

New series: SDR for the Electronics Lab

eLektor

www.elektor.com

AVR
Software Defined Radio
Generating precision signals
using an ATTiny micro



Android Switch Interface

Using an Android smartphone as a remote control or user interface for your microcontroller projects

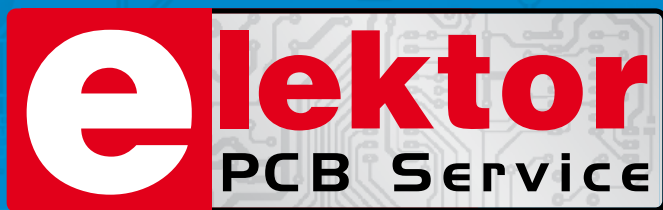
Economic Longwave Receiver

The perfect circuit to learn about & reinvent the pleasure of electronics

Platino in Arduino Land

A guide to integrating custom hardware





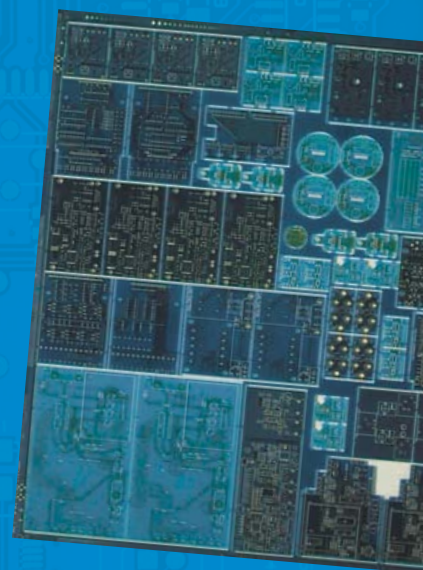
Brought to you by Eurocircuits

The European reference for PCB prototypes and small series

Speed up your design cycle and keep costs down using our fast, easy-access online PCB pooling services.

We deliver:

- Top quality PCBs at low pooling prices
- Fast deliveries to suit your requirements
- Comprehensive technology support
- No set-up or tooling charges
- No minimum order charge - order from 1 PCB
- Online ordering without pre-payment
- Stencil service



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

- 1 or 2 PCBs in 2, 3, 5 or 7 working days
- DRC-checked and fully finished with 2 soldermasks and 1 legend, 150µm technology
- 1 x 100 x 80mm in 7WD - 2 layer 47,02 € - 4 layer 95,52 €
- 2 x 100 x 80mm in 7WD - 2 layer 36,89 € each - 4 layer 74,76 € each

Prices quoted include 21% Belgian VAT but exclude transport costs

STANDARD pool – widest choice of Eurocircuits pooling options

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

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

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

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

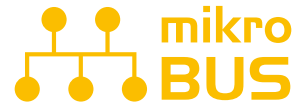
- Single layer Insulated Metal Substrate PCBs
- 1.5mm aluminium base with 75µm thermally conductive dielectric and 35µm Cu foil
- from 3 working days

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

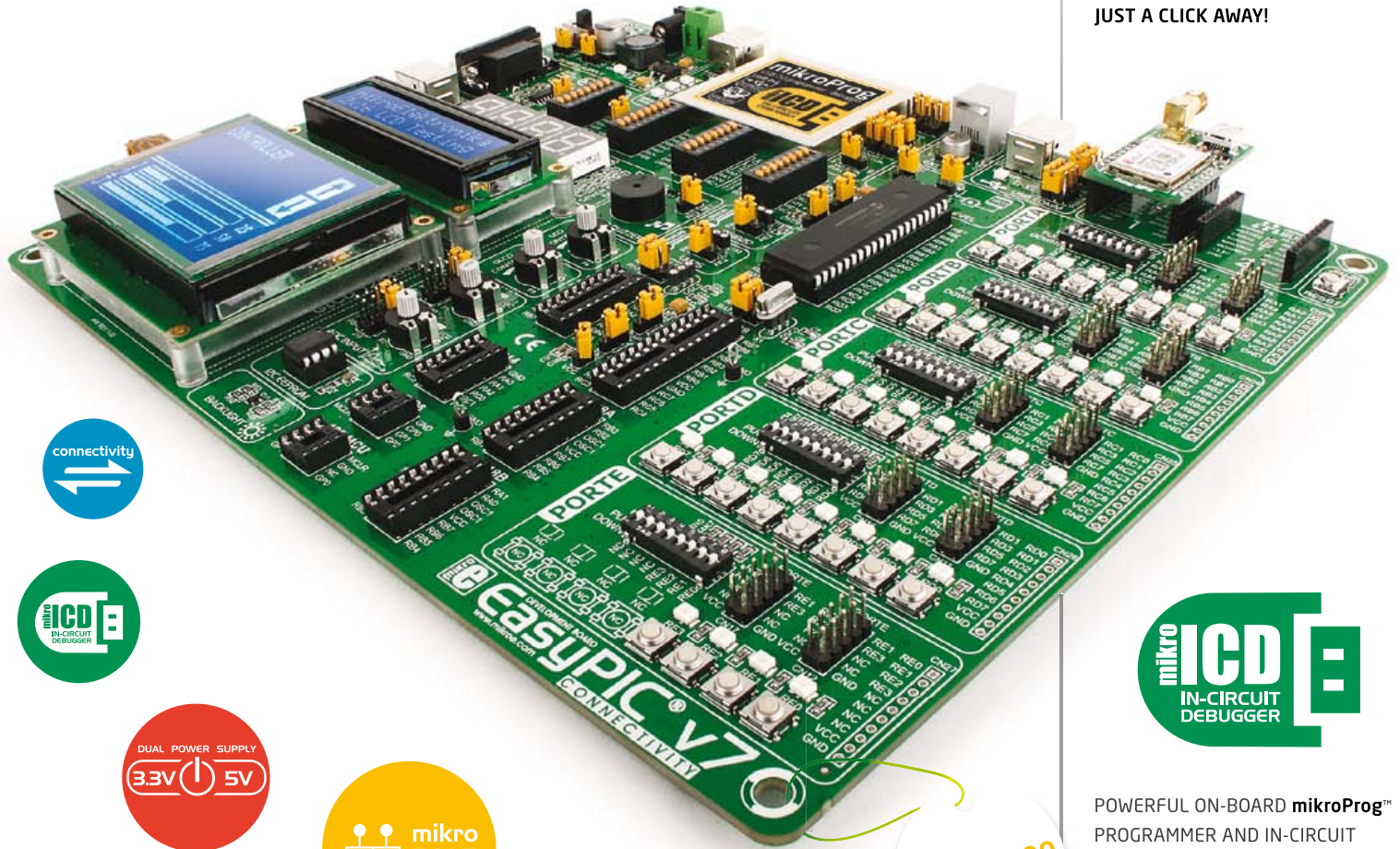
- 1-16 layers down to 90µm technology
- RF and high Tg materials
- from 2 working days

www.elektorPCBservice.com

EasyPIC v7 connectivity



JUST PLUG IN ONE OF YOUR **CLICK BOARD™**, AND IT'S READY TO WORK. SPECIALIZED **mikroBUS™ COMMUNICATION INTERFACE** MAKES YOUR DEVELOPMENT EASIER, AND ALLOWS SIMPLE, YET HIGHLY EFFECTIVE CONNECTIVITY. EVERYTHING IS NOW **JUST A CLICK AWAY!**



connectivity

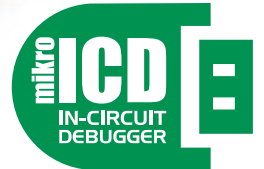


mikro ICD E
IN-CIRCUIT DEBUGGER

DUAL POWER SUPPLY
3.3V 5V

mikro BUS

\$149⁰⁰



POWERFUL ON-BOARD **mikroProg™** PROGRAMMER AND IN-CIRCUIT DEBUGGER CAN PROGRAM ALL **PIC10, PIC12, PIC16 AND PIC18 MICROCONTROLLERS**. OUTSTANDING PERFORMANCE AND EASY OPERATION ARE AMONG IT'S TOP FEATURES. YOU WILL NEED IT, WHETHER YOU ARE A PROFESSIONAL OR A BEGINNER.

DUAL POWER SUPPLY



EASYPIC V7 IS THE ONLY DEVELOPMENT BOARD IN THE WORLD TO SUPPORT BOTH 3.3V AND 5V MICROCONTROLLERS. SMART ENGINEERING ALLOWED US TO SUPPORT **OVER 250 MICROCONTROLLERS** IN A SINGLE BOARD. IT'S LIKE HAVING TWO BOARDS INSTEAD OF ONE!

