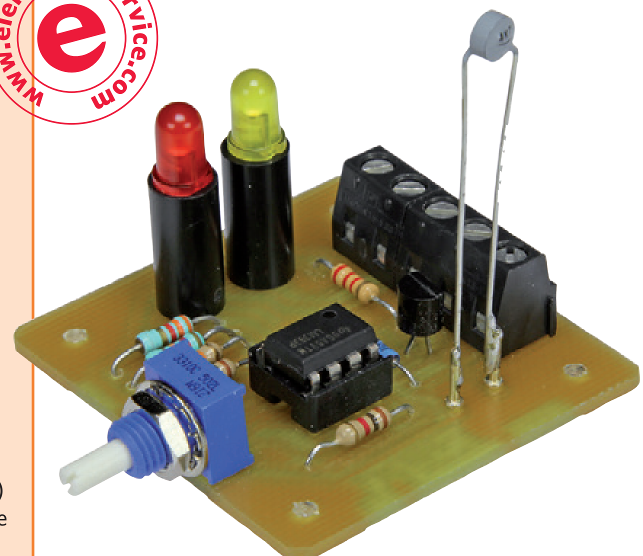


Figure 2. The printed circuit board is very easy to assemble. The NTC is connected directly (if necessary by extending the wires) to the PCB, the power supply and 2N3055 are connected using screw terminal blocks.

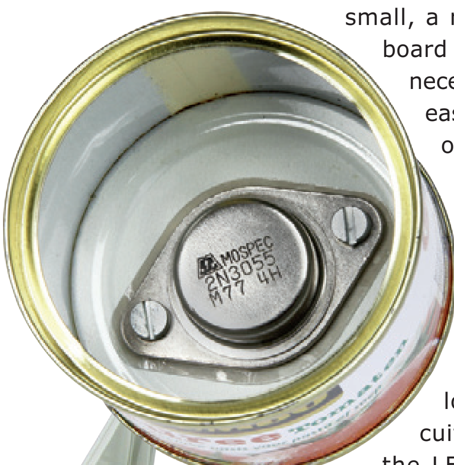


Figure 3. The mounting of the 2N3055 on the bottom of the tin. The transistor still needs to be covered with sealant to make everything waterproof.

small, a real printed circuit board layout is not really necessary. You can also easily build the circuit on a small amount of prototyping board, as the author has done (see the contribution on the Elektor.Labs website [2]). Depending on the location of the circuit board in the box the LEDs can be placed on the component side or the solder side of the board. To give access to the trimpot you can make a small hole in the box that can be closed with a small rubber plug. Also make a couple of holes where the LEDs are and seal them with silicon adhesive.

Then follows the mounting of the 2N3055 transistor to the bottom of the water bowl (see photo in **Figure 3**) and the mounting of the electronics and batteries in the box. If necessary, extend the wires of the NTC with longer (insulated) wires, depending on the position of the board in the box and glue it to the side of the water bowl. The 2N3055 and the batteries are then connected with wires to the terminal blocks. Adjust the trimpot to the desired temperature (this is a case of keeping an eye on the water feeder for a few days and adjusting it so that the water is just above freezing, otherwise too much energy is wasted). So, may we have some good frost soon. We (and the birds) are ready for it!

(130256)

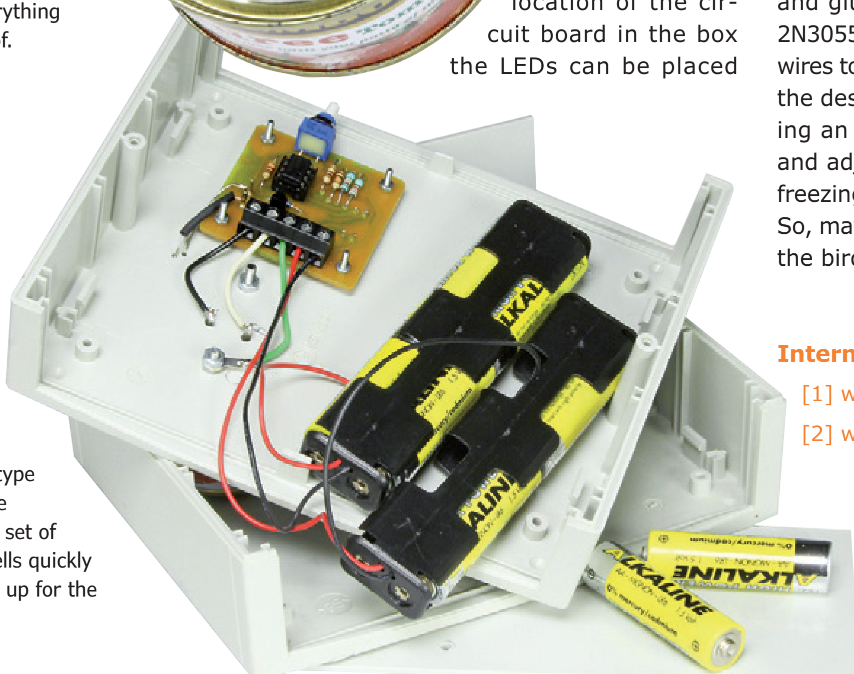


Figure 4. The prototype built in the lab uses a set of alkaline cells quickly connected up for the occasion.

Internet Links

[1] www.elektor-magazine.com/130256
[2] www.elektor-labs.com/130256


electronics
engineering
embedded
DISCUSS
design tips
tutorial
software
contests
engage
engineering
tools
mobile
system
audio
business
networking
media
data
COMMUNITY
social media
talk
information
projects
product news
Want to talk to us directly?
Share your interests and opinions!
Check out our New Social Media Outlets for direct engagement!
CC ROBOT
CIRCUIT CELLAR / AUDIOXPRESS / ELEKTOR

DesignSpark Tips & Tricks

Day #7: 3D Modding & Modeling

By Neil Gruending (Canada)

Today we're going to make a 3D view of our example board using DesignSpark PCB and also see how to export our design into DesignSpark Mechanical. These 3D views are very useful when designing a board.

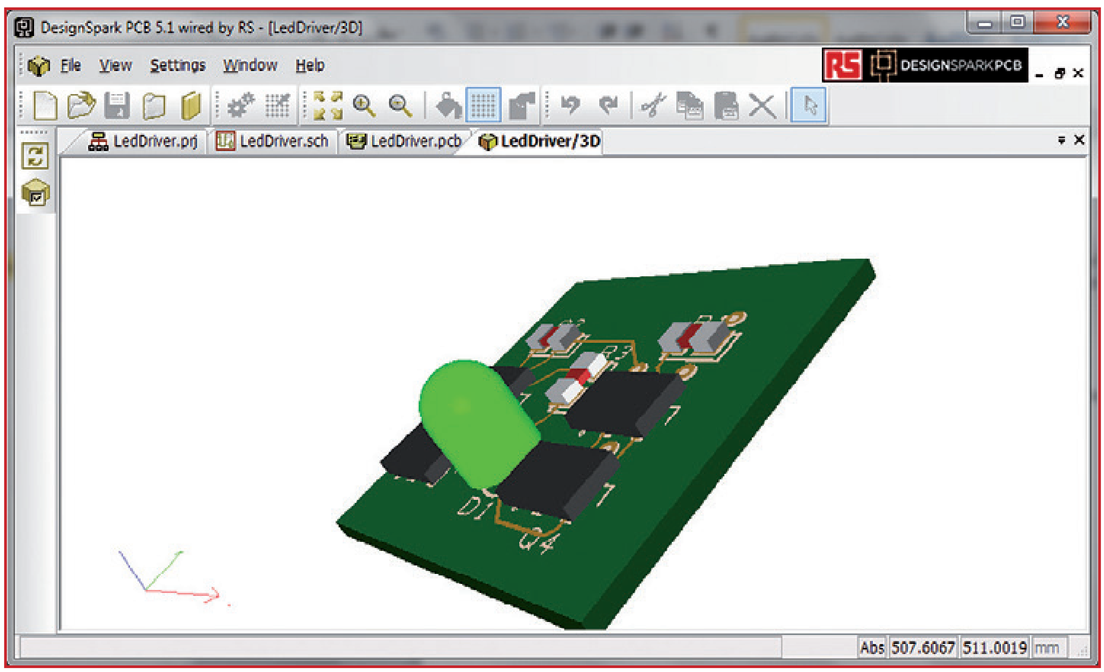


Figure 1.
A 3D rendering of our example board designed using DesignSpark PCB.

DesignSpark 3D Library Introduction

DesignSpark will generate a 3D view of a circuit board when you open the PCB file and go into the 3D→3D View menu. **Figure 1** shows our example board.

DesignSpark was able to create a 3D view even though our library doesn't contain any 3D models because it has default models available based on the PCB footprint name. This is different than other CAD packages where the 3D model is linked to the component in the library, and it means that the DesignSpark's 3D libraries are completely independent from the component libraries. You link the 3D to the PCB footprint name by defining a pattern matching rule in the 3D library, much like a search term or regular expression.

You can't use a custom 3D shape (like a STEP model) for a component, but DesignSpark does have a built-in 3D modeling tool that can generate most component shapes. One cool feature of the 3D tool is that it's parametric so that one 3D model can be used for many different real components. For example, one model for a 0.1-inch male pinheader can be used for headers with different pin counts because the number of pins assigned to the model can change with the footprint name.

Creating a Component 3D Model

The default SOT23 model used by our transistor doesn't resemble their real physical shape, so let's try making one. We will make a simple model because it reduces the design complexity and it will make it easier to export mechanical files in other CAD programs if DesignSpark PCB supports it in the future. It's also a good guideline to use with other PCB packages because otherwise the exported files can be so large that they're difficult for CAD software to manage.

The first step will be to click on the New Item button in the Library Manager 3D View tab. Use "SOT-23-L" for the PCB Symbol Name because that's the symbol used by our transistors. The name needs to be as specific as possible to make sure that DesignSpark will apply our model when it's matching the PCB symbols to 3D models. **Figure 2** shows what the Edit 3D Package window will look like after making the necessary changes. I set the Package Style to Shape because DesignSpark will make a best fit area that includes the component silkscreen and pads which it will extend in the *Z* direction by the Height measurement. In this case the height is 1.10 mm which is the maximum component height from the datasheet. The Inside parameter lets you reduce the size of the Shape. In this case I used the value of 0.80 mm to reduce the shape width to expose the component pads. You can calculate this number, but I just estimated it by comparing it to the silkscreen width. An easy way to do this was to click in the preview window and rotate it until I had a view where I could see the shape and silkscreen outline. I also changed the pin style to Gullwing.

So now let's save the 3D model, it should look like **Figure 3**. It's looking much better now except that transistor bodies aren't lined up with the center of the component. Remember how a 3D shape tries to do a best fit over a components silkscreen and pads? In this case there's a pin-1 designator that's far to the left of each transistor which is being included by the shape object. To fix it, open the SOT-23-L PCB symbol from the library, delete the pin-1 designator and then update the component in the PCB. The new 3D view would look like **Figure 4**. Now the transistor bodies are centered but they aren't quite wide enough. Also now all the pins are rendered properly whereas previously one of them was rendered

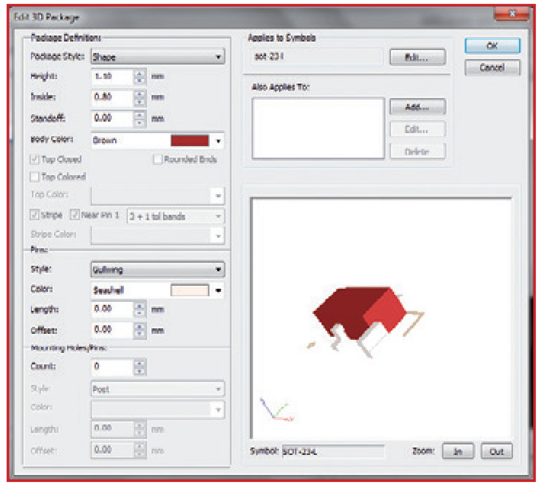


Figure 2.
Editing the 3D parameters of the SOT-23-L case.

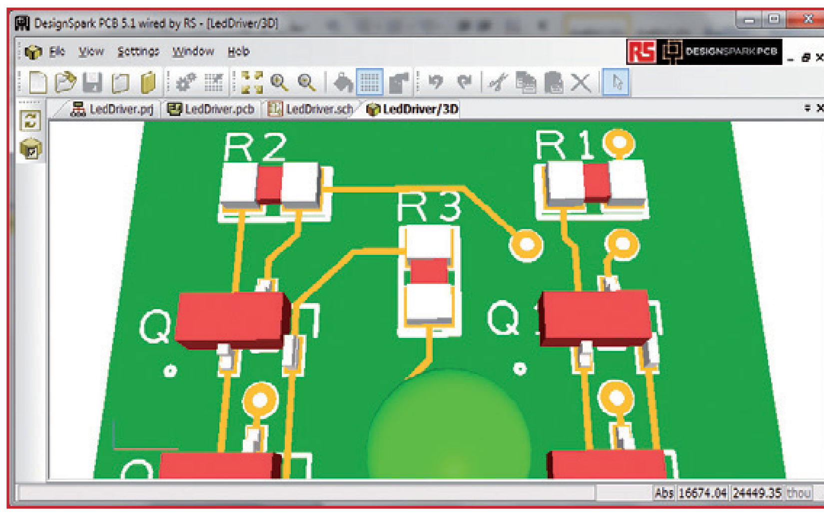


Figure 3. The transistor bodies look more realistic in 3D view now.