Our famous PIC development board enters it's 7th generation of development. State of the art design, functionality and quality are it's best features. With four connectors for each port EasyPIC v7 has an **amazing connectivity**. Port headers are logically grouped with their corresponding LEDs and Buttons. Powerful on-board **mikroProg™** In-Circuit Debugger and programmer supports over 250, both **3.3V and 5V devices**. Three types of displays, Serial EEPROM, two Temperature Sensors, Piezo Buzzer, USB connector, RS-232 and FTDI, Oscilloscope GND pins, as well as **mikroBus™ support** make this board an irreplaceable PIC development station.

AAA/A Rating achieved – no laurels awarded

Elektor at the core is a cheerful conglomeration of elecsmiths and wordsmiths, each striving for supremacy in printed matter as well as online activities but always aiming to strike balances between quality and quantity, tech and play, beginner and pro, commercial and non-profit, and nice-to-know and need-to-know. While debating the content of the current edition (and determining its balance) with some colleagues I wondered if Elektor could benefit from a ratings bureau of the Standard & Poor's variety, so overrated these days for their high school grading of whole nations. Having seen a few critical TV documentaries on rating bureaus and their earnings from other people's misery, I thought I'd try a personal rating first of this very edition of Elektor. Without too much trouble and by looking at article titles only I was able to give it an AAAA rating. Here we go.

AVR: not even Atmel's USA head office knows what it stands for, but who cares if one of these ATtiny micros can be made to 'do' Software Defined Radio like we describe on page 16 in the first part of a new series.

Arduino: few people know about the Italian (!) origins of the name, but who cares – it's a great platform that's successfully pulled lots of newcomers into the world of embedded programming and project building. On page 48 Elektor's Platino enters Arduino Land.

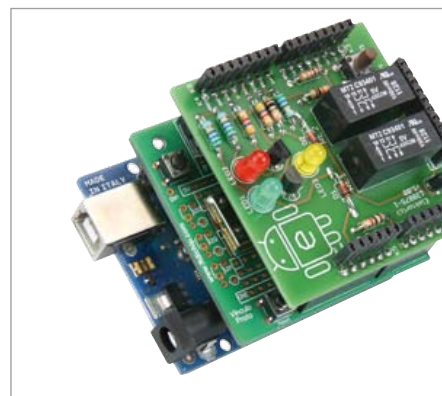
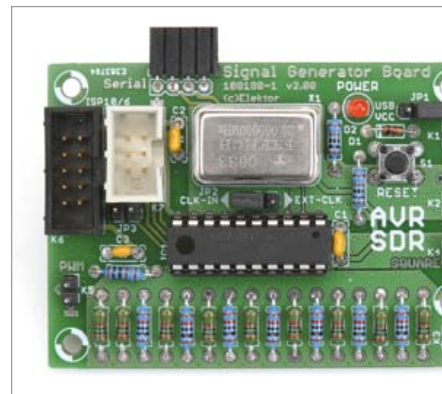
Android & AndroPod: there's about five people left knowing the original Greek (!) word, the rest are buying the product on the premise of "24/7 connectivity with the world". So on pages 24, 36 and 43 you can see what real interfaces for the Android look like and how they work.

In my school days, learning one's ABC meant starting from A you had to reach Z in a sort of orderly fashion and without help from financial people. Today, the whole world seems content with the first letter only, if possible followed by a + sign.

Enjoy reading this edition,
Jan Buiting, Managing Editor



- 6 Colophon**
Who's who at Elektor.
- 8 News & New Products**
A monthly roundup of all the latest in electronics land.
- 14 DesignSpark chipKIT™ Design Challenge**
A global electronics design competition brought to you by Circuit Cellar, Elektor and RS Components.
- 16 AVR Software Defined Radio (1)**
Making a microcontroller 'do' SDR goes beyond receiving any old shortwave station on a PC. Here we kick off a fascinating series starring a simple AVR micro.
- 24 Android Switch Interface**
Smartphones are little computers totally designed for 'connectivity' we are told. And that's exactly what our project aims at: remote control!
- 30 EasyPIC v7**
An extensive review of the latest development board in the EasyPIC series from MikroElektronika.
- 34 Hard Disk Activity Monitor**
This simple circuit provides a visual indicator of that poor old hard disk having a hard time in terms of head and motor activity.
- 36 AndroPod (2)**
In this second instalment we discover the ease of making Elektor's AndroPod respond to HTML and Java.
- 43 E-Labs Inside: AndroPod's 3-way jumper**
An innovative arrangement of the traditional jumper pins on circuit boards.
- 44 E-Labs Inside: Wanted: SMPSU DC-out filtering**
Tuning switch-mode power supplies for cleaner output.
- 44 E-Labs Inside: Plug-o-(d)rama**
A showcase of almost fossil connectors we stumbled upon.



CONTENTS

Volume 38
March 2012
no. 423

16 AVR Software Defined Radio (1)

Atmel AVR microcontrollers are very popular, not least because of the free development tools that are available. In this series we shall show how these processors can be pressed into service for digital signal processing tasks like SDR. Let's generate some signals!



24 Android Switch Interface


In this article we tell you how to implement various wireless sensing and switching functions using an Arduino board with a Bluetooth shield. We also describe how you can program your own Android app for this purpose and what (free) PC software you need for this. Here we discuss smartphone programming in detail, from downloading and installing the software to programming the various components of the desired user interface.

34 Hard Disk Activity Monitor

Even using the most up to date PC with super fast hard disks and a powerful processor it can sometimes happen that while some complex application is running you find yourself staring at the monitor, not quite knowing why the machine is performing slowly or has indeed decided to stop responding altogether. In such situations it would be helpful if an add-on independent 'hardware' gave some insight into what's happening inside your PC. Here it is.

36 AndroPod (2)

To control your own circuitry, you basically need an Android app that includes a tailored user interface. The obvious approach is to use the powerful Android framework and the Java programming language to implement this user interface. However, programming in the Android environment has a very steep learning curve for beginners, so let's try a simpler alternative first like HTML.

- 45 **E-Labs Inside: SMD LED polarity**
Finding the polarity indicator on SMD parts can be tricky as we discovered.
- 46 **E-Labs Inside: The Bus Conductors**
A brief report on a test we ran to see if the ElektorBus can drive 100 feet of Cat5 cable.
- 48 **Platino in Arduino land**
Here we investigate if the Elektor Platino system is compatible in any way with the Arduino platform.
- 54 **Electronics for Starters (3)**
This month we present the basics of the transistor's gain characteristics as well as its amplifier configurations.
- 60 **Economic Longwave Receiver**
Radio started electronics and some say it will eventually put it to bed. Use this circuit to check if there's any truth in this statement.
- 64 **Ground Control for Aircraft HSI**
We found this Horizontal Situation Indicator among vintage aircraft electronics and decided to give it a contemporary application.
- 69 **Low Cost DMX Mixer Desk**
This ultra-simple circuit allows you to remotely control the colour produced by LED spotlights.
- 72 **Component Tips: USB Protection** 
Raymond's Pick of the Month: TPDEUSB30ADRTR and NCP380LSN05AAT1G
- 73 **Hexadoku**
Elektor's monthly puzzle with an electronics touch.
- 74 **Retronics: Elektor's Simple Function Generator (1978)**
Series Editor: Jan Buiting
- 77 **Gerard's Columns: Consulting**
The monthly contribution from our US columnist Gerard Fonte.
- 84 **Coming Attractions**
Next month in Elektor magazine.

The Team

Managing Editor: **Jan Buiting** (editor@elektor.com)

International Editorial Staff: **Harry Baggen, Thijs Beckers, Eduardo Corral, Wisse Hettinga, Denis Meyer, Jens Nickel, Clemens Valens**

Design staff: **Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser, Christian Vossen**

Membership Manager: **Raoul Morreau**

Graphic Design & Prepress: **Giel Dols, Jeanine Opreij, Mart Schroijen**

Online Manager: **Carlo van Nistelrooy**

Managing Director: **Don Akkermans**

The Network




Our international teams

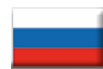
 **United Kingdom**
Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com

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

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

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

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

 **Russia**
Nataliya Melnikova
8 10 7 (965) 395 33 36
nataliya-m-larionova@yandex.ru

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

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

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

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

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

 **South Africa**
Johan Dijk
+27 78 2330 694 / +31 6 109 31 926
j.dijk@elektor.com

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

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

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

Volume 38, Number 423, March 2012 ISSN 1757-0875

Publishers: Elektor International Media, Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, England. Tel. (+44) 208 261 4509, fax: (+44) 208 261 4447 www.elektor.com

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

Elektor is published 11 times a year with a double issue for July & August.

Subscriptions: Elektor International Media, Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, England. Tel. +44 208 261 4509, fax: +44 208 261 4447 Internet: www.elektor.com/subs Email: subscriptions@elektor.com Rates and terms are given on the Subscription Order Form.

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

Distribution: Seymour, 2 East Poultry Street, London EC1A, England. Telephone: +44 207 429 4073

UK Advertising: Elektor International Media b.v. P.O. Box 11 NL-6114-ZG Susteren The Netherlands.

Membership Counter

We
now have

27932

members
in

83

countries.

Not a member yet?

Sign up at www.elektor.com/member



Take out a free subscription to Elektor Weekly

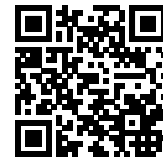
Do you want to stay up to date with electronics and information technology?
Always looking for useful hints, tips and interesting offers?
Subscribe now to Elektor Weekly, the free Elektor Newsletter.

Your benefits:















- The latest news on electronics in your own mailbox each Friday
- Free access to the News Archive on the Elektor website
- You're authorized to post replies and new topics in our forum



Register today on
www.elektor.com/newsletter



Supporting Companies

	<p>AudioXpress www.cc-webshop.com 87</p>		<p>Labcenter www.labcenter.com 88</p>
	<p>Beta Layout www.pcb-pool.com 13</p>		<p>Microchip www.microchip.com. 11</p>
	<p>DesignSpark chipKIT™ Challenge www.chipkitchallenge.com. 15</p>		<p>MikroElektronika www.mikroe.com 3</p>
	<p>Eurocircuits www.elektorpcbsevice.com 2</p>		<p>National Instruments www.ni.com/daq/usb 59</p>
	<p>EzPCB www.siliconray.com 13</p>		<p>Pico Technology www.picoscope.com/PS132 51</p>
	<p>Future Technology Devices, Showcase www.ftdichip.com 78</p>		<p>Renesas Contest www.circuitcellar.com/RenesasRL78Challenge. . . 33</p>
	<p>Jackaltac www.jackaltac.com 9</p>		<p>Schaeffer AG www.schaeffer-ag.de 21</p>

Not a supporting company yet?

Contact Johan Dijk (j.dijk@elektor.com, +27 78 2330 694) no later than 24 January 2012
to reserve your own space for the next edition of our members' magazine

Telephone: +31 46 4389444, Fax: +31 46 4370161
Email: j.dijk@elektor.com
Internet: www.elektor.com
Advertising rates and terms available on request.

Copyright Notice

The circuits described in this magazine are for domestic use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, software carriers and article texts published in our books and magazines

(other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning and recording, in whole or in part without prior written permission from the Publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The submission of designs or

articles implies permission to the Publisher to alter the text and design, and to use the contents in other Elektor International Media publications and activities. The Publishers cannot guarantee to return any material submitted to them.

Disclaimer

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

© Elektor International Media b.v. 2012

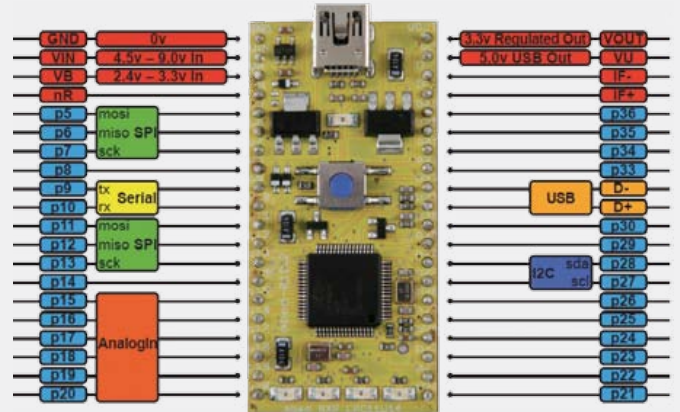
Printed in the Netherlands

RS Components introduces new NXP ARM Cortex-M0 based mbed development board

RS Components has announced availability of the new NXP LPC1114U24 32-bit ARM® Cortex-M0 based mbed embedded development board for low-risk and rapid professional prototyping of microcontroller-based systems.

The NXP LPC1114U24 microcontroller is particularly suitable for prototyping low-cost USB devices, battery-powered applications and 32-bit ARM® Cortex-M0 based designs. It is packaged as a small DIP form-factor convenient for prototyping with through-hole PCBs, stripboards and breadboards, and includes a built-in USB drag 'n' drop flash programmer. The mbed microcontrollers are supported with an online tools platform available at both mbed.org and designspark.com that provides experienced embedded developers with a productive environment for building proof-of-concepts.

For developers new to 32-bit, mbed is an accessible way to build projects with the backing of libraries, resources and support shared in the mbed.org developer community, including: a lightweight online compiler for instant access to a working environment on Windows, Linux or Mac OS X; a C/C++ SDK (Software Development Kit) for high-level programming of peripherals; in addition to a wealth of libraries and code examples. The NXP LPC1114U24 MCU runs at 48 MHz and device peripherals



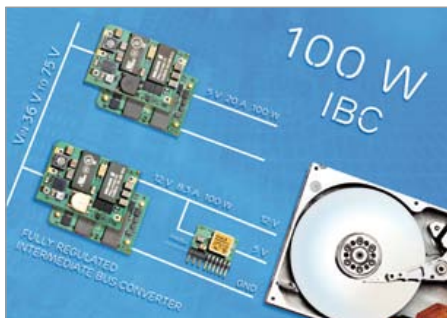
include 32 KB of flash memory, up to 8kB of SRAM data memory, configurable Full-Speed USB 2.0 device controller; one Fast-mode Plus I2C-bus interface, two SPI interfaces, UART, six ADCs (Analogue-to-Digital Converters) and 54 GPIO (General-Purpose I/O) pins.

LPC1114U24 mbed development board is available to order now from RS Components' stock at www.rs-components.com. Support and resources are available at mbed.org and DesignSpark.com.

www.mbed.org www.designspark.com
(120120-IX)

Sixteenth-brick power converters ideal for low-profile applications

Ericsson has introduced two new isolated DC/DC power converters that come in an industry-standard sixteenth-brick form factor and deliver an output current up to 20A, or an output power equivalent to 100 W. The new PKU-C series has been developed to respond to the increased demand in the mid-range IT and communications power



applications that are migrating from multiple-output power modules to fully regulated intermediate bus converters, and require compact DC/DC modules that save space for core components, in addition to

providing a height of less than 9 mm for low-profile applications.

Designed to operate with a 36 V to 75 V input voltage range, the PKU-C series is primarily designed for applications that are powered by a centralized 48 V bus. However, due to its efficient and flexible power performance, the series is also suitable for a wide range of applications including those in industrial, process control and automation. A key application for the devices is decentralized local data-storage equipment that requires 12 V and 5 V to power disk drives. Systems designers of these applications gain significant design freedom with the combination of the new 12 V PKU4104C and 5 V PKU4105C. Traditionally mid-power applications, and especially decentralized disk-drive storage systems, have used multiple-output DC/DC converters, but cost-optimization and shorter time-to-market demands, in the high-power segment, have led to the migration to intermediate bus voltage architectures.

Because the PKU-C devices are isolated and feature a floating output, the two modules can also be combined to power the 12 V and 5 V rails in a disk-drive-array application requiring higher power than 100W. Additionally, designers can combine the

12 V PKU4104C with Ericsson's BMR462 3E* digital voltage regulator to generate the 5 V rail in single-disk applications.

Available in an industry-standard sixteenth-brick footprint, with dimensions of 33 x 22.9 with a low profile height of only 8.5 mm (1.3 x 0.9 x 0.334 in), the devices deliver up to 100W, and provide 92.7% efficiency at 50% load and 12V output, guaranteeing low power dissipation and reduced energy consumption. The PKU-C series also delivers outstanding thermal performance for applications that have limited airflow. The series delivers full performance up to 80 degrees C with a 3.0 m/s airflow, and, in worse case conditions, full output power with natural convection up to 65 degrees C. Offering an MTBF (Mean-Time-Before-Failure) of 4.27 million hours, the PKU-C devices meets safety requirements according to IEC/EN/UL 60950-1, and offers input/output isolation of 1500V(DC) and various protection features, including Output-over-Voltage Protection (OVP), input under voltage shutdown, Over-Temperature Protection (OTP), output short-circuit protection and pre-biased output. The PKU-C series is available in both through-hole (PI version) and surface-mount (SI version).

www.ericsson.com/powermodules
(120120-VII)

New Cypress CapSense® Express™ 4x4 matrix system for keypad applications

Cypress Semiconductor Corp. announced the new CY8CMBR2016 device that enables designers to implement capacitive matrix button systems of up to 4x4 buttons without having to write firmware or learn new software tools. A matrix system is a row and column array of buttons typically found in keypad applications in a variety of industrial applications. This is the latest addition to the popular CapSense® Express™ capacitive touch-sensing controller line from Cypress that leverages Cypress's revolutionary SmartSense™ Auto-tuning algorithm, which completely eliminates the requirement for system tuning and is the only solution that offers run-time environmental compensation.