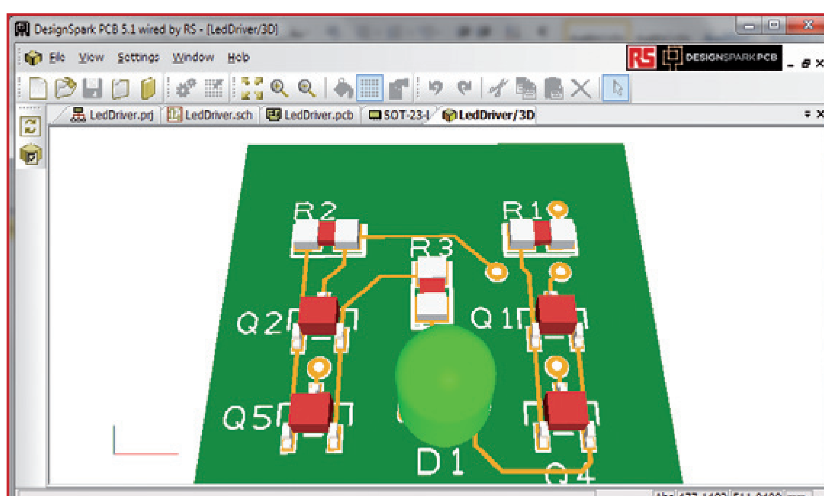


Figure 4. Although the transistor bodies (Q1, Q2, Q4, Q5) are within their designated areas indicated by the silkscreen, sadly they're not quite wide enough.

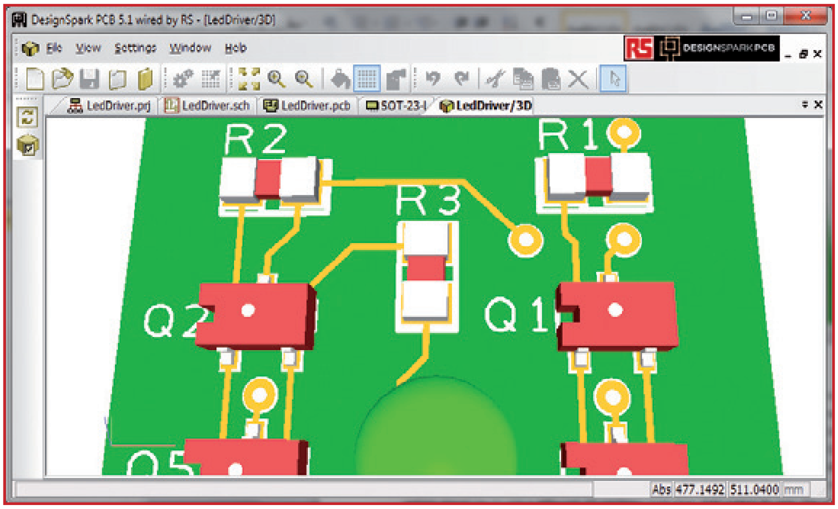


Figure 5. As close as we can get to the real thing.

as a large rectangle. DesignSpark doesn't have a setting to independently control the 3D inside parameter for the width and length of the generated shape so some creativity is required. You could add some dots or lines 0.8 mm from the left and right edges of the transistor silkscreen to fool the 3D shape calculation. It would render nicely but then you're adding weird marks to the silkscreen which could be confusing on the real board. Another option would be to use a different 3D Package Style that does a best fit over the component silkscreen only like DIL or DILSwitch. The DIL style will add a pin-1 notch to the generated view whereas the DILSwitch will add a switch like structure to the generated view. Choosing the DIL style and setting the Inside parameter to 0 will produce a layout like in **Figure 5**. It's not perfect, but I think it's as close as we can get to the real part and still get a realistic photo view.

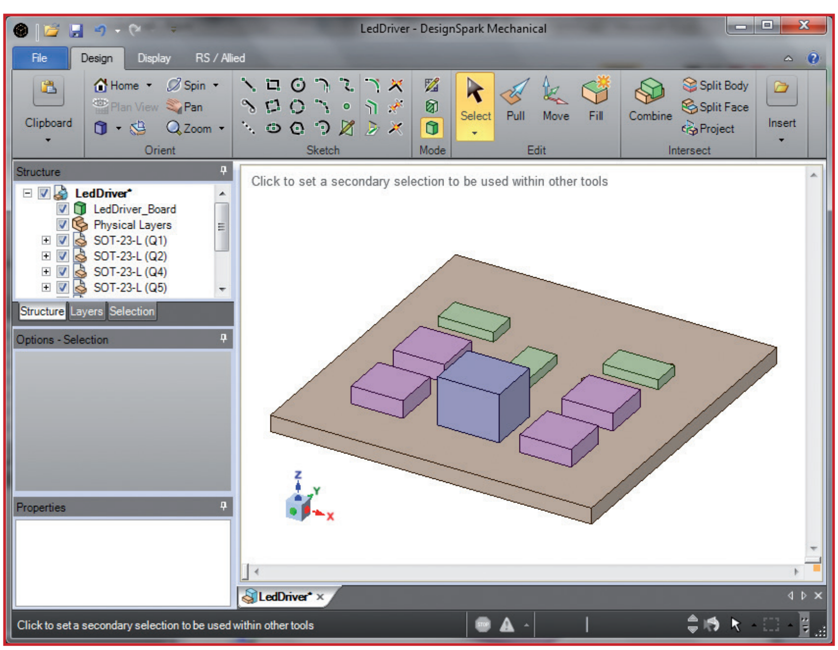


Figure 6. Initial result of importing the 3D model of our example board into DesignSpark Mechanical.

Importing a Design into DesignSpark Mechanical

DesignSpark PCB can't export the 3D view directly to a CAD program but it can export an IDF which is a standard interchange file format that can be imported by many CAD programs including DesignSpark Mechanical. Exporting an IDF file is done from the PCB using the Output→IDF menu. You have to specify the board thickness (1.6 mm) and which layer to use for the component outlines which is usually Silk Screen. **Figure 6** shows what it looks like in DesignSpark Mechanical.

All of the basic design parameters are there, including component heights and placement even though it doesn't include our component 3D model information. The DesignSpark website has more information about this process in the tutorials found at [1] and [2].

Conclusion

Today we experimented with DesignSpark PCB's 3D modeling capabilities. Next time we'll look at some of DesignSpark PCB's other mechanical abilities.

(130303)

[1] http://designspark.com/eng/tutorial/exporting-designs-from-designspark-pcb-to-mechanical-cad

[2] www.designspark.com/eng/tutorial/designspark-mechanical-importing-electronic-designs-pcbs

Join the Elektor Community

Take out a GOLD Membership now!

Your GOLD Membership contains:

- 8 Regular editions of Elektor magazine in print and digital
- 2 Jumbo editions of Elektor magazine in print and digital (January/February and July/August double issues)
- Elektor annual DVD-ROM
- A minimum of 10% DISCOUNT on all products in Elektor.STORE
- Direct access to Elektor.LABS
- Direct access to Elektor.MAGAZINE; our online archive for members
- Elektor.POST sent to your email account (incl. 25 extra projects per year)
- An Elektor Binder to store these 25 extra projects
- Exclusive GOLD Membership card

ALSO AVAILABLE:

The all-paperless GREEN Membership, which delivers all products and services, including Elektor.MAGAZINE, online only.

Take out your Membership now at www.elektor.com/member

Arduino Yún
Bridging Two Worlds?

Source: www.arduino.cc

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

The Cloud is where you have to be today. However, to connect to the cloud you need an internet connection, preferably wireless. This is where the new Arduino Yún board can help. Yún means cloud in Chinese and the board is equipped with a Wi-Fi module to connect to it. But the Yún is more than Wi-Fi—it also runs Linux. Arduino on Linux? Linux on Arduino? How does that work?

What's on board?
Quite a lot, actually. Let's start with what we are familiar with: the Arduino part.
Like all Arduino boards the Yún is based on a microcontroller unit (MCU) from Atmel. In this case it is an ATmega32u4 from the 8-bit AVR family. This is the same MCU as the one on the Arduino Leonardo board, and the Yún can actually be viewed as a Leonardo with an on-board Linux co-processor a.k.a. Wi-Fi/Ethernet/USB/SD-card shield.
This complicated shield is built around an Atheros AR9331 chip (we have to trust the Arduino documentation on this as there is no print on the metal shield covering it). According to the datasheet of this chip, it's a highly integrated IEEE 802.11n 1x1 2.4 GHz System-on-a-Chip (SoC) for wireless local area network (WLAN) access point (AP) and router platforms. The heart of this SoC is a 32-bit MIPS 24K processor, which goes to show that you don't always have to use ARM. The SoC communicates with the Leonardo part over a serial link.
The Atheros chip provides Wi-Fi connectivity, but it's also connected to an Ethernet connector providing a wired network interface. Furthermore, it has access to an AU6350 single chip integrated USB2.0 hub and multimedia card reader controller from Alcor Micro (mounted on the underside of the Yún board—**Figure 1**). This chip offers the Yún its USB host connector and micro SD-card slot. The AU6350 communicates over a USB connection with the Atheros chip.
Summarizing, the Arduino Yún combines—on a single PCB the size of an Arduino Uno

(53 x 69 mm)—three processors, a Wi-Fi interface, the four Arduino shield extension connectors, an Ethernet connector, a female USB-A host connector, a micro USB-B connector, a micro-SD card connector, a bunch of LEDs and three (3) reset pushbuttons.

Breaking with traditions

All other Arduino boards have Italian and English names, Yún is Chinese. This is not the only notable change; the Yún differs from other Arduino boards in several ways:

- The Yún is a far cry from the simple and easy-to-build Uno and earlier boards. The only through-hole parts are the connectors; all other parts are so small you can hardly see them. Repairing a faulty board will be grueling;
- I may be mistaken, but the Yún seems to be the first Arduino board for which the hardware CAD files are not (yet?) published, thus breaking with the Open Hardware tradition. The schematics are available as a PDF document, but I have not been able to find any information on the PCB. This is probably a 4-layer (or more) design, making DIY Yún boards difficult anyway—not to mention assembling them;
- There is no external power supply input—the Yún can be powered from 5 V only;
- The Arduino shield connectors have nice labels stuck on them that show which pin is which. There is not enough space on the Yún board to print these labels;
- The board sports three tiny reset pushbuttons, one for the WLAN, one for the Yún and one for the Leonardo, so always make sure you press the right one. Note that the Leonardo and therefore the Yún too exhibit different reset behavior compared to the Uno or related boards. To restart a sketch on the Yún you have to press the reset button twice!
- The Yún can be programmed over Wi-Fi and Ethernet (if you set it up properly);
- It appears that the design has not been done by the Arduino team but by a Boston-based company named *Dog Hunter* specialized in home automation control systems. They are probably going to notice an increase in traffic on their unfinished website;
- The Yún is manufactured in Taiwan, not in Italy.

Specifications

Arduino Leonardo (ATmega32U4) with

- 20 digital input/output pins (7 of which can be used as PWM outputs and 12 as analog inputs)
- 16 MHz crystal oscillator
- micro USB connection
- ICSP header

Atheros AR9331 running Linino,
an OpenWRT-based Linux distribution with

- Ethernet
- Wi-Fi
- USB-A host connector
- micro-SD card slot
- 3 reset buttons

While researching and writing this article Intel's Galileo board was launched. This Arduino compatible board is based on a 32-bit Pentium Quark SoC X1000 Application Processor, yet another new direction for Arduino. Also the Arduino Tre was announced, based on the 1 GHz ARM Cortex-A8 Sitara AM3359AZCZ100 from Texas Instrument. Let's hope that Arduino will not implode under the pressure of complication and diversification.

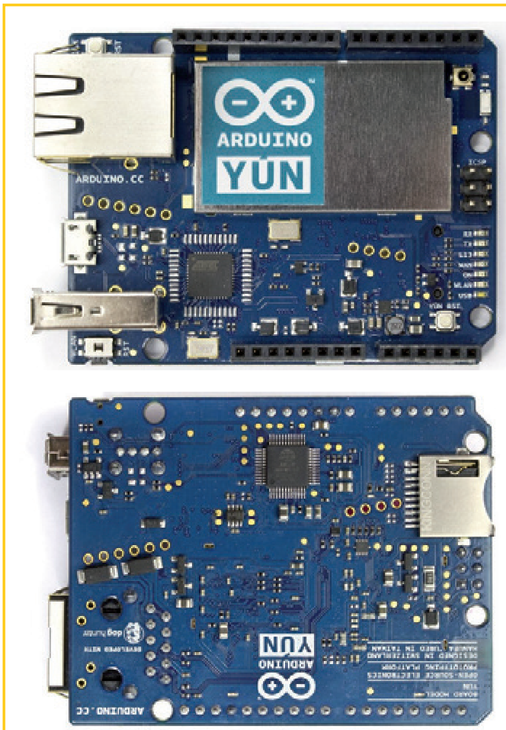


Figure 1.
Front and rear view of the Arduino Yún showing the SD card connector and the Alcor Micro AU6350 integrated USB2.0 hub and multimedia card reader controller.
(Source: www.arduino.cc)

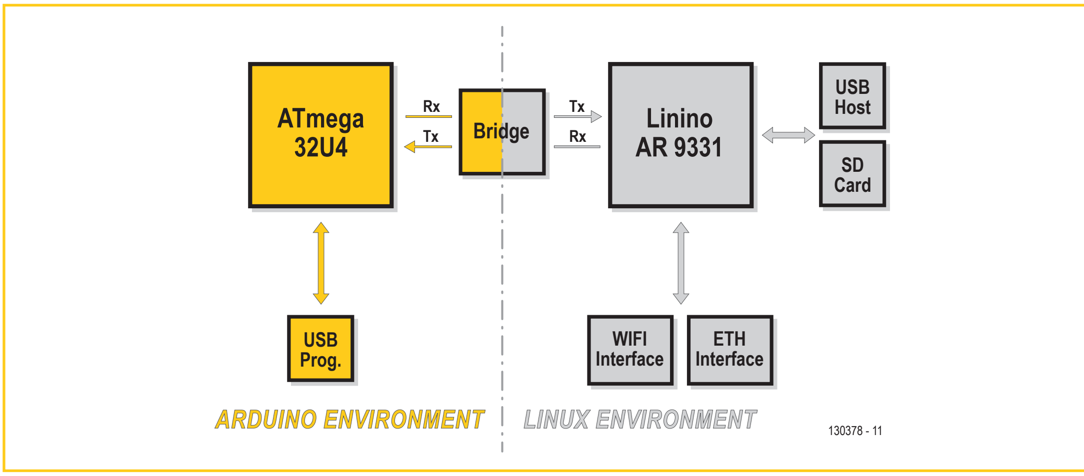


Figure 2.
Overview of the Arduino Yún showing the board's subsystems and the way they are bridged together.