The CY8CMBR2016 supports industry standard 'keypad scan' and 'truth table' host interface protocols, enabling customers to leverage their existing host processor firmware. The device also features multi-touch, perfect for matrix applications that require simultaneous button presses to enter various user interface modes. The CY8CMBR2016 operates over a 1.71 V to 5.5 V range making it ideal for a wide range of regulated and unregulated battery applications, and a supply current in run mode as low as 15 uA/button. It is targeted for applications such as security panels, card reader keypads, biometric scanner keypads, fire

alarm control panels, thermostats, home appliances as well as any system requiring up to 16 individual CapSense buttons. More information is available at the url below.

SmartSense Auto-tuning dynamically optimizes the capacitive baseline and detection threshold for each button. It adjusts for the optimal capacitance sensing range at power-up and during runtime as environmental conditions change, including temperature, humidity, and noise. Eliminating the need to tune is a significant advantage for large and small manufacturers alike, as it saves engineering time and yield loss that can occur with even slight variations in manufacturing tolerances. This savings is greatly multiplied for customers with a global factory footprint and supply chain. SmartSense Auto-tuning can eliminate the need for additional test steps required by competing solutions to address variations in PCBs and overlays.

The device offers reliable operation in the harshest sensing conditions and ensures superior immunity to conducted and radiated noise. It also integrates a voltage regulator to address power supply noise as well as filters for any spurious noise.

Cypress's accompanying design toolbox provides detailed resources to ensure optimal CapSense performance, and advanced system debug features allow customers to take designs directly to production for significantly shorter time-to-market.

www.cypress.com/go/capsense
(120202-1)

First ACE Selective Soldering Workshop a sellout; more planned for Q2

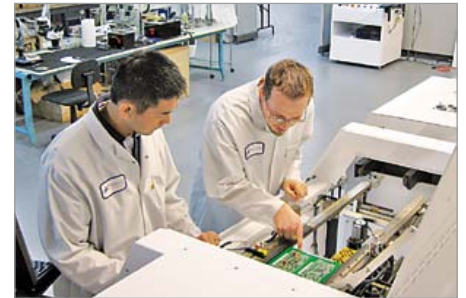
It took barely a week, but the first Selective Soldering workshop planned by ACE Production Technologies sold out; now Alan Cable, President, is planning more. "We

expected our program to be well-received, but quite frankly, we didn't expect it to sell out so quickly," he said.

The early March 2012 workshop is intended to be the first in a series of comprehensive hands-on 2-day workshops designed to educate and enhance an attendee's working knowledge of the selective soldering process. The workshops were instituted in response to many requests from customers and friends, and are held at the ACE factory in Spokane, Washington, USA.

"Over the past decade the selective soldering process has proven to provide significant labor savings over hand soldering, as well as superior quality. However, one has much to learn in order to master this process and reap the maximum benefits from it," Cable adds.

Well-known process expert Bob Klenke of ITM Consulting conducts the classroom portion of the workshop. "Bob is well respected as an expert in all soldering techniques with



emphasis on the selective soldering process," Cable says. "The day is divided evenly into halves; Bob teaches for one half of the total time, and the balance is spent in the ACE lab working with our expert staff. This provides the right combination of instruction and hands-on experience that attendees need." Hand-outs and reference materials are also provided.

This workshop is intended for equipment operators, shop floor technicians, process engineers as well as manufacturing and quality personnel who are experienced in SMT and through-hole assembly who want to further their expertise in implementing

Advertisement



**PCBs
Muuuuch Cheaper...**

No-frills policy

16.94 EURO*

5 pcs, 100 mm x 100 mm
*per piece, incl. VAT (21%)
+ shipping costs e. g. Germany 10.71 EURO



www.jackaltac.com

and troubleshooting; More importantly, it will give attendees a perspective on how to implement and optimize the selective soldering process in the most effective manner. The workshop teaches the essentials of selective soldering and describes process variables serving as guidance to enhance the flexibility, reliability and quality provided by the selective soldering equipment. The curriculum is based on real-world practice defining the proper understanding of component limitations, clearance restrictions, thermal requirements and solder joint reliability issues that will insure complete knowledge of the selective soldering process.

www.ace-protech.com
 Contact: cgoodell@ace-protech.com
 (120202-II)

Intersil: industry's lowest voltage, micropower RS-485/RS-422 transceiver family

Intersil Corporation's new ISL3261x series of RS-485/RS-422 transmitters and receivers are designed to meet the strict power budgets of battery and remote sensing applications. Operating from an industry-low 1.8 V supply voltage and consuming less than 150 μ W, this new family is the optimal solution for designers who need to increase battery life and/or reduce the size of their power supply.



The ISL3261x series includes the ISL32613/14 transmitters and the ISL32610/11/12 receivers. They each operate with voltage ranges from 1.8 V to 3.6 V, and data rates that reach 256 kbps. All of the devices in the series provide 16.5 kV ESD protection and operate at up to 125 degrees C, making them ideal for harsh environments and battery-powered or

solar-powered sensor applications. The ISL32613E/14E transmitters consume up to two orders of magnitude less current when operating at 1.8 V versus 3.6 V. The ISL32610/11/12E receivers feature symmetrical switching points (± 200 mV) and increased hysteresis. The symmetrical switching points eliminate duty cycle distortion introduced by 'full-failsafe' type receivers, while the larger hysteresis increases noise immunity. Ultra-low bus currents of ± 40 μ A allow more than 256 transmitters on a network, with no repeaters, while still meeting the RS-485 specification's 32-unit load maximum. Device quick features and specifications:

- Low quiescent current of 80 μ A max

for transmitters and 110 μ A max for receivers;

- Wide supply range of 1.8 V to 3.6 V for maximum flexibility;
- High 16.5 k ESD protection for enhanced reliability in harsh environments;
- Industrial temp range as well as 125 degree C extended high temp.

The ISL3261x series transmitters and receivers are available in space saving SOT-23 packages.

www.intersil.com/products/deviceinfo.asp?pn=ISL32613E
 www.intersil.com/products/deviceinfo.asp?pn=ISL32610E
 (120202-III)

Configurable power supplies up to 2500 watts

XP Power's fleXPower X15 and XM15 series of multiple output configurable AC-DC power supplies should suit high power medical and IT/industrial applications. The X15 series complies with the internationally recognized EN/UL 60950 safety specification for IT and industrial applications. Meeting the demanding requirements of the latest 3rd edition medical safety standards, and having low leakage, the XM15 series suit applications in medical diagnostic appliances, CAT scan and MRI equipment.

Capable of delivering 1500 W output at low line input, and 2500 W output at high line (>180 VAC input), these compact fan cooled units measuring 279.4 x 127 x 127 mm (11 x 5 x 5 inches) have a power density of up to 9.09 watt per cubic inch. The mechanical chassis approach of the fleXPower X15 and XM15 series provide 20 slots for arranging 2- or 3-slot output modules. The modules, available with single or dual outputs,



cover all the popular nominal output voltages from +3.3 to +60 VDC in a variety of output current ratings. The ability to parallel up outputs for more power, or use modules in series for non-standard output voltages offer design engineers thousands of different output combinations. An auxiliary 5 V / 1 A always-on output is available to power logic or control circuits

in the end system without the need for any additional voltage source or step down converters. Monitor and control signals include AC OK, Power Fail, DC OK, Global Inhibit, Fan Fail and Module OK / Inhibit.

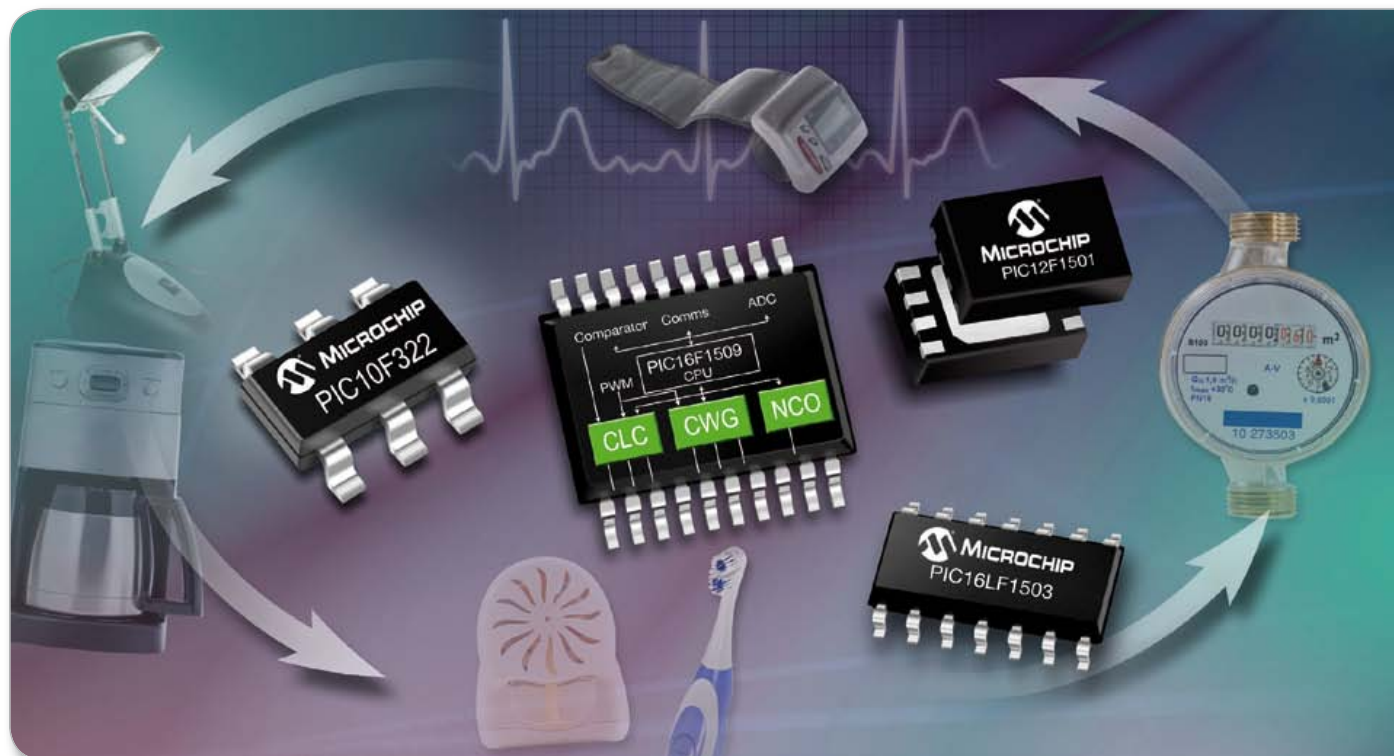
An optional fan speed control is available to order that reduces fan speed, and its associated noise, at lower power levels. The fleXPower series is available in