Getting started

Breaking with traditions is one thing, but how does it affect the ease of use of the board? Let's find out.

First of all, you need a micro USB-B cable to connect the board. I have hundreds of Mini USB cables, but only one Micro USB cable as part of my phone battery charger. While you are looking for a suitable cable, download in the meantime the Arduino IDE 1.5.4 or later (I downloaded version 1.5.4 r2), it's only 134 MB and you will need it. Install the software before connecting the board.

Now connect the board to the computer. On OSX or Linux Ubuntu 10.0.4 and up everything should work straight out of the box. On Windows you may have to install a driver or two. Luckily a Windows installer is available which has the advantage that it can install the necessary drivers automatically. If you prefer to do it manually, the drivers are in the `drivers` folder of the IDE distribution.

After connection and installing drivers where needed the board should be operational. On my board this meant that the red LED L13 started blinking irregularly and the green On LED lit up. When, on your computer, you inspect the available wireless networks, you should see a new network named "Arduino Yun-XXXXXXXXXXXX" (where the X's represent hexadecimal characters). You may now be tempted to upload a sketch to the board, but the *Guide to the Arduino Yún* ('The Guide' from here on) on the Arduino website [1] suggests that you first set up the Wi-Fi connection, so let's do that now.

Connect your computer to the Yún Wi-Fi network and open a browser. Point it to the address 192.168.140.1 (the suggested link http://arduino.local did not work for me). You should now see a page that asks for a password. The default password is "arduino". Enter it and click on the Log In button. On the Welcome page that opens, click on the Configure button.

Set the time zone and select the Wi-Fi network that you want to use in the future, enter its pass phrase, etc. I did not change the default password, because I know I will forget it. I also set the REST API (the API that allows you to issue Arduino port commands as URLs, see below) to open, but that is of course up to you. When done, click Configure & Restart.

(Re)Connect your computer to the Wi-Fi network that you selected for the Yún and start the Arduino IDE. In the Tools→Board list choose the Yún, in the Tools→Port list pick the very last option "Arduino at xxx.xxx.xxx.xxx (Arduino Yún)" where the x's form a valid IP address (192.168.2.6 in my case).

Now you can try out a sketch. Open for instance the Blink (or BlinkWithoutDelay) example and click the Upload button. After compilation you will be prompted for a password. Enter the one you assigned to your board ("arduino" for me since I didn't change it). Uploading of the sketch starts, the board is restarted (you may hear a USB disconnect/connect sound on Windows) and the sketch is executed. Note that you don't have to enter the password every time you upload a new sketch, you have to do it only once at the beginning of a programming session.

That was not too bad, was it? It seems a bit silly to program a board that is physically connected to your computer over Wi-Fi, but why not? BTW, it is also possible to use the Ethernet port to do all this, but I leave that as an exercise for the reader.

Linux

So, now you have an Arduino Leonardo that you can program wirelessly over Wi-Fi or with a cable over Ethernet. However, that was probably not the main reason why you decided to invest in an Arduino Yún board—more likely it was the Linux part that triggered your interest.
When you look at the circuit diagram of the Yún you may notice signal labels referring to 'Hornet'. Feeding 'Hornet' to a search engine together with 'AR9331' takes you to OpenWRT, an open-source community project that allows commercial network routers to be used as Linux computers. The Atheros processor on the Yún runs such an OpenWRT-based Linux distribution named Linino. Linino can be configured wirelessly. Enter the Yún's IP address in a browser, log in, and then click on the "advanced configuration panel (luci)" link. This will open a status page from where you can access all kinds of parameters. At the top is a black menu bar with many options. Take your time to browse around. Have a look at the kernel log page to see what happened during start-up. Use the System→Software menu to install or remove software, use the System→Startup to see what is loaded during Linux startup. Here you can also add commands to be executed at the end of the boot sequence. Clicking the "Arduino Web Panel" link at the bottom of each page takes you back to the Yún's homepage.

Take me to the bridge

According to The Guide, you can access the Arduino pins (or ports) from the browser by typing in the right URL (through the REST API). For example, it is possible to make the URL (replace the first "arduino" by the name of your board)

http://arduino.local/arduino/digital/13/1

set digital output pin 13 as if you called the Arduino API function

```
digitalWrite(13,1);
```

You can also go the other way around and control Linino Linux from within a sketch. This is made possible by the Arduino library "Bridge" that allows you to access the USB, Ethernet, Wi-Fi and SD card devices in a sketch as well as run scripts and communicate with web services on the Linux module (**Figure 2**).
To familiarize yourself a little with the Bridge, I'd suggest entering the Console example sketch from The Guide. When you compile and upload it to the board, you will be able to use the serial monitor included in the Arduino IDE to communicate with the board (as if it was a normal Arduino board). After receiving the welcome message type a capital 'H' to switch on the red LED L13. Typing an 'L' will switch the LED off.
To make things a bit more complicated, you can do this also from a terminal like Tera Term or PuTTY. I used Tera Term and here is how to do it: Open Tera Term. In the New Connection dialog select TCP/IP; for the Host enter the IP address of your Yún. Set the Service to SSH and the Port to 22. Click OK. (You may get a warning about the Host not listed in your cache or something; simply allow whatever the program wants.) Now a new dialog window is opened where you are required to enter a user name ("root") and a Passphrase (the password you set for your board, "arduino" by default). If all goes well you should now see the screen from **figure 3** meaning that you are connected.

At the prompt ("root@Arduino~#") type "telnet localhost 6571" and hit Enter. Nothing will happen. Nothing? Try typing 'H' or 'L'. With these two commands you can control LED L13. Cool, right? The Guide gives an example sketch that runs a little Linux program named "curl". Before trying

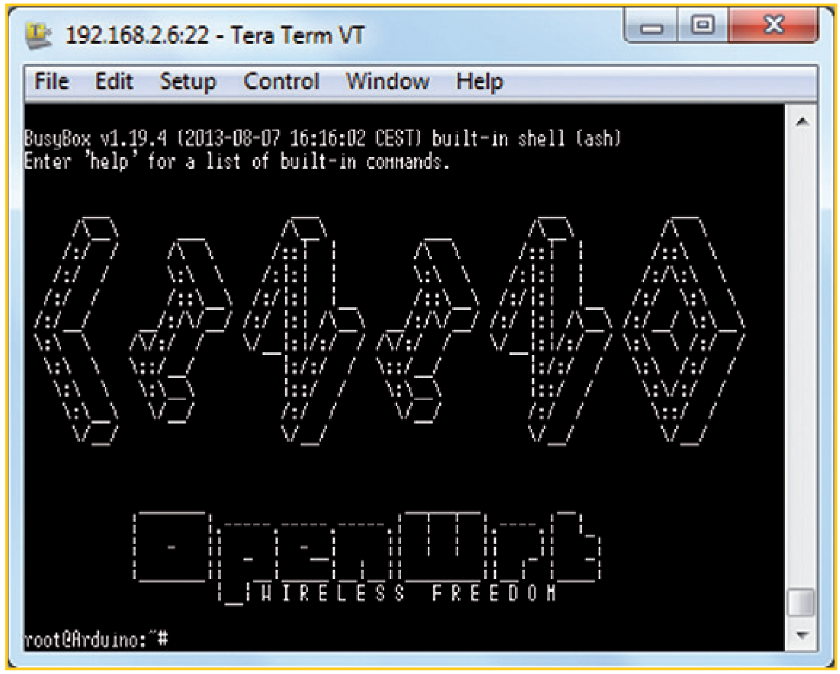


Figure 3.
Welcome to the Linino command shell.

Raspberry Pi

Looking at the Arduino Yún one cannot help thinking of Raspberry Pi (R-Pi), the $35 Linux board. So how do the two compare?
The RPi has been available for over a year now, a large user community has developed and many RPi to Arduino projects have been published together with lots of projects that accomplish the same tasks as those targeted by the Yún. The RPi does not have on-board Wi-Fi, but you can stick a cheap Wi-Fi dongle in its USB port. The RPi has support for graphical displays, the Yún does not. The RPi requires Linux knowledge whereas the Yún does not. The R-Pi does not need a host computer to develop applications on, the Yún does. However probably the most important difference: Yún will cost you twice as much as an RPi

the sketch you may want to see its result in the Linux console. At the prompt, type

```
curl http://arduino.cc/asciilogo.txt
```

Press Enter and wait a few seconds. You should see appear the Arduino logo in ASCII (**Figure 4**). The point of this demonstration is to show that whatever you can do in the Linux console, you can also do it from within an Arduino sketch.

To be honest, I have tried the example sketch and it compiles and uploads without problems, but I have not succeeded in making the result visible. I must have missed something here.

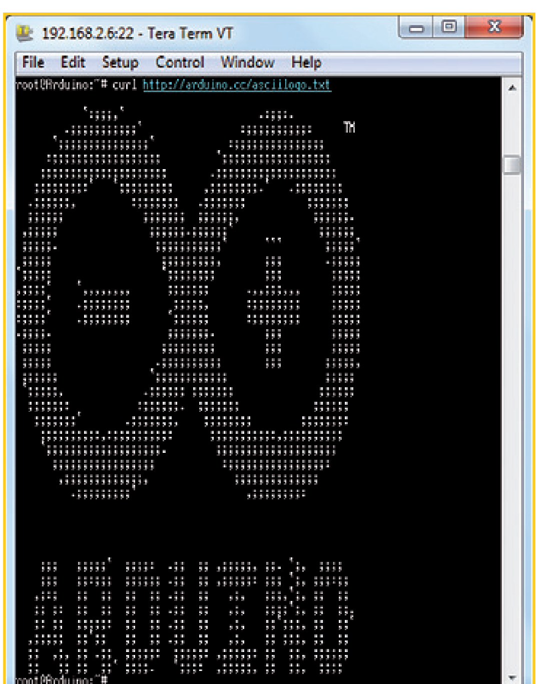


Figure 4.
The Arduino logo in ASCII as fetched by curl.

Feeling lost?

I don't know about you, but all this leaves me with a rather strange feeling. On the one hand I have the familiar and extremely simple Arduino IDE running on my PC while on the other I have access—through a browser on the same PC—to a web interface with tons of Linux options. Both programs target the same little blue board lying next to my PC.

The Guide almost starts with Python and other—to me unfamiliar—Linux concepts, and I cannot help feeling a bit lost. Why would I use a little microcontroller to run a Linux script if I can run the script also directly on the on-board Linux system? What does the Arduino interface add for the Linux programmer? Arduino is targeted at people with little to no programming experience, yet the user is supposed to know how to control Linux from the command line?

It is not that bad. For starters, you can skip the whole Linux bit and just be happy with how the Arduino team configured it for you. Using the REST API you can control the Arduino board from a web application through URLs that you process in your own sketch. Thanks to the Temboo [2] library you can easily interface an Arduino sketch with Twitter, Dropbox, Gmail, MySQL and many more web services. For the more adventurous there is Spacebrew [3], a software toolkit based on Websockets to interconnect interactive things. Once you master the Yún's built-in Cloud support you can go further and install your own programs and utilities on the Linux module to enhance its capabilities and your options.

Steep

The Arduino Yún is not for beginners. For example, connecting the board to the Arduino IDE is not Plug 'n' Play at all. And once you get it going you are supposed to use it for client-server web applications. However, once you master all this, the Yún can be an excellent tool to create fun applications.

(130378-I)

[1] Guide to the Arduino Yún: http://arduino.cc/en/Guide/ArduinoYun

[2] Temboo: https://temboo.com/arduino

[3] Spacebrew: http://docs.spacebrew.cc/

What it takes to make an LED blink

By **Thijs Beckers**
(Elektor Labs)

In the Old Days blinking an LED was done with no more than a switch or a timer IC and the inevitable current limiting resistor added. Nowadays it seems the amount of electronics needed—or better, *used*—to perform even the simplest of tasks seems to have grown to extraordinary proportions.

Although microcontroller development platforms have become more powerful than ever (look closely and spot the Raspberry Pi and Beagle Bone Black in the pile, both very potent boards, as is the Xilinx Zedboard FPGA dev system), many tutorials faithfully start out with a short lecture on 'switching something' which in most cases boils down to 1 (say, one) LED.

Behold, then, the staggering complexity of the systems we work with today. Melancholy strikes and I cannot help reminiscing for a while on the simple life back then.

(130417)

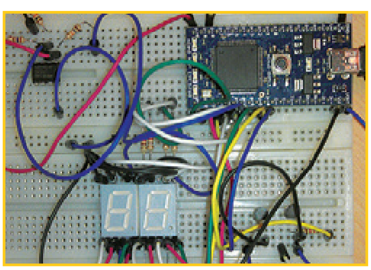
Join the Fourth Industrial Revolution!

By Clemens Valens
(Elektor.Labs)

Intelligent interconnected objects, devices and machines are revolutionizing the world we live in. The world-encompassing network dubbed The Internet of Things (IoT) is expected to change the way we interact with our environment in a dramatic way. Wireless and energy harvesting technologies are key to make this metamorphosis come about. Elektor.Labs and its users already jumped the bandwagon—what about you?

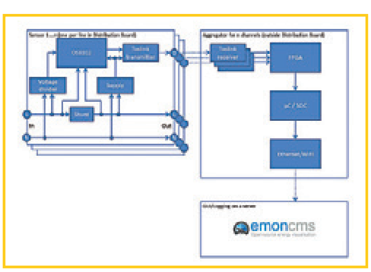
Wireless Resonant Power for a 2.4GHz Security Camera

In this project a wireless security camera is modified to become wireless. And we mean it—no power cord and no batteries either. The power is transferred to the camera by a pair of air-coupled coils (transmitter, receiver) helped by resonant capacitors.
http://www.elektor-labs.com/node/3653



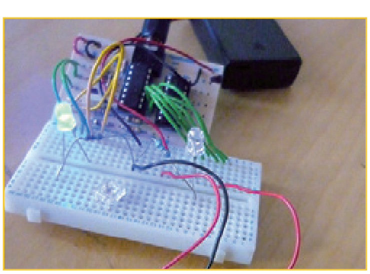
Tweeting Freezer

OP **mvnieuw** posted a smart freezer project employing an NXP mbed module not just to to monitor his freezer's 2-digit 7-segment LED temperature display, but also publish its status on Twitter. Of course I became a staunch follower of the OP's freezer; I simply feel uneasy when I don't know how it is doing.
http://www.elektor-labs.com/node/3065



Multi-channel Isolated Smart Energy Meter for Distribution Board

Smart Metering and the Smart Grid are two hot IoT / Industry 4.0 topics. OP **markusrr** is working on such a project. His goal is to monitor his energy consumption in as much detail as possible by individually monitoring every connection to and from the distribution board of his home.ß
http://www.elektor-labs.com/node/3322



Charge Controller for Off-Grid Systems

Quite a number of people are working on alternative power. The problem here is to charge a battery the proper way using power sources like a solar panel or a hydro/wind turbine. OP **Chunky** is tackling this with a charge controller designed for off-grid battery-based systems at voltage levels of 12 V or 24 V DC and currents up to 20 A DC.
http://www.elektor-labs.com/node/3482

Note: OP stands for Original Poster—the person who started an online project or discussion. OPs wishing to qualify for their project being published in Elektor magazine (i.e. on paper) must regularly check the email address they use to access Elektor.Labs. This is our only means of contact.

Where There's Smoke There's Fire

By **Thijs Beckers**
(Elektor.Labs)

Recently the EU Parliament adopted a new Amendment to Article 18 of the Directive of the European Parliament and of the Council on the approximation of laws, regulations and administrative provisions of the Member States concerning the manufacturing, presentation and sales of tobacco and related products [1].

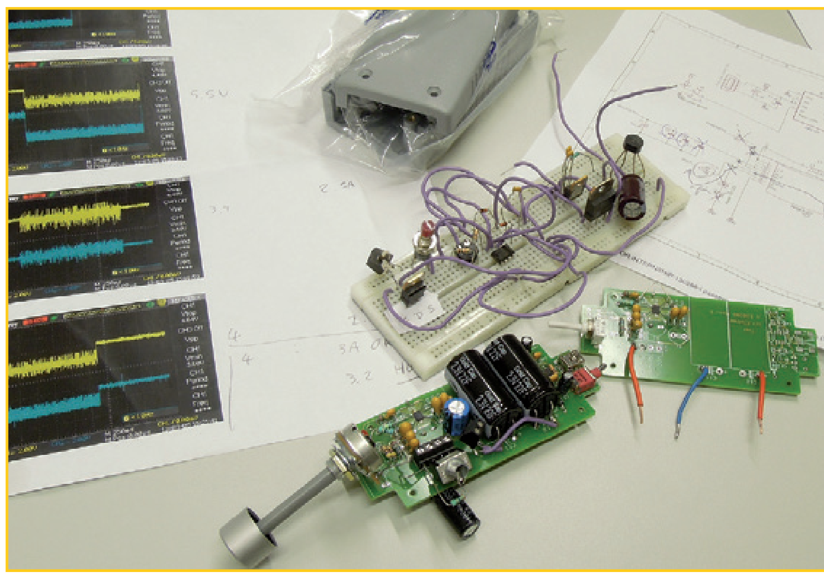
If you're still reading after this prolix sentence, this is about the currently popular *e-cigarettes* and *personal vaporizers*. This amendment—no. 170—compels a limit of 30 mg/ml of nicotine in liquids, dictates tht labels must list all ingredients contained in, and emissions resulting from the use of the product, on vials and instructions. It allows flavors to be used in the products, puts a limit on advertising, sponsorship, audiovisual commercial communication and product placement of products containing nicotine. It prohibits sales to minors, and defines a couple of other details.

In short, the latest amendment means the *e-cig* is still alive and kicking in the EU, even though some political parties tried to re-categorize it as a pharmaceutical product (which was rejected). To cap it all, intelligent regulations and improved quality standards have been set up.

One of our former trainees, French Aurélien Moulin, came up with the idea to design a personal vaporizer that runs on USB power. Since 'smoking' a vaporizer is generally allowed in non-smoking areas, this would be an ideal solution for office-resident vaporizer addicts. No more dead batteries and no more deferred satisfaction.

Aurélien started working on his project and soon it became apparent it wouldn't be that easy to implement a proper circuit. The vaporizer part needs (a lot) more peak power than a standard USB connection can provide, so he decided to implement two 2.3-volt, 22-farad (!) Goldcap bulk capacitors to soften the current surge and distribute the power drawn from the USB port more evenly.

Although he hasn't been able to finish his project within the time set for his traineeship (mainly because of other higher priority projects) the project still lingers in the back of his mind. For the time being, other obligations—again mainly his studies—take up too much of Aurélien's time to continue with this project. It is our sincere wish that someone else, be it a colleague or a Member of our Elektor Labs platform [2], picks up Aurélien's idea and turns it into a usable product.



On a side note: there's this trifling matter of limitations to advertising for and commercial communication about tobacco products as set out in Directive 2003/33/EC and 2010/13/EC. Technically, this design, when finished, doesn't contain any nicotine, but it could be considered a tobacco-related product, to which all the aforementioned rules and regulations apply. Food for our attorneys, I guess.

(130246)

Internet Links

[1] http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-%2F%2FEP%2F%2FNONSGML%2BAMD%2BA7-2013-0276%2B169-170%2BDOC%2BPDF%2BV0%2F%2FEN
or google 'amendment 170 tobacco'.

[2] www.elektor-labs.com

Sensirion Joins Atmel's Sensor Hub Platform Ecosystem

Sensirion has partnered with leading platform manufacturer Atmel to create an easy-to-use and power efficient sensor hub solution. This relationship allows Sensirion to provide its customers with turnkey sensor hub solutions and enables designers to create smarter, connected devices including mobile devices and wearables.

Atmel's sensor hub solutions combine inputs from different sensors, which range from motion sensors to environmental sensors, such as Sensirion's humidity and temperature sensors. These sensor hub solutions not only provide real-time direction, orientation and inclination data, but also now include environmental information, bringing visibly superior performance to a range of applications including gaming, navigation, augmented reality, and contextual awareness.

"As devices get smarter and more connected, there is an increasing need for environmental sensors, such as humidity and temperature," said Johannes Winkelmann, Technology Evangelist, Sensirion. "By partnering with Atmel on their ultra-low power sensor hub solutions, our customers can implement always-on sensor solutions while maintaining low-power consumption to increase the battery life for many of these battery-operated devices."

"With the increasing number of sensors in consumer devices today, low power is a key differentiator specifically for battery-powered devices," said Espen Krangnes, Senior Product Marketing Manager, Atmel Corporation. "The platform is the first of many devices in this series that is specifically tailored for sensor hubs. Our ultra-low power and devices features, and the flexible development ecosystem bundled with the best sensors and sensor software in the market allows customers to create unique and differentiating products. We are excited to partner with key sensor companies in the market to offer the best solutions for our customers."

According to market analyst BCC Research, the market for sensors is expected to increase to nearly $91.5 billion by 2016. With more sensors being integrated into mobile, gaming, consumer, wearables, and healthcare devices, there is an increasing demand to offload the application processor with a standalone microcontroller that fuses the sensor data.

www.atmel.com/SensorHub www.smart.sensirion.com (130364-IV)

Ultra-Low Forward Voltage Schottky Diodes

Toshiba Electronics Europe (TEE) has extended its family of surface mount Schottky barrier diodes (SBDs) with a new device CxS15S30 device based on the latest Toshiba semiconductor process, which improves forward voltage and reverse current performance.

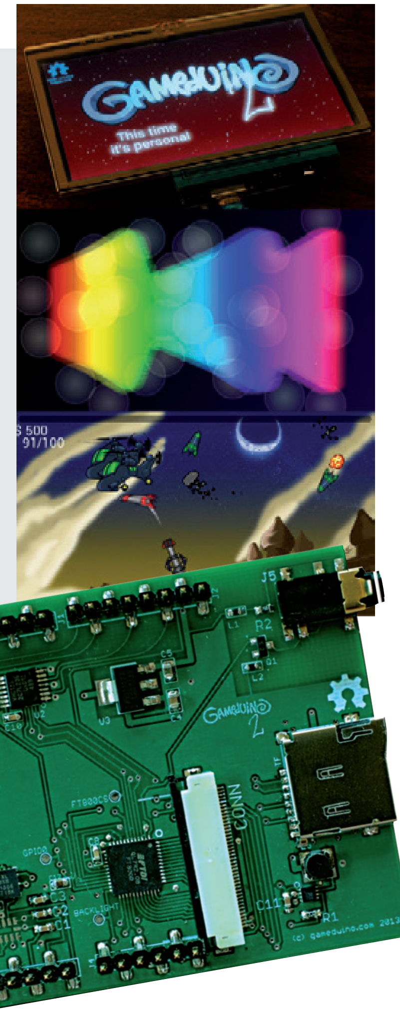
The CxS15S30 SBD offers 1.5 A and 30 V maximum ratings for average rectified current and peak reverse voltage, respectively. Package options comprise an ultra-compact LGA type CST2C package (CCS15S30) measuring only 1.6 mm x 0.8 mm x 0.48 mm, and a standard SOD-323 package (CUS15S30).

Toshiba's new diode will be particularly suited to space-limited applications where high current handling and low forward voltage (V_F) characteristics are key requirements. The CxS15S30 delivers the high-efficiency operation demanded by battery-powered and other power-sensitive designs. A low typical forward voltage rating of V_F = 0.39 V at 1.5 A and the low typical reverse current of only 200 µA ensures lowest loss operation in most common applications such as LED backlight circuits or current backflow prevention in battery charging.

Featuring a total typical diode capacitance of just 200 pF means that the CxS15S30 can also be used in general high-speed switching applications.

www.toshiba-components.com (130364-I)

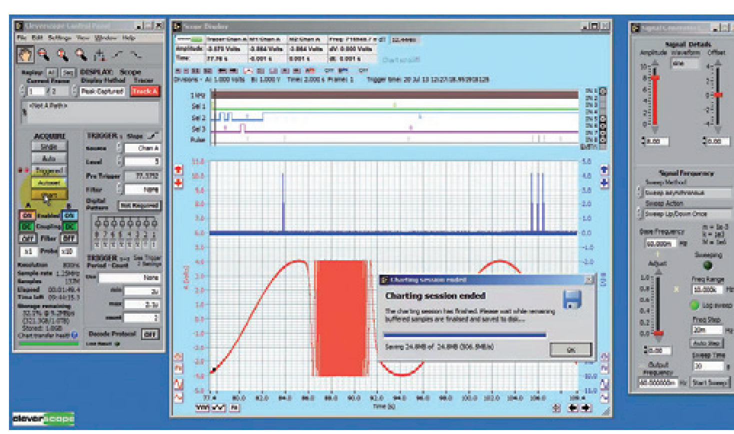
Gameduino has FTDI Chip's FT800 Graphic Controller Technology



FTDI Chip has confirmed that its ground-breaking FT800 Embedded Video Engine (EVE) is a key component in an exciting new Kickstarter project—the Gameduino2 game adaptor shield from Excamera. In 2011, the Kickstarter-funded Gameduino made major industry impact by successfully bringing vintage gaming to the popular Arduino platform. Now, with the introduction of Gameduino 2, users will be able to transform their Arduino units into modern handheld gaming systems that feature touch control, 3-axis accelerometers, headphone audio outputs and microSD data storage for game assets. In addition, they will benefit from the shield's support of next generation graphics via its built-in 4.3-inch display.
The Gameduino 2's graphical capacity, which stems from FTDI Chip's highly-integrated and easy-to-use FT800 IC, is much greater than that of its predecessor, thereby dramatically enhancing the gaming experience that results. Furthermore, its OpenGL-style command set makes programming far simpler to carry out. It can load JPEGs, support alpha transparency and has a full 32-bit color pipeline.
Incorporating a 4-wire touch controller and a single channel audio controller that allows midi-like sound quality, the FT800 EVE graphic controller employs an object-oriented approach (where objects are images, fonts, specific sounds, templates, overlays, etc.). This renders images in a line by line fashion with 1/16th of a pixel resolution, while still maintaining high-quality graphical representation. It means that system designs based on it are a lot more streamlined - requiring fewer supporting components, less board space, a lower power budget and shorter development times. "With the EVE concept FTDI Chip is looking to change the way in which people interact with everyday technology, by providing the display, audio and touch functionality needed to create innovative new products, while simultaneously being very cost effective and not placing heavy demands on the developers." states Fred Dart, CEO and Founder of FTDI Chip. "Gameduino 2 is a prime example of what can be achieved. We were all very pleased to be involved in this project."

www.kickstarter.com/projects/2084212109/gameduino-2-this-time-its-personal (130364-V)

Cleverscopes Stream to Disk



The latest release of software for Cleverscope includes streaming for the first time! Yes, the folks at Cleverscope have finally ironed out the kinks and you can now stream to hard drive at up to 1.6MSamples/sec for days—just make sure the hard drive is large enough! Update now and give it a try. To get it going, first point to where the files will be saved in File Options. And then set how fast you wish to go in Acquisition Settings. Click on the Stream button… And that's it! Stand back and watch the bits fly.
A video on the website shows how to set it up and investigate the resulting output. This software release also allows owners of "A" version Cleverscopes to upgrade to the CS701 Isolated AWG Sig Gen. The CS701 is useful for driving a small signal into the feedback loop in things such as power supplies, audio power amplifiers, servo amplifiers and positioning systems, and then measuring the resulting correction. By doing this over the operating frequency range you plot gain and phase, and can directly measure the system stability, all while the system is used normally. You need isolation because the feedback loop is not usually ground referenced.
Before the CS701 you needed an expensive network analyzer, so give it a go! Cleverscope are working on an option to make setting up for FRA even easier.