six power levels from 400 W to 2500 W and offers a high efficiency power source in a configurable format for fast time to market.

The fleXPower X15 and XM15 are available from approved regional distributors, or direct from XP Power and come with a 3-year warranty.

www.xppower.com
 (120202-V)

New 8-bit Microcontrollers with integrated configurable logic in 6- to 20-pin packages



Microchip's new PIC10F/LF32X and PIC12/16F/LF150X 8-bit microcontrollers (MCUs) let you add functionality, reduce size, and cut the cost and power consumption in your designs for low-cost or disposable products, with on-board Configurable Logic Cells (CLCs), Complementary Waveform Generator (CWG) and Numerically Controlled Oscillator (NCO).

The Configurable Logic Cells (CLCs) give you software control of combinational and sequential logic, to let you add functionality, cut your external component count and save code space. Then the Complementary Waveform Generator (CWG) helps you to improve switching efficiencies across multiple peripherals; whilst the Numerically Controlled Oscillator (NCO) provides linear frequency control and higher resolution for applications like tone generators and ballast control.

PIC10F/LF32X and PIC12/16F/LF150X MCUs combine low current consumption, with an on-board 16MHz internal oscillator, ADC, temperature-indicator module, and up to four PWM peripherals. All packed into compact 6- to 20-pin packages.

FAST-START DEVELOPMENT TOOLS



PICDEM™ Lab Development
Kit - DM163045



PIC16F193X 'F1' Evaluation
Platform - DM164130-1



PICkit™ Low Pin Count Demo
Board - DM164120-1

Free CLC Configuration Tool:
www.microchip.com/get/euctool

Go to www.microchip.com/get/eunew8bit to find out more about
low pin-count PIC® MCUs with next-generation peripherals

microchip
DIRECT
www.microchipdirect.com

www.microchip.com

MICROCHIP

Apem: Controlmec™ pushbutton switch

APEM Components introduces Controlmec™, a five position integrated switch solution. Designed for easy activation in indoor and non-dusty applications, Controlmec can be used to control a unit, navigate in a display, or to operate indoor auxiliary equipment on a vehicle or boat. The Controlmec can also be supplied as an IP67 sealed solution.



Controlmec is a single pole/momentary switch, rated for a lifetime of 10 million operations. The solid cap has a temperature range of -40 to +65, while the LED operates from -40 to +85°C. It is available in nine cap colors and eight LED colors or color combinations.

An extension of the industry-standard Navimec™ system, the Controlmec includes one complete solid cap, a unit panel plate of the customer's design, and a printed circuit board (PCB) including the switches. The PCB with the switches is assembled according to MEC specifications and mounted to the back side of the front panel. Configurations can be made with through-hole or surface mount device (SMD) switch versions, provided they are accurately positioned.

www.apem.com
(120202-IV)

Multi-output regulators with supervisory and watchdog timers

Analog Devices, Inc. (ADI) is continuing to help industrial, medical and communications equipment designers improve power system performance by reducing board space with the introduction today of the ADP5041 and ADP5040 multi-output regulators. The regulators meet the increasing demand for greater power density by combining a high-efficiency, 3-MHz, 1.2-A buck regulator and



two 300-mA LDOs (low dropout regulators) in a small 20-lead LFCSP package. Unlike discrete power solutions that require up to 14 components and 126 mm² of board space, these highly integrated regulators provide a more integrated circuit solution using only nine components and 50 mm² of board space, to increase performance and reliability and lower system cost

The ADP5041 and ADP5040 are ideal power management companions to mid-range FPGAs, microprocessor and DSP systems where core, IO and memory voltages are necessary. The ADP5041 regulator's on-chip watchdog timer provides greater reliability by monitoring code execution integrity in processor-based systems and resetting the processor if it fails to strobe within a preset timeout period. The ADP5041 also features a high accuracy ($\pm 1.5\%$ over temperature) reset generator that can be externally programmed to monitor low voltage power supply rails. A wide range of ordering options also is available to address multiple reset and watchdog timings.

The ADP5041 has a special circuit that detects a three-state condition when applied to the watchdog refresh input at the WDI pin typically controlled by a processor/DSP output port. When the processor sets this port in three-state mode, the watchdog refresh timer is disabled, preventing a watchdog reset to the processor. This feature is important when supporting processor/DSP sleep operation where the core is disabled and watchdog timer cannot be refreshed.

The ADP5040 and ADP5041 multi-output regulators reduce thermal dissipation by using high-efficiency switching regulators with up 96 percent buck power efficiency. For low-noise analog circuit applications, the LDOs maintain a power supply rejection greater than 60 dB for frequencies as high as 10 kHz while operating with a low head-

room voltage. The ADP5041 and ADP5040 also provide a three-rail system power supply (1.2 A buck regulator and two 300-mA LDOs) with adjustable output voltages, which allows output voltages to be easily set using an external resistor divider network. The 3-MHz buck regulator switching frequency allows small ceramic inductors to be used to further reduce solution size and cost. These features allow the ADP5041 and ADP5040 to be easily and quickly modified for a variety of applications with short product design schedules such as portable medical and industrial devices.

www.analog.com/ADP5041
www.analog.com/ADP5040
(120202-VIII)

World's thinnest waterproof piezoelectric speaker

Murata today has commenced mass production of the ultra thin VSLBG2216E waterproof piezoelectric speaker. With a thickness of just 0.9 mm it is believed to be the world's thinnest IPX7 speaker. Designed for consumer electronics applications, it is unlike other similar devices in that it does not require any additional waterproof membrane to be applied during end-product manufacture. Such additional membranes increase product time, costs and impacts the audio output quality of the speaker. The VSLBG2216E series offers a flexible, space saving and cost effective design together with a higher quality of music reproduction. By using a piezoelectric element, without any magnet, ensures that the speaker is free of the ingress of metallic particles and has no electromagnetic noise that might affect other magnetic sensors that are incorporated into the product.

Typical applications include mobile phones, tablet computers, e-book readers, digital cameras and portable music players.

www.murata.eu
(120202-VI)





- Boards • Kits • Modules • Components • Tools • Instruments
- FOR
- Companies • Professionals • Students • DIYers • Amateurs



USB STC 8051 MCU Programmer
Free For 8051 Developers

\$0

Add to cart



SEN1060 GPS Module With Antenna
Small, Low Cost, Easy To Develop

\$50

Add to cart



NRF2401 2.4G Transceiver Module
Small, Versatile, Simple, Ultra Low Power

\$7.5

Add to cart



LM3886 68Wx2 HiFi Audio Amplifier
HiFi Sound At Affordable Price

\$49.9

Add to cart

Simplify your electronics projects by visiting
<http://www.siliconray.com>



THE ORIGINAL SINCE 1994
PCB-POOL
Beta LAYOUT

Free Stencil
Get a free SMD laser stencil
with every Prototype order

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

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



PCB-POOL® is a registered trademark of

www.pcb-pool.com



Fascinated by technology's impact on the future?