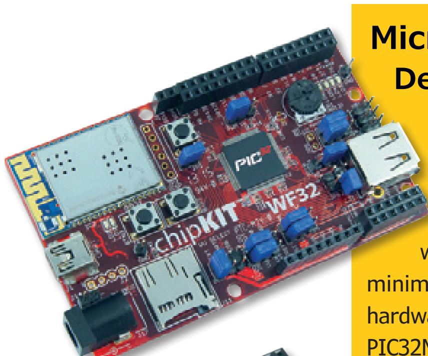
www.cleverscope.com/videos/ www.cleverscope.com/fra/ (130364-VI)

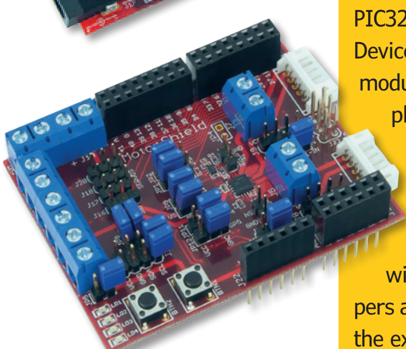
New Model KPSI 745 Submersible Level Transducer

The new KPSI® 745 submersible hydrostatic level transducer is specifically designed with a 3.5″ OD allowing installation into 4″ pipes to provide precision depth measurement under rigorous environments encountered in slurry or highly viscous applications. The 745 transducer can be ordered with level ranges of up to 115 feet (35m), custom polyurethane or ETFE cable lengths, and optional lifetime lightning protection. A protective cage option is also available to shield the elastomeric diaphragm from impact with suspended solid materials.

Measurement Specialties, Inc. designs and manufactures sensors and sensor-based systems. The company, recently ranked No. 76 on the Forbes Top 100 List, produces a wide variety of sensors and transducers to measure precise ranges of physical characteristics such as pressure, force, vibration, torque, position, temperature, humidity, fluid properties, mass air flow and photo optics. Measurement Specialties uses multiple advanced technologies—including piezo-resistive, electro-optic, electro-magnetic, variable reluctance, magneto resistive, digital encoders, thermistors, thermocouples, RTDs, capacitive, resonant beam, application specific integrated circuits (ASICs), micro-electromechanical systems (MEMS), piezoelectric polymers and strain gauges to engineer sensors that operate accurately and cost-effectively in customers' applications.

www.meas-spec.com (130364-VII)





Microchip: Arduino Compatible chipKIT Ecosystem with Wi-Fi® Development Board, IoT Cloud Software and Motor Control Shield

Microchip Technology Inc. announced the expansion of its Arduino™ compatible chipKIT™ ecosystem, with two new development tools from Digilent, Inc., and an embedded cloud software framework. Digilent's chipKIT WF32 board minimizes the need for users to purchase additional hardware or shields, by integrating Microchip's 32-bit PIC32MX695F512L MCU with Full Speed USB 2.0 Host/Device/OTG, its agency-certified MRF24WG0MA Wi-Fi® module and an energy-saving switch-mode power supply that employs Microchip's MCP16301 DC-DC converter, along with a microSD card—all while maintaining an Arduino hardware-compatible form factor. Digilent's chipKIT Motor Control Shield enables the development of applications using a wide variety of motor types, including Servos, Steppers and DCs, while allowing users to take advantage of the extra I/O pins found on many of the chipKIT development boards. This additional I/O provides added connectivity and more features than traditional, lower pin-count Arduino shields.

On the software side, an embedded cloud software framework enables designers to easily create "Internet of Things" (IoT) applications with the chipKIT WF32. Additionally, Digilent facilitates the rapid development of wireless HTTP server applications, via its comprehensive sample application that supports static pages loaded from the chipKIT WF32's microSD card, as well as dynamically generated Web pages.

Hobbyists, makers, students and academics are looking for an easy way to add wireless connectivity to their Arduino projects, which is provided by the combination of Digilent's chipKIT WF32 base board and its HTTP server example application. This board also provides professional engineers with a rapid method for evaluating Wi-Fi in their embedded designs, and for creating embedded cloud computing services using Exosite. Additionally, as with all chipKIT base boards, the chipKIT WF32 can be connected to Microchip's PICkit™ 3 programmer/debugger, allowing users to seamlessly move into Microchip's professional MPLAB® X IDE and XC32 C and C++ compilers.

Robotics applications are particularly popular with hobbyists, makers, students and academics. And, their robots are driven by exactly the motor types that the chipKIT Motor Control Shield is designed to support.

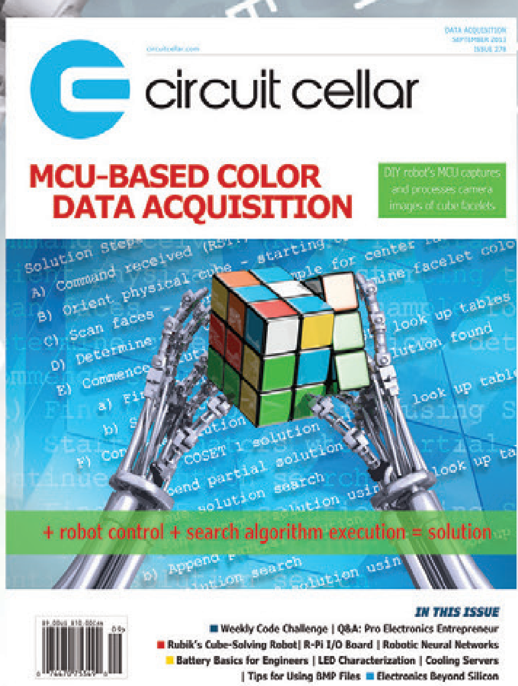
Digilent's chipKIT WF32 (part # TDGL021, $69.99) and chipKIT Motor Shield (part # TDGL020, $29.95) are both available today. They can be purchased from microchipDIRECT.

The chipKIT WF32-compatible embedded cloud computing framework, including source code and quick-start information, can be downloaded today from www.microchip.com/get/LS3W. Digilent's HTTP server example application can also be downloaded today from www.microchip.com/get/1V8L.

For more information on any of the above products, or for additional chipKIT resources, please visit the chipKIT Community Site, url below.

www.microchip.com/get/2T2W (130364-II)

We're celebrating a new look — and the next 25 years.



25% OFF CIRCUIT CELLAR

Whether it's **programming advice** or **design applications**, you can rely on *Circuit Cellar* for solutions to all your electronics challenges. **Raspberry Pi, embedded Linux, low-power design**, **memory footprint reduction** and more! Become a member, and see how the hottest new technologies are put to the test.

JOIN TODAY!

www.circuitcellar.com/sepN13

Hexadoku The Original Elektorized Sudoku

We ring in the New Year with a fresh installment of our celebrated Hexadoku conundrum for the electronically minded (and/or their spouses). Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of five Elektor book vouchers.

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

Correct entries received enter a prize draw. All you need to do is send us **the numbers in the gray boxes**.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for five Elektor Book Vouchers worth **$70.00 (£40.00 / €50.00)** each, which should encourage all Elektor readers to participate.

Participate!

Before March 1, 2014,
supply your personal details and the solution (the numbers in the gray boxes) by email to:
hexadoku@elektor.com

Prize winners

The solution of the November 2013 Hexadoku is: **E75F4**.
The Eurocircuits $140.00 (£80.00) voucher has been awarded to Christian Basler (Germany).
The Elektor $60.00(£40.00) book vouchers have been awarded to Wojtek Stoduly (Poland), Håkan Jönsson (Sweden); Ciril Zalokar (Slovenia).
Congratulations everyone!

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

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

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

ORDERING INFORMATION

To order, contact customer service for your region:

USA / CANADA
Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
E-mail: service@elektor.com

Customer service hours:
Monday-Friday 8:30 AM-4:30 PM EST.

UK / ROW
Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344
E-mail: service@elektor.com

Customer service hours:
Monday-Thursday 9:00 AM-5:00 PM CET.

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

COMPONENTS

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

TERMS OF BUSINESS

Shipping Note:
All orders will be shipped from Europe. Please allow 2–4 weeks for delivery.

Returns
Damaged or miss-shipped goods may be returned for replacement or refund. All returns must have an RA #. Call or email customer service to receive an RA# before returning the merchandise and be sure to put the RA# on the outside of the package. Please save shipping materials for possible carrier inspection. Requests for RA# must be received 30 days from invoice.

Patents
Patent protection may exist with respect to circuits, devices, components, and items described in our books, magazines, online publications and presentations. Elektor accepts no responsibility or liability for failing to identify such patent or other protection.

Copyright
All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, discs, and software carriers published in our books and magazines (other than in third-party advertisements) are copyrighted and may not be reproduced (or stored in any sort of retrieval system) without written permission from Elektor. Notwithstanding, printed circuit boards may be produced for private and educational use without prior permission.

Limitation of liability
Elektor shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connection with, the supply of goods or services by Elektor other than to supply goods as described or, at the option of Elektor, to refund the purchaser any money paid with respect to the goods.

MEMBERSHIPS

Membership renewals and change of address should be sent to the Elektor Membership Department for your region:

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

UK / ROW
Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344
E-mail: service@elektor.com

Do you want to become an Elektor GREEN or GOLD Member or does your current Membership expire soon?

Go to www.elektor.com/member.

Elektor AC Power Supply (1984)

Rescued from e-debris @ e-labs

By **Jan Buiting** (Elektor Editor-in-Chief)

The previous months we've been discussing equipment in the complex, bizarre, rare or expensive categories (tick as applicable—multiple options allowed). Here, we choose to go back to a primary requirement of lots of electronic DIY projects working at low voltages (say, 3 to 20 volts): the alternating voltage from the transformer secondary.

Smell out the trouble

It's not unwise or bold to say that (a) not a lot of electronics works without a proper power supply voltage and (b) that same power supply is the commonest source of failure in vintage equipment that's gone faulty.

While today problems in circuits are "addressed" the non-invasive way by firmware updates arriving over the web from the helpdesk in Pakistan, in the old days—like 25 years ago—you'd begin with removing fuses, followed by disconnecting the power supply input cabling from the rectifier section. Next, a thorough visual and olfactory investigation of the power supply section. Everything beyond suspicion so far, you'd reinstall the primary fuse and do a few AC voltage measurements on what *you think* are points with the transformer secondary voltage(s) on them. One hand in pocket, if applicable. Good voltage now. Somehow the DC power supply is overloading the transformer, or the raw-DC circuitry is faulty or interrupted somewhere. Where? Bridge rectifier and/or electrolytic reservoir caps (the big cans) are the usual suspects, and a good way to find out is apply external AC to the rectifier inputs, and slowly step up to the nominal value.

Professionals and sticklers might argue that a double-isolated variable transformer (variac) is the best way to tackle such cases, not realizing such a beast would be a rare occurrence in the early 1980's hobbyist workshop, let alone under the kitchen table. The crux: you need to carefully step up the voltage to the rectifier **inputs**, i.e. its AC terminals. Applying judicious measurement in the low-voltage parts of the power supply, this slow-and-careful method will disclose any errors without the risk of smoke, exploding electrolytics and more fuses successfully passing the test. Confidence now rules in your workshop instead of anxiety. Enter the adjustable AC Power Supply published in Elektor magazine's April 1984 edition.

Still @ Labs

There can only be three reasons for the tatty looking instrument pictured here to be around in Elektor Labs after almost 30 years: 1. it's fool-proof and easy to use; 2. it's downright useful; 3. it does not contain a microcontroller.

I found the little box with its two-tone imitation Tektronix case on Elektor Labs' only general-purpose workbench where all sorts of projects get prepped for photography—where chaos reigns and nobody bothers to clean up anything before finding pizza slices with 3 months' worth of free organic growth on them.

As it turned out, the instrument I rescued from the e-debris was a prototype that survived countless moves and equipment clear-outs, and eventually became an Elektor Labs resident.

Thirty years back

With the help of Harry Baggen, senior Editor of Elektor's Dutch edition of and a walking encyclopedia, I was able to trace the publication of the "AC-POWER SUPPLY" (*sic*) back to the April 1984 edition. I joined Elektor in October 1985. At just three pages the article entitled *a.c. power supply* (*sic*) is really trifling compared to some of the blockbusters we've seen in *Retronics* over the years, and Harry may have forgotten all about it if the actual instrument had not surfaced in our labs.

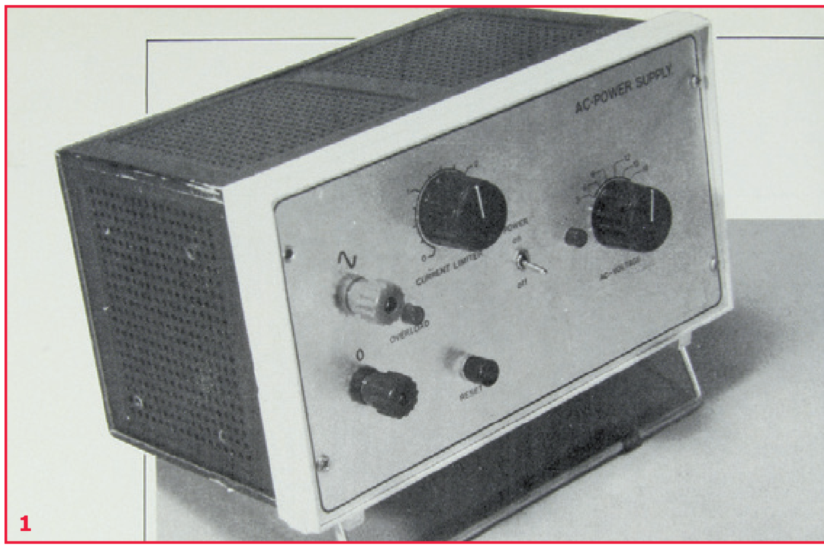
With the original article on my desk, it struck me that the instrument pictured there (**Figure 1**) was not an exact copy of the instrument I was looking at (**introductory photograph**). The arrangement of the controls and labels is different and a black front panel is used on the real thing with the famous diode-k prominent in the brand name 'elektor'. The enclosure with the nice tilt stand appears to be the same though. The most striking variance however is the use on the actual instrument of six banana sockets for the 3-6-9-12-15-18 VAC outputs instead of a single AC OUT socket and a 6-position rotary selector switch as advertised in the article.

When I opened the instrument by unscrewing the front and rear panels (**Figure 2**) I discovered the circuitry was built on a 110 x 80 mm piece of veroboard (a.k.a. stripboard; perfboard), while the 1984 article showed a nicely designed 110 x 45 mm printed circuit board. I have not undusted the inside.

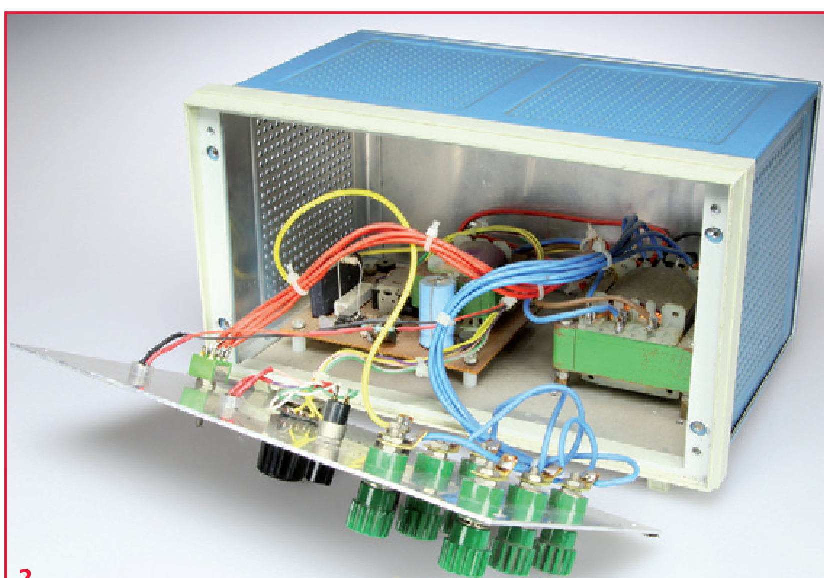
ESTD 2004

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

This leaves me with the question why the instrument has an artsy-techno black front panel of which the design was never published. No one on the current staff knows. The style closely resembles that of the 1977 ElektorScope.



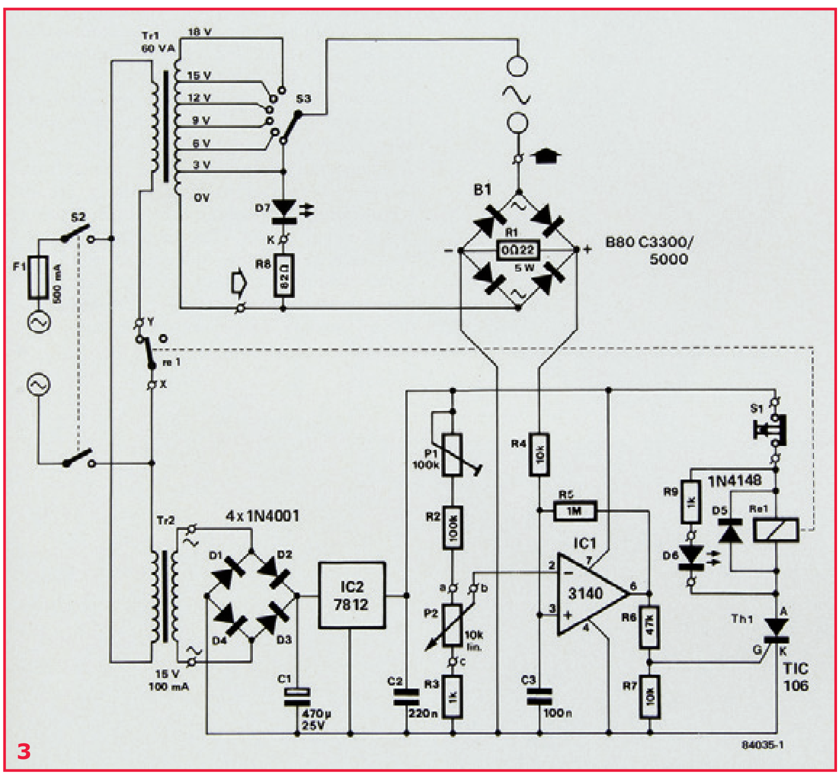
1


2

I have not checked the veroboard construction against the circuit as published in the magazine in 1984, but they appear identical functionally and in terms of components used. Note that 1-megohm resistor soldered directly across pins 3 and 5 of the CA3140.

How it worked—and works

Compared to DC adjustable benchtop power supplies, AC variants are rare, probably because we assume that the AC power transformer is a reliable component and while expensive, is easy to replace. However there are many situations in which you want to be in accurate control of the AC voltage applied to a power supply, like when the CEO or Lab Manager want you to be accurate about the minimum or maximum input voltage to a DC regulator.



3

Silently I was hoping for the circuit to provide an adjustable, stabilized output voltage, but instead it has selectable 3-volt steps from 3 V_{AC} to 18 V_{AC} directly off a transformer secondary. The built-in current limiter though is adjustable though with a control on the front panel, so my fuses can remain in their boxes. The instrument shown here can be adjusted from about 0 to 1.5 amps.

The schematic is reproduced in **Figure 3**. The output current drawn from Tr1 is sensed by R1, and the resulting pulsed DC voltage drives the current limiter circuitry around IC1. Note that this circuit is powered separately from Tr2, D1-D4 and IC2. IC1 compares a fixed (ermm…) voltage at its pin 2 with that developed across R1, where the former voltage is adjustable with the pair of preset P1 for the maximum output current, and potentiometer P2 for the continuous adjustment between 0.2 A up to the maximum. The maximum current is obviously dependent on the transformer used in position Tr1—the 1984 article suggests a 60 VA multi-tap type. As soon as the current draw on the output exceeds the level set on P2, the output of comparator IC1 toggles, causing Thyristor Th1 to be fired through R6-R7. Consequently the coil in relay Re1 is energized and the primary current to Tr1 is cut off through the X-Y contact. LED D6 also lights to indicate an overload condition has occurred needing investigation followed by repairs!
The thyristor will remain conductive even after the gate pulse has passed; meaning the only way to reset the circuit and restore the output voltage is to press S1—after clearing the (disastrous) cause of the current limiter action of course!
Although rotary switch S3 is duly specified at 5 amps in the parts list, I much prefer the six discrete banana sockets on the instrument as I have it.

A fine little instrument

I'd highly recommend systematically substituting faulty, suspicious, unfathomable or "beyond-Ebay" subassemblies of electronic circuits by **Known Good BoatAnchors** (KGBs). Doing so is highly educational and cheaper in the end than forever replacing new parts that blow due some other part having blown due to reasons beyond you.
Elektor's 1984 AC Power Supply is now safe from harm, and a KGB on my Retronics workbench. To celebrate the arrival the original 1984 (!) article is available as a .pdf file for free downloading by all fans of AC [1].

(130367)

Internet Reference

[1] a.c. power supply, Elektor April 1984, www.elektor-magazine.com/130367

Pro-Tronics

By Gerard Fonte (USA)

Last month I talked about what it was like as hobbyist fifty years ago. This month I'll look into the future to see what the next fifty years will bring.

Hardware

Starting easy: solid state drives will replace magnetic as well as optical drives. CDs and DVDs will go the way of the vinyl record. It may still be possible to buy hard copies of audio and video recordings, but they'll just be chips.

The 3-D printer will revolutionize PCB (Printed Circuit Board) proto-typing with the use of conductive polymers. This makes "plated"-through-holes and two-sided boards easy. There will be no more soldering. Components will be attached with conductive polymer glue. By the way, check out 3M's Z-axis adhesive tape (9703). It only conducts through the tape—top to bottom, not side to side. So, if you want a quick and easy way to attach a 100 pin TSSOP IC to your PCB, you might consider this. (Unfortunately, the AC electrical specifications aren't well defined by 3M.)

Batteries will disappear to be replaced with capacitors. Aerogel techniques have created capacitors with staggering values in a very small volume. I have a 2600 farad capacitor (that's right 2.6 Kilo-farad!) that's about the size and weight of a soda can (full). Capacitors (for hand tools) can be re-charged in 90 seconds and up to 500,000 times (versus 90 minutes and 500 times for conventional batteries). Coleman (USA) and GMC (Australia) have attempted to introduce the technology with screwdrivers but consumers don't want to pay $100 for a cordless drill, even if it means no more batteries to buy. Toyota is experimenting with capacitor assist in its Yaris Hybrid-R concept car.

Expensive power transformers may be replaced with components that operate directly from rectified 120 VAC household voltage. (That's 170 volts DC for bridge rectified and about half of that for half-wave.) There are plenty of inexpensive power MOSFETS that operate at 250 volts or more. Linear Technology has an op-amp that runs off of 140 volts (LMC6090) for about $4.00. Of course, off-line switching supplies will also be in general use.

Communication

Conventional local broadcast TV and radio will be gone. That's not to say that TV and radio will disappear. It's those multi-kilowatt transmitters that will fade away. The internet will carry the programs and music instead. Many radio stations already "broadcast" on the web. The ether will become quieter but will be much more active. Hard-wired installations will be rare. Low-power cell-like communications will even replace fiber-optic cables for TV and internet. I wouldn't be surprised to see a resurgence in amateur (Ham) radio on some of the frequencies left open by radio and TV.

Your cellphone will hold all your important records. This includes credit card numbers, medical information, driver's license, passport, insurance data, resume and everything else. But since people are always misplacing their phones, this actual data will be attached to you; like a wristwatch. It will communicate to your actual cellphone with a very short-rage, encrypted transceiver. It's possible to put everything on your wrist, but that's too small to be convenient. (Remember those early digital watches with the built-in calculator and very tiny buttons?) Biometric sensors will make your data useless to anyone else. Of course, this means death to cash. Money, credit cards and checks will no longer be used. Direct fund transfer from account to account will be the norm. Electronic receipts will be generated and all transactions will be recorded. In fact, it's very possible that non-electronic transactions may be banned by the government because they can't be traced and taxed (especially taxed!). Cash may only be found in the black market. The good news is that filling out your income tax forms will also be obsolete. Filing will be done automatically at the push of a button since all of your financial records are immediately available.

Home

The current method of electrical wiring is to cut the power cable at every outlet or switch box and then reconnect the wires together to continue on to the next outlet or switch box. It's really a rather silly technique. In the future, the power cable will be clamped at the box and insulation displacement connections will be made by simply tightening the clamp with screws. Fast and simple with no cutting of the wire. Insulation displacement has been around for a long time and is very successful. I think the biggest impediment is probably the archaic building codes.

Your roof (the southern facing part) will be made from flexible photovoltaic "shingles". They will be automatically self-connected during installation. Being flexible, they won't crack from expansion/contraction and will weather the weather well. (PowerFilm Solar, Silicon Solar and others already manufacture flexible panels.) The use of distributed solar power generation will significantly reduce atmospheric carbon emissions.

The last item is not electrical but does play an important role in energy conservation. This is the use of foamed concrete as thermal insulation. It's cheap, effective, insect repellent, non-toxic, non-flammable, water resistant, sound damping and increases the structural strength of walls. Check out AirKrete for more information.

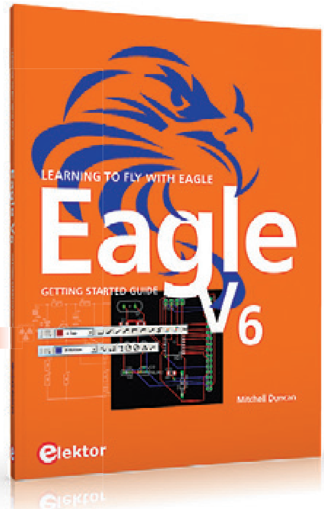
Print Lives

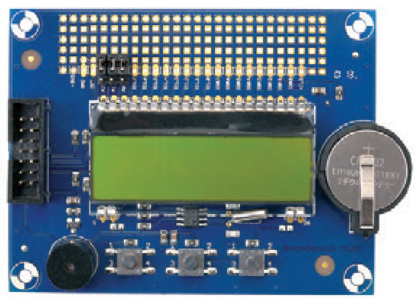
I think Elektor (or its equivalent) will still be around because hard-copy is just too convenient to disappear. And electronic hobbyists will always want to learn and do. So, if I'm not there in fifty years—feel free to start without me. And let me know how it turns out.