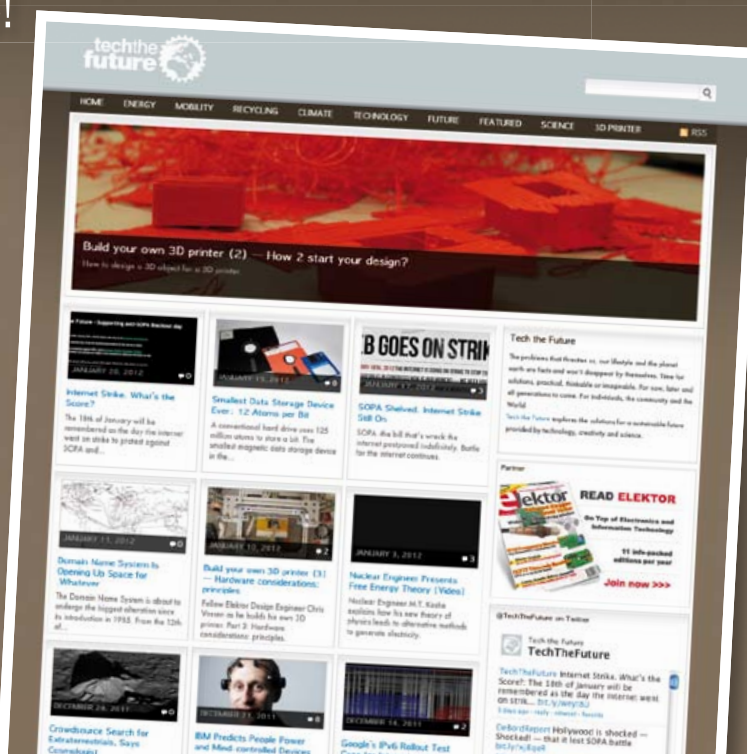
Check out Tech the Future!

Computing power and global interconnectivity are pushing tech innovation into overdrive.

Pioneering technologies and creative workarounds affect even the couch potato 24/7. Tech the Future reports on technology strides that shape the future — yours included.

www.techthefuture.com

Follow Tech the Future



DesignSpark chipKIT™ Design Challenge



Now in its final month, DesignSpark chipKIT™ Design Challenge for energy-efficient applications is still open for submissions... so, if you're quick...

By Ian Bromley (UK)

The DesignSpark chipKIT competition is now in its final weeks, and there remains just enough time to enter for a chance to win a prize. Total cash prizes of \$10,000 are there to be won, including a first prize of \$5,000. The response and enthusiasm for the competition has been fantastic and we've been delighted by the high quality of the submitted ideas. As a reminder of what has gone before, or for those hearing about this for the first time, the DesignSpark chipKIT challenge is all about encouraging engineers, students and hobbyists to develop new and innovative energy-efficient solutions, while also maintaining an eco-friendly footprint.

So, as the competition approaches its climax, entrants will now be well advanced in their designs for energy-efficient and environmentally friendly applications based on the chipKIT™ Max32™ Arduino-compatible development platform from Digilent. The kit features Microchip's 32-bit PIC32 microcontroller and enables developers to easily and inexpensively integrate electronics into their projects. Just to wet your appetite for the smorgasbord of ideas we've had, here is just a few of them we've received — which are also available at the online DesignSpark community at www.designspark.com.

One very interesting project is the development of an unmanned underwater vehicle (UUV) that employs gliding as its method of propulsion, enabling the UUV to operate for an extended period of time due to its extremely low power consumption. An extender board being developed for the vehicle will interface to the sensors and actuators required for operation of the UUV. The design will include a 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer for underwater orientation, in addition to a GPS receiver to determine global position on the surface. A buoyancy engine, together with pitch and roll actuators will control the UUV's movement through the water. In addition, a CTD sensor will measure conductivity, temperature and depth, with the data being recorded in Flash memory on board the UUV, enabling profiling of salinity characteristics.

Another project is a hydroponics water and nutrient-control system that will monitor climatic conditions to determine the level of nutrient-feed requirements for plants in a hydroponic environment,

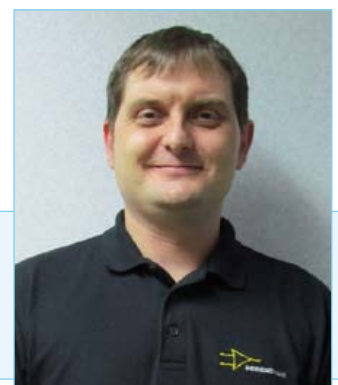
thereby saving on water usage and nutrient chemicals. As most hydroponics farms currently use established feed regimes that run without adjustment for environmental conditions, the design could deliver both financial and water savings.

A 'smart garden' project for the control of the natural cycle of a plant with minimal human intervention is yet another innovative entry. The project aims to combine renewable energy sources — such as solar power — with low energy consumption of the board control circuitry operating at significantly reduced power. The system will include various sensors such as those for humidity, temperature and light, and actuators such as an irrigation pump. Additionally, the modular system will also be able to be expanded, for example, adding Bluetooth technology to enable user notifications — a possibility created by the on-going 'Internet of Things' revolution — about which we expect to be hearing a great deal more this year. And these are just three of many highly innovative on-going projects. As a quick final reminder on the competition, all entries must include an extension card developed using RS' free-of-charge and award-winning DesignSpark PCB software tool with code compiled using Microchip's MPLAB® IDE software. Entrants are also strongly encouraged to engage and interact with other members of the online DesignSpark community by posting information on their projects, providing updates on progress, and sharing comments and ideas on their respective designs. Participants will automatically qualify for entry into bonus Community Choice Awards, in addition to admission into spot prize draws for the best collaboration to win vouchers exchangeable for products ordered from RS Components/ Allied Electronics.

The competition entries will be judged on the level of energy efficiency and the quality of the extension card's PCB design. Entries are due on March 27, 2012 and the winners will be announced in April 2012.

(120188)

Further details and registration for the DesignSpark chipKIT™ challenge are available at: chipkitchallenge.com.



Ian Bromley is a Technical Marketing Engineer at RS Components and the Project Manager for the DesignSpark PCB software tool. Prior to working for RS, Ian worked for many years as a design support consultant with Texas Instruments, in addition to working as a field applications engineer immediately following his graduation in 1994 with an honours degree in microelectronic engineering.



DESIGNSPARK

Deadline for Entries: March 27, 2012

Turn a **hot**
idea into a
cool
solution.

DesignSpark chipKIT™ Challenge

Get ready to win your share of \$10,000 in cash prizes!

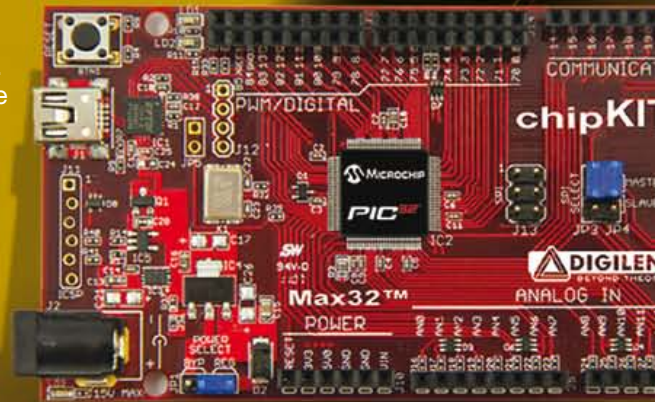
It's time to see if your hard work and superior engineering skills have paid off. The deadline for the **DesignSpark chipKIT™ Challenge** is just around the corner. It's time to finalize your design and prep your entries for the judges!

Will your design change the world? Reduce power consumption? Improve energy efficiency? There's only one way to find out.

Manage your project entry by clicking on the 'My Project' tab at www.designspark.com/chipkitchallenge-projects/latest. Be sure to upload, and clearly label, all materials necessary for judging your entry including an abstract, complete documentation, and source code.

For more information and tips on how to enter, visit www.designspark.com/chipkitchallenge/faq.

Don't delay! The DesignSpark chipKIT™ Challenge ends on March 27, 2012 at 18.00 GMT (13.00 EST).



Visit www.chipkitchallenge.com

for complete rules and details.