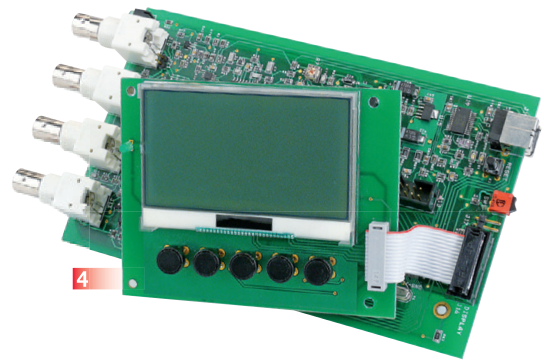
(130418)

1


15% DISCOUNT
for GREEN and GOLD Members!
www.elektor.com/january

2


3


4


5


Circuits & Projects Guide

1 Arduino

The Arduino user is supported by an array of software libraries. In many cases, detailed descriptions are missing, and poorly described projects tend to confuse rather than elucidate. This book represents a different approach. All projects are presented in a systematical manner, guiding into various theme areas.

In the coverage of must-know theory great attention is given to practical directions users can absorb, including essential programming techniques like A/D conversion, timers and interrupts—all contained in the hands-on projects. In this way readers of the book create running lights, a wakeup light, fully functional voltmeters, precision digital thermometers, clocks of many varieties, reaction speed meters, or mouse controlled robotic arms. While actively working on these projects the reader gets to truly comprehend and master the basics of the underlying controller technology.

260 pages • ISBN 978-1-907920-25-7
£34.95 • € 39.95 • US $56.40

Learning to fly with Eagle

2 EAGLE V6 Getting Started Guide

This book will quickly allow you to obtain an overview of the main modules of EAGLE: the schematic editor; layout editor and autorouter in one single interface. You will apply your knowledge of EAGLE commands to a small project, learn more about some of the advanced concepts of EAGLE and its capabilities and understand how EAGLE relates to the stages of PCB manufacture. After reading this book while practicing some of the examples and completing the projects, you should feel confident about taking on more challenging endeavors.

208 pages • ISBN 978-1-907920-20-2
£29.50 • € 34.50 • US $47.60

Display, buttons, real time clock and more

3 Elektor Linux Board Extension

This extension board was developed to further propel our Embedded Linux series of articles and the matching GNUblin board. It has a display, buttons, a real time clock and 16 GPIOs. Linux devotees, switch on your solder irons. The Linux extension board includes everything needed to provide the user interface for a wide variety of projects!

Module, SMD-populated and tested board, incl. LCD1, X1, K1-K4, BZ1, BT1 for home assembly
Art.# 120596-91
£31.10 • € 34.95 • US $50.20

The luxury of precision within everyone's reach

4 500 ppm LCR Meter

The remarkable precision of this device and its amazing ease of use are the result of careful design. It works so well behind its uncluttered front panel that one could almost forget the subtleties of the measurement techniques employed. A dream opportunity for our readers who are passionate about measurement to enjoy themselves. If, like us, you wonder at the marvels modern techniques bring within our reach, come along and feel the tiny fraction of a volt.

Set: main board and LCD board, assembled and tested
Art.# 110758-93
See www.elektor.com/lcrmeter

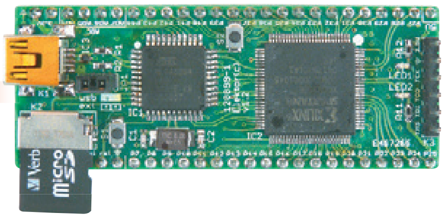
140 Minutes video presentation and more

5 DVD Feedback in Audio Amplifiers

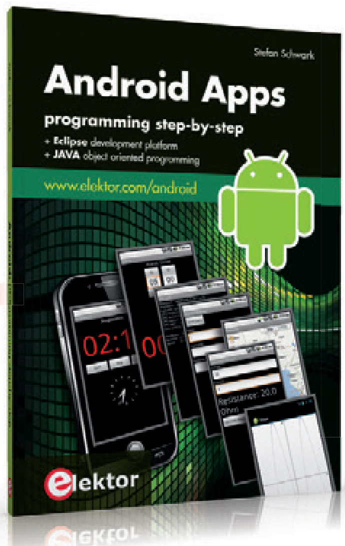
In this Masterclass we address several aspects of feedback in audio amplifiers. The focus of this Masterclass, although not entirely math-free, is on providing insight and understanding of the issues involved.

Presenter Jan Didden provides a clear overview of the benefits that can be obtained by feedback and its sibling, error correction; but also of its limitations and

6

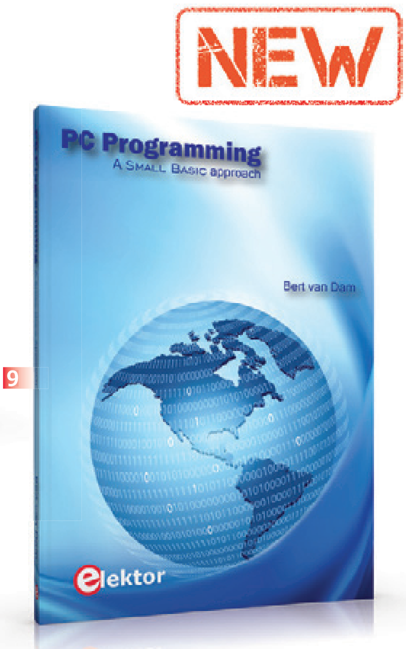


7



8

9



disadvantages. Recommended to audio designers and serious audio hobbyists!

ISBN 978-907920-16-5

£24.90 • € 29.95 • US $40.20

Taming the Beast

6 FPGA Development Board

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. This FPGA development board (designed in the Elektor Labs) shows how easy it is for any electronics enthusiast, whether professional or amateur, to work with these programmable logic devices.

Module, ready build and tested

Art.# 120099-91

See www.elektor.com/fpgaboard

Programming step-by-step

7 Android Apps

This book is an introduction to programming apps for Android devices. The operation of the Android system is explained in a step by step way, aiming to show how personal applications can be programmed. A wide variety of applications is presented based on a solid number of hands-on examples, covering anything from simple math programs, reading sensors and GPS data, right up to programming for advanced Internet applications. Besides writing applications in the Java programming language, this book also explains how apps can be programmed using Javascript or PHP scripts. When it comes to personalizing your smartphone you should not feel limited to off the shelf applications because creating your own apps and programming Android devices is easier than you think!

244 pages • ISBN 978-1-907920-15-8

£34.95 • € 39.95 • US $56.40

MIFARE and Contactless Cards in Application

8 RFID

MIFARE is the most widely used RFID technology, and this book provides a practical and comprehensive introduction to it. Among other things, the initial chapters cover physical fundamentals, relevant standards, RFID antenna design, security considerations and cryptography. The complete design of a reader's hardware and software is described in detail. The reader's firmware and the associated PC software support programming using any .NET language. The specially developed PC program, "Smart Card Magic.NET", is a simple development environment that supports sending commands to a card at the click of a mouse, as well as the ability to create C# scripts. Alternatively, one may follow all of the examples using Visual Studio 2010 Express Edition. Finally, the major smart card reader API standards are introduced. The focus is on programming contactless smartcards using standard PC/SC readers using C/C++, Java and C#.

484 pages • ISBN 978-1-907920-14-1

£44.90 • €49.90 • US $72.50

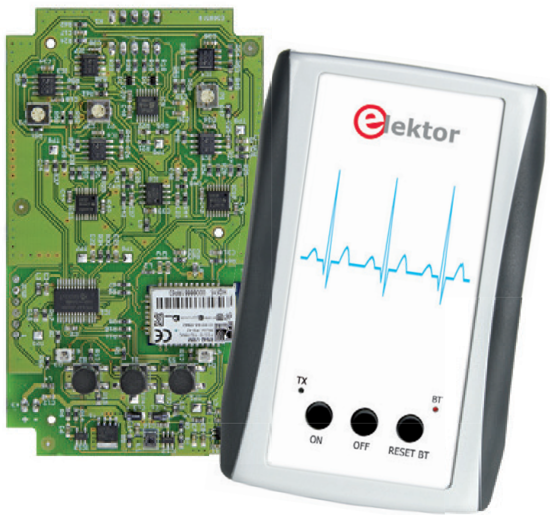
A Small Basic approach

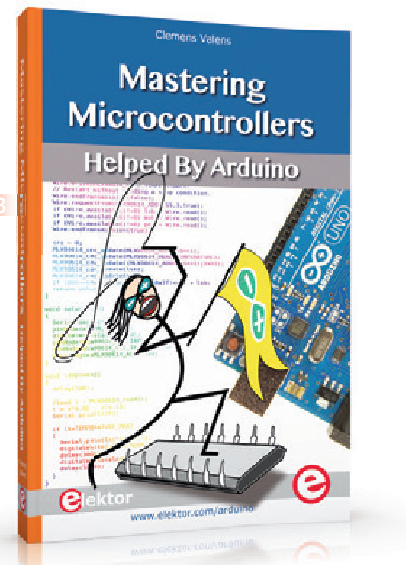
9 PC Programming

There are many different PC programming languages available on the market. They all assume that you have, or want to have, a knack for technology and difficult to read commands. In this book we take a practical approach to programming. We assume that you simply want to write a PC program, and write it quickly. Not in a professional environment, not in order to start a new career, but for plain and simple fun... or just to get a task done. Therefore we use Small Basic. You will have an application up and running in a matter of minutes. You will understand exactly how it works and be able to write text programs, graphical user interfaces, and advanced drivers.

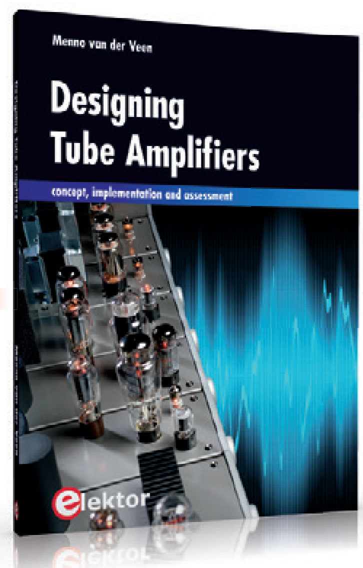
194 pages • ISBN 978-1-1-907920-26-4

£29.50 • € 34.50 • US $47.60

10


13


11


12


Wireless and button-free

10 Android Elektorcardioscope

Instructive, fascinating, and potentially useful to everyone: perform your own electrocardiograms on your Android smartphone or tablet! The project involves skillfully combining a small PIC interface to control an analog input stage with a great deal of software.

Our ECG interface is available in the form of a ready-to-use module to which you just have to add four electrodes and an Android application for smartphone or tablet; there's no physical connection between this terminal and the interface, as it uses Bluetooth communication!

Ready assembled board

Art.# 120107-91

See www.elektor.com/elektorcardioscope

110 issues, more than 2,100 articles

11 DVD Elektor 1990 through 1999

This DVD-ROM contains the full range of 1990-1999 volumes (all 110 issues) of Elektor Electronics magazine (PDF). The more than 2,100 separate articles have been classified chronologically by their dates of publication (month/year), but are also listed alphabetically by topic. A comprehensive index enables you to search the entire DVD. What's more, this DVD also contains the entire 'The Elektor Datasheet Collection 1...5' CD-ROM series, with the original full datasheets of semiconductors, memory ICs, microcontrollers, and much more.

ISBN 978-0-905705-76-7

£69.00 • € 89.00 • US $111.30

Concept, implementation and assessment

12 Designing Tube Amplifiers

This book looks at tube amplifiers from more than just a theoretical perspective. It focuses primarily on the design phase, where decisions must be taken with regard to the purpose and requirements of the amplifier, and it addresses the following questions: How do these aspects relate to subjective and objective criteria? Which circuits sound the best, and why? If you want to develop and market an amplifier, what problems should you expect? What are the significance and meaning of measurements? Are they still meaningful, or have they lost their relevance? Thanks to the enormous processing power of computers, we can now measure more details than ever before. How can these new methods be applied to tube amplifiers? Menno van der Veen will give you all the answers!

188 pages • ISBN 978-1-907920-22-6

£29.50 • € 34.50 • US $47.60

Helped By Arduino

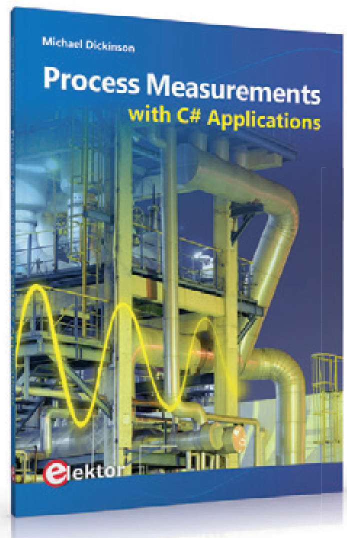
13 Mastering Microcontrollers

The aim of this book is not only to let you enter the World of Arduino, but also to help you emerge victorious and continue your microcontroller programming learning experience by yourself. In this book theory is put into practice on an Arduino board using the Arduino programming environment.

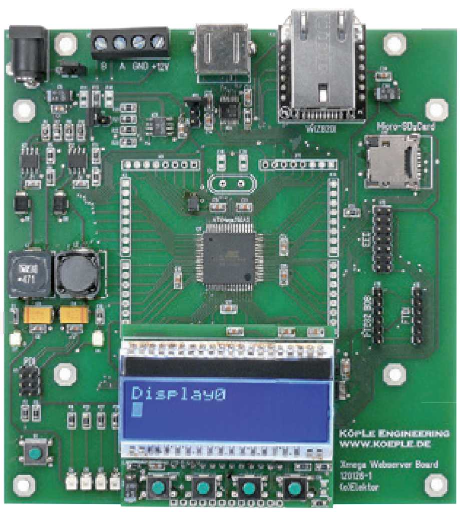
Having completed this fun and playful course, you will be able to program any microcontroller, tackling and mastering I/O, memory, interrupts, communication (serial, I²C, SPI, 1-wire, SMBus), A/D converter, and more. This book will be your first book about microcontrollers with a happy ending!

348 pages • ISBN 978-1-907920-23-3

£34.95 • € 39.95 • US $56.40

15


16


14


Display, SD card, Ethernet, RS-485, buttons and LEDs

14 XMEGA Web Server Board

This microcontroller board is particularly well suited to monitoring and control applications. The plug-in TCP/IP module allows you to implement a web server and other network-oriented applications and a microSD card provides mass storage. Four LEDs, four buttons, and a removable display provide the user interface options. And of course the board comes with a wide range of external interfaces.

Controller Module

Art.# 120126-91

See www.elektor.com/xmega

15 Process Measurements with C# Applications

Measurement is vital to the successful control of any process. This book introduces PC based measurement systems and software tools for those needing to understand the underlying principles or apply such techniques. Throughout the book, the C# programming language is used to give the reader immediate practical desktop involvement. C-Sharp has a wide support base and is a popular choice for engineering solutions. The basics of measurement and data capture systems are presented, followed by examples of software post-processing. Application examples are provided from a range of process industries, with reference to remote monitoring, distributed systems and current industrial practices.

144 pages • ISBN 978-1-907920-24-0

£24.50 • € 27.50 • US $39.50

Ideal reading for students and engineers

16 Practical Digital Signal Processing using Microcontrollers

This book on Digital Signal Processing (DSP) reflects the growing importance of discrete time signals and their use in everyday microcontroller based systems. The author presents the basic theory of DSP with minimum mathematical treatment and teaches the reader how to design and implement DSP algorithms using popular PIC microcontrollers. The author's approach is practical and the book is backed with many worked examples and tested and working microcontroller programs. The book should be ideal reading for students at all levels and for the practicing engineers who may want to design and develop intelligent DSP based systems. Undergraduate students should find the theory and the practical projects invaluable during their final year projects. Similarly, postgraduate students should be able to develop advanced DSP based projects with the aid of the book.

428 pages • ISBN 978-1-907920-21-9

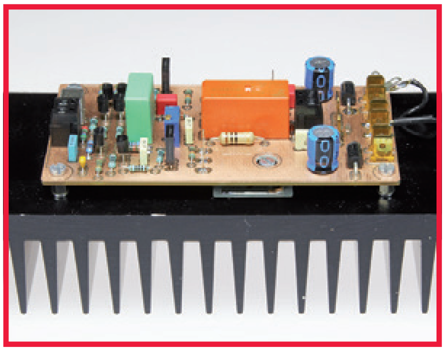
£44.90 • € 49.90 • US $72.50

Further Information and Ordering: www.elektor.com/store
or contact customer service for your region

UK / ROW
Elektor International Media
78 York Street
London - W1H 1DP United Kingdom
Phone: +44 20 7692 8344
E-mail: service@elektor.com

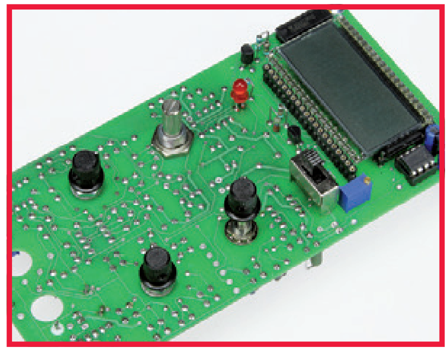
USA / CANADA
Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108 USA
Phone: 860.289.0800
E-mail: service@elektor.com

NEXT MONTH IN ELEKTOR MAGAZINE



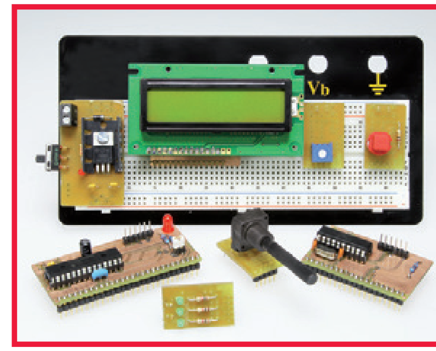
Small Audio Output Stage

A small audio power amplifier with a modest ability but still good specifications is a handy circuit for a variety of audio applications. Just think of an active two-or three-way speaker system, in which the power amplifiers are incorporated in the box together with the power supply. The amplifier is based on a number of classic schematics with proven quality. The design is compact and offers space for your own extensions.



Precision Adjustable DC Current Source

An adjustable current source is a handy tool for testing diodes, zener diodes and LEDs to mention just a few parts. Provided it is sufficiently accurate, the instrument should also allow you to determine the brightness of an LED or to record the voltage-current characteristic of a zener diode. Our instrument has 20 measuring ranges of 10 nA to 20 mA and also has a built-in digital 3½-digit voltmeter with two ranges.



Mini Breadboard Modules

Many electronics designers use a breadboard to build a circuit, so they can experiment extensively. In doing this you often have to replicate the same sub circuits over and over again. That can be avoided by using a number of standard modules that fit on virtually any breadboard. This way you can create a power supply, a microcontroller, or a display, and easily add it to a circuit.

Article titles and magazine contents subject to change, please check www.elektor-magazine.com for updates.
Elektor March 2014 is processed for mailing to US, UK and ROW Members starting February 17, 2014.

See what's brewing @ Elektor Labs 24/7

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

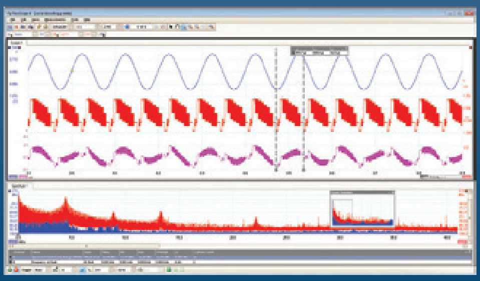
pico Technology

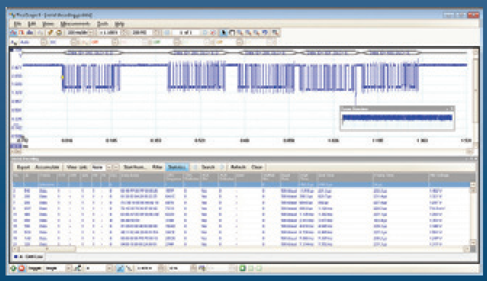
WHY COMPROMISE
SPEED v ACCURACY?
HAVE IT ALL

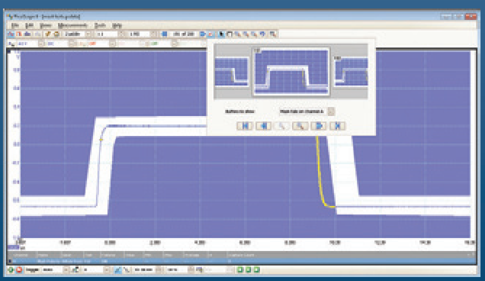
PicoScope	PicoScope 5442A	PicoScope 5442B	PicoScope 5443A	PicoScope 5443B	PicoScope 5444A	PicoScope 5444B
Channels	4					
Bandwidth	All modes: 60 MHz		8 to 15-bit modes: 100 MHz 16-bit mode: 60 MHz		8 to 15-bit modes: 200 MHz 16-bit mode: 60 MHz	
Sampling rate - real time	2.5 GS/s		5 GS/s		10 GS/s	
Buffer memory (8-bit) *	16 MS	32 MS	64 MS	128 MS	256 MS	512 MS
Buffer memory (≥ 12-bit)*	8 MS	16 MS	32 MS	64 MS	128 MS	256 MS
Resolution (enhanced)**	8 bits, 12 bits, 14 bits, 15 bits, 16 bits (hardware resolution + 4 bits)					
Signal Generator	Function generator	AWG	Function generator	AWG	Function generator	AWG

2 Channel models also available * Shared between active channels ** Maximum resolution is limited on the lowest voltage ranges: ±10 mV = 8 bits • ±20 mV = 12 bits. All other ranges can use full resolution.

FLEXIBLE RESOLUTION OSCILLOSCOPE



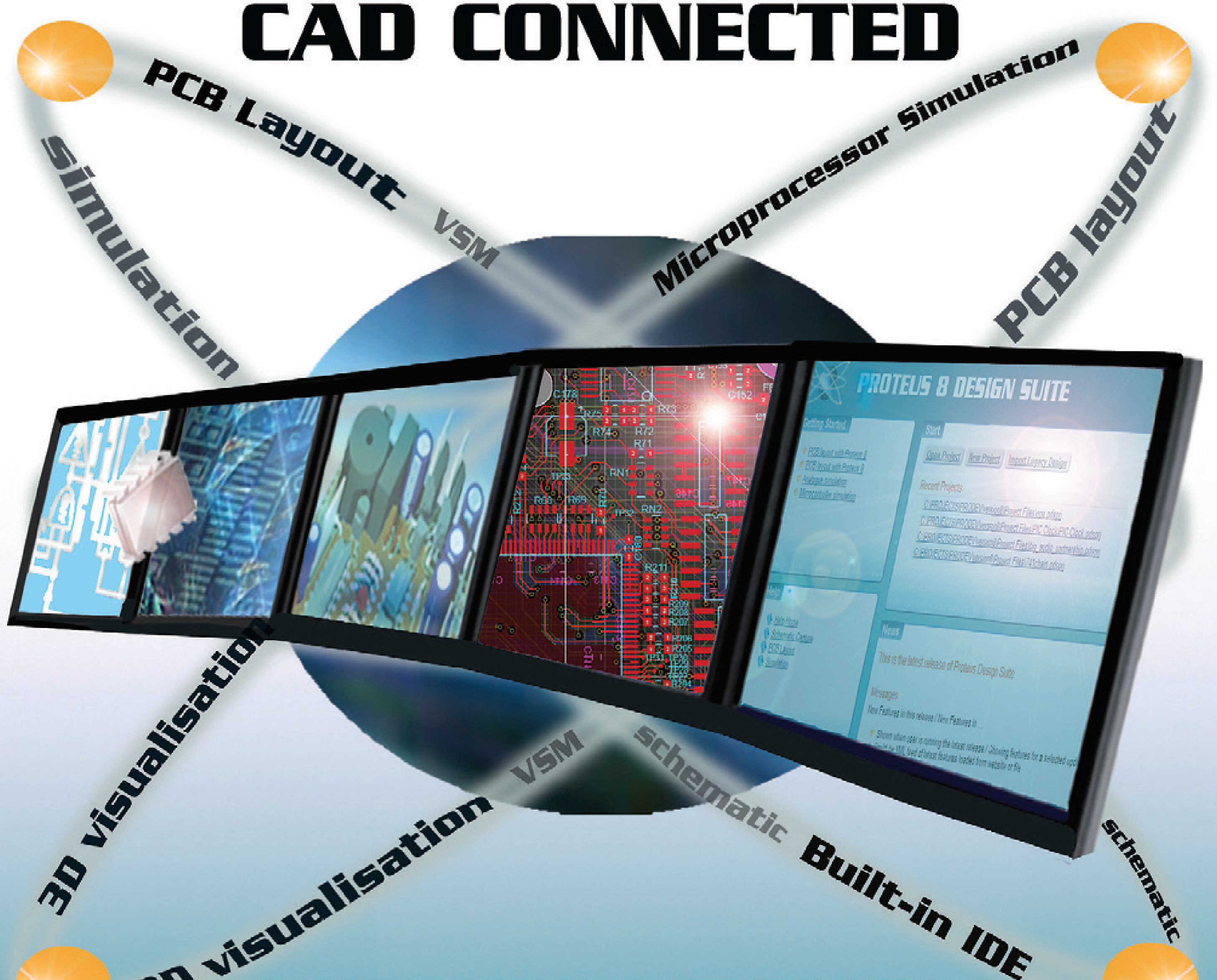




ALL MODELS INCLUDE PROBES, FULL SOFTWARE AND 5 YEAR WARRANTY. SOFTWARE INCLUDES MEASUREMENTS, SPECTRUM ANALYZER, SDK, ADVANCED TRIGGERS, COLOR PERSISTENCE, SERIAL DECODING (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASKS, MATH CHANNELS, ALL AS STANDARD, WITH FREE UPDATES.

www.picotech.com/PS239

CAD CONNECTED



PROTEUS DESIGN SUITE VERSION 8

Featuring a brand new application framework, common parts database, live netlist and 3D visualisation, a built in debugging environment and a WYSIWYG Bill of Materials module, Proteus 8 is our most integrated and easy to use design system ever. Other features include:

. 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 CADCAM, ODB++, IDF & PDF Output.
. Integrated 3D Viewer with 3DS and DXF export.
. Mixed Mode SPICE Simulation Engine.
. Co-Simulation of PIC, AVR, 8051 and ARM MCUs.
. Direct Technical Support at no additional cost.

Version 8.1 has now been released
with a host of additional exciting new features.
For more information visit.
www.labcenter.com

Labcenter Electronics Ltd. 21 Hardy Grange, Grassington, BD23 5AJ.
Tel: +44 (0)1756 753440, Email: info@labcenter.com
Registered in England 4692454

Labcenter Electronics www.labcenter.com