IN ASSOCIATION WITH:

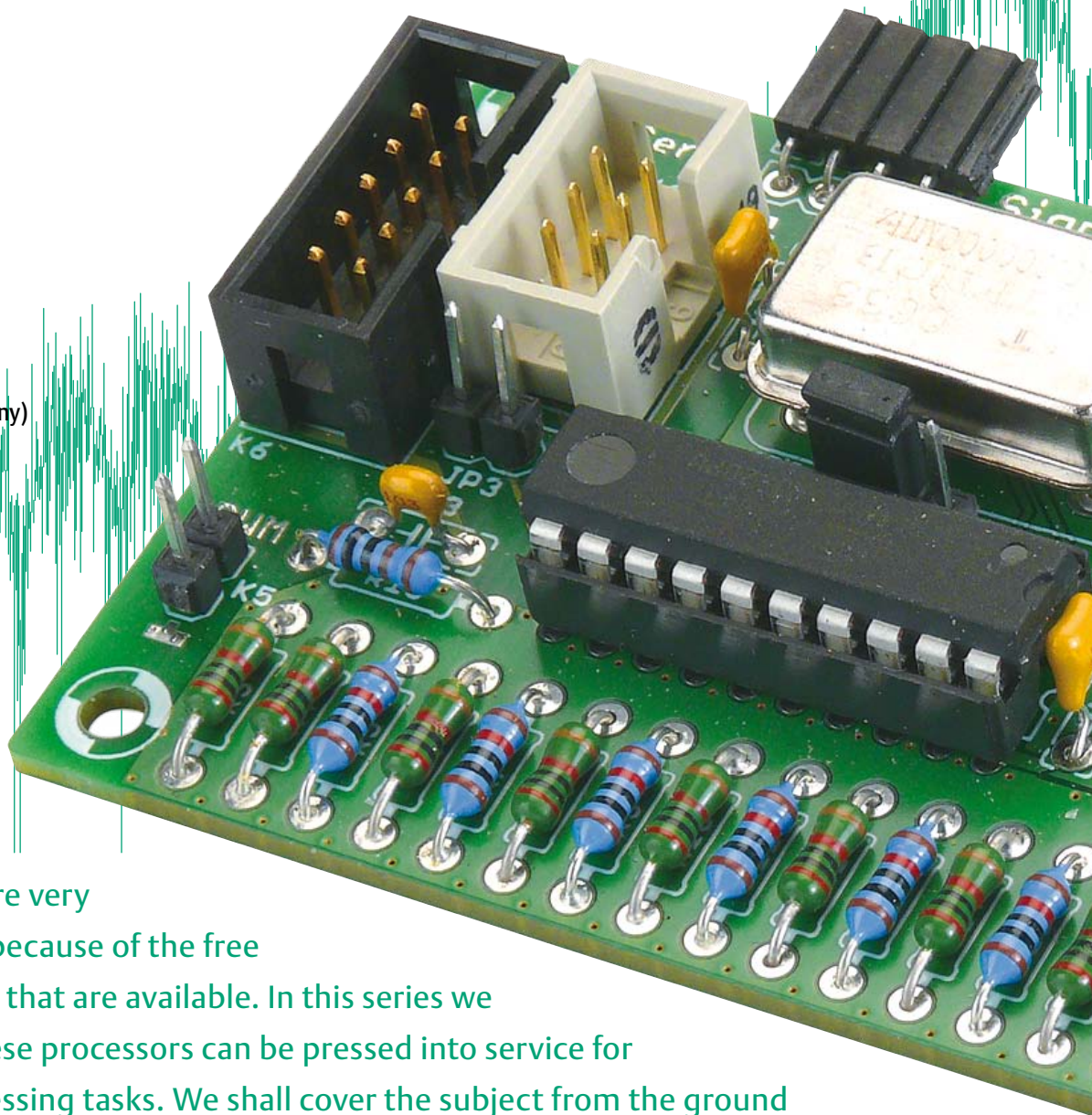


AVR Software Defined

Generating precision signals using

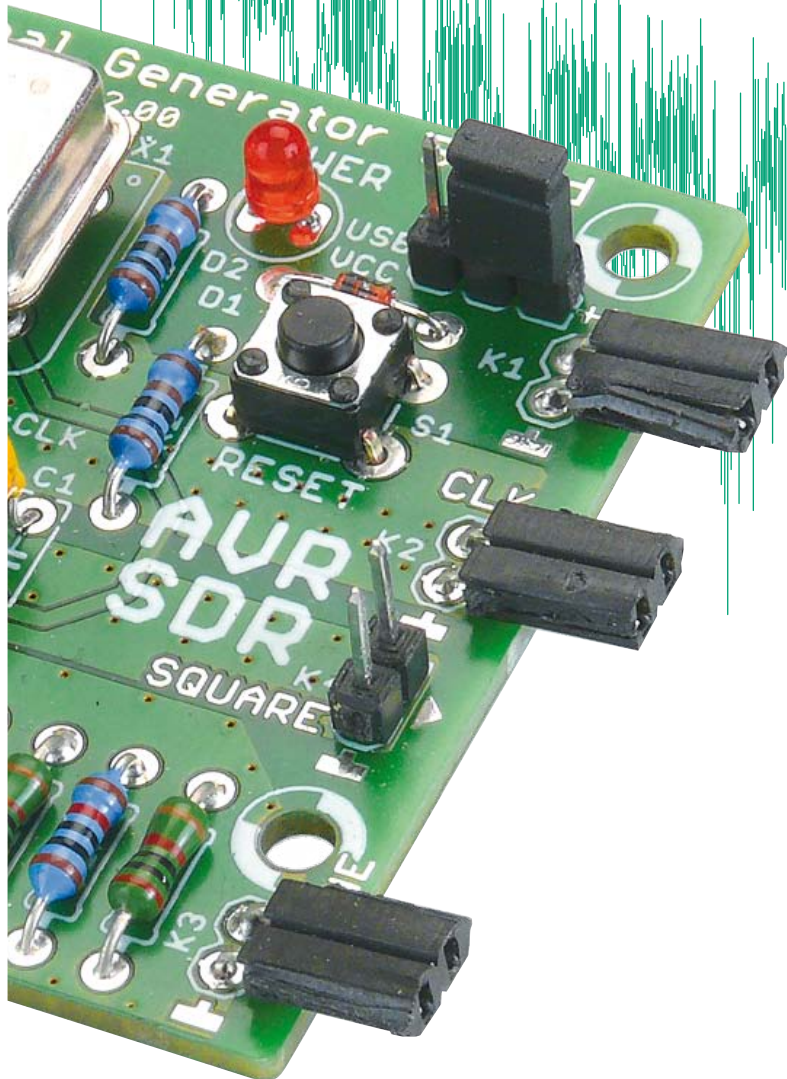
By Martin Ossmann (Germany)

Atmel AVR microcontrollers are very popular, not least because of the free development tools that are available. In this series we shall show how these processors can be pressed into service for digital signal processing tasks. We shall cover the subject from the ground up, making the series suitable for beginners, and in true Elektor style the focus will be on practical experiments. You can build the hardware yourself or you can obtain boards from Elektor, and the software is as ever available for download as source code from our website. Let's generate some signals!



Radio Part 1

Using an ATtiny micro



First a quick peek at what is in store in this series. The first board, which includes an ATtiny2313, a 20 MHz oscillator and an R-2R DAC, will be used to make a signal generator. The second board will fish signals out of the ether. It contains all the hardware needed to make a digital software-defined radio (SDR), with an RS-232 interface, an LCD panel, and a 20 MHz VCXO (voltage-controlled crystal oscillator), which can be locked to a reference signal. The third board provides an active ferrite antenna. The software for all these projects is written using the WinAVR GCC compiler in AVR Studio and can be downloaded as C source code (plus fuse settings) or as hex files.

The series is built around practical experiments. We can look forward to sine- and squarewave generators, an RMS voltmeter, experiments in FM, AM and PM, FIR and IIR filters, wireless data transmission, reception of the DCF timecode signal, RTTY weather messages, BBC long-wave radio transmissions and much more!

Before we get started, one word of warning: fluorescent energy-saving light bulbs are based on switching regulators which splatter interference all over the long-wave band. So we advise carrying out the more sensitive experiments with the fluorescents off and the mains halogen lights on (or by candlelight!).

Signal generator board

The signal generator board is based on an AVR microcontroller clocked at 20 MHz and an R-2R ladder forming a digital to analogue converter to produce the output voltages. This is hardly a novel circuit, but we will show how it can be used in a wide range of applications. In particular we will use it to generate outputs useful for testing other circuits, such as frequency- and phase-modulated signals. Then, for even greater precision, we will connect the signal generator to an external clock source which is in turn locked to a frequency

Elektor Products and Support

- Signal generator kit including printed circuit board and all components: # 100180-71
- BOB-FT232R USB-to-TTL converter, ready built and tested: # 110553-91
- USB AVR programmer, printed circuit board with SMDs fitted, plus all other components: # 080083-71
- Free software download (hex files and source code): file # 100180-11.zip

All products and downloads are available via the web pages for this article: www.elektor.com/100180

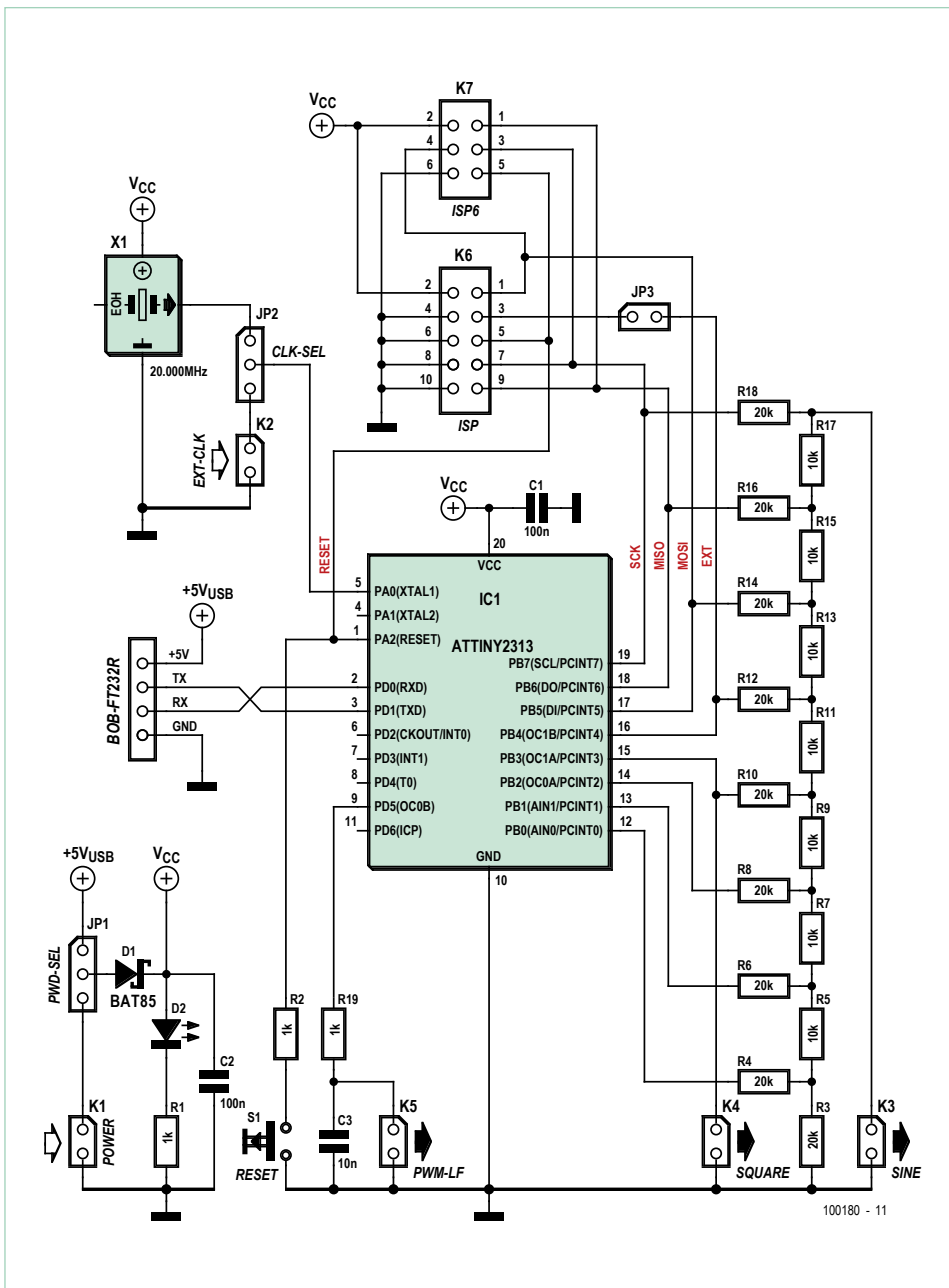


Figure 1. Circuit diagram of the signal generator.

standard such as the German DCF77 signal on 77.5 kHz or French TDF signal on 162 kHz.

The circuit of the signal generator is shown in **Figure 1**. The central component is the ATtiny2313 microcontroller, with the R-2R ladder connected to port B forming the digital-to-analogue converter. The analogue output signal appears on K3 (SINE). Note, however, that the output impedance of the circuit is relatively high at 10 kΩ. PWM output OC1A of the microcontroller is also available at K4 (SQUARE). We will use this output to generate square waves with frequencies of up to a few hundred kilohertz, as well as to modulate other signals. Another PWM output, OC0B, is brought out to K5 (PWM-LF) via a low pass filter comprising R19 and C3: this is suitable for generating low-frequency analogue signals.

The processor is clocked at 20 MHz by oscillator X1. It is a good idea to choose a relatively high-precision component here (50 ppm or better). Using a socket makes it easier to try out different types of oscillator or oscillators of different frequencies. Jumper JP2 allows the use of an external clock signal, which should be supplied at K2 (EXT-CLK).

The signal generator software programs allow a certain amount of external configuration using the microcontroller’s UART. The relevant pins are brought out to a connector on the board (which is available from *Elektor* in the form of a kit including all the components). The connector is suitable for directly attaching the BOB-FT232R USB-to-serial converter [1]. JP1 allows power to be obtained over the USB connection when the unit is used with a PC: in this case no additional AC power adaptor is needed. Populating the printed circuit board (**Figure 2**) should present no particular difficulties: all the components are ordinary leaded types. It is worth using a socket for the processor in addition to the clock oscillator. Be sure to observe correct polarity on the programming connectors K6 and K7. Programming can be done using the *Elektor AVRprog* [2]. It is of course important to get the fuse configuration right: the source code gives this along with the compiler options in each case.

DDS sinewave generator

Our first application is a simple sinewave generator programmed in C. The basic sample clock is produced by one of the timers built in to the microcontroller, arranged to trigger an interrupt. The interrupt routine is responsible for calculating the next sample value of the sinewave (**Figure 3**). Call the *k*th sample $S[k]$. Writing $p[k]$ for the phase of this sample, we have

$$S[k] = \sin(p[k]).$$

Between one sample and the next the phase advances by a constant value *d* (the ‘phase increment’):

$$p[k+1] = p[k] + d.$$

In a perfect sinewave generator these calculations must be carried out exactly, which of course is not practical. Instead, the direct digi-

COMPONENT LIST

Resistors

R1,R2,R19 = 1kΩ
 R5,R7,R9,R11,R13,R15,R17 = 10kΩ
 R3,R4,R6,R8,R10,R12,R14,R16,R18 = 20kΩ

Capacitors

C1,C2 = 100nF 100V
 C3 = 10nF

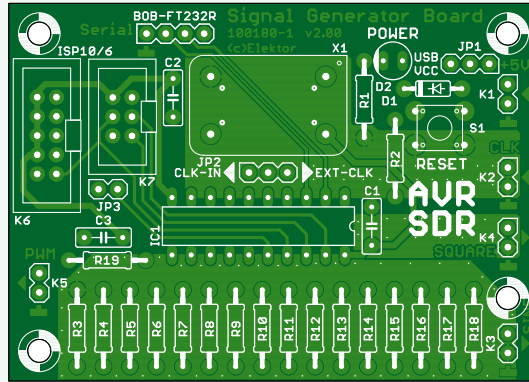
Semiconductors

D1 = BAT 85 (Schottky diode)
 D2 = LED, green
 IC1 = ATtiny2313-20PU, programmed

Miscellaneous

S1 = pushbutton
 K4,K5 = 2-pin pinheader, lead pitch 0.1” (2.54mm)
 JP3 = 2-pin pinheader, lead pitch 0.1” (2.54mm) with jumper

JP1,JP2 = 3-pin pinheader, lead pitch 0.1” (2.54mm) with jumper



K1,K2,K3 = 2-way receptacle, right-angled
 BOB = 4-way receptacle, right-angled
 K6 = 10-way ISP boxheader
 K7 = 6-way ISP boxheader
 X1 = 20MHz quartz crystal (with 4 receptacles Harwin type H3153F01)
 BOB-FT232R-001 = Elektor ‘BOB’ USB/TTL converter (ready assembled and tested, # 110553-91)
 Printed circuit board

Alternatively
 Kit, including board and all parts: # 100180-71.

Figure 2. The printed circuit board is available from Elektor as part of a kit including all the components.

tal synthesiser (DDS) stores the current phase value DDS_p to finite precision as an m -bit number in the so-called ‘phase accumulator’. One complete period of the sinewave corresponds to this value covering the range of values from 0 to $2^m - 1$. The same precision is used for storage of the phase increment and for the phase addition operation.

The next step is to convert the phase value into the corresponding sinewave sample. This is done using a look-up table which stores a complete sinewave period. If we were to store a sample for each of the 2^m possible values in the phase accumulator the table would be unmanageably big: instead we use just the top n (where $n < m$) bits of the phase accumulator to index the table, which now need only contain 2^n samples. The values stored in the table are rounded from their exact values to r -bit samples $S[k]$, where r is the number of bits in the digital-to-analogue converter. **Figure 4** illustrates the process. In our first program we use $m=32$ and $n=8$. A 32-bit phase accumulator gives us very precise frequency control over our signals. We use a sine table with $256=2^8$ entries and an 8-bit DAC ($r=8$). The program EXP-SinusGeneratorDDS-T1INT-V01.c [3] is configured to

produce a fixed frequency output at 1 kHz; the result can be verified on an oscilloscope (**Figure 5**). The interrupt service routine code is shown in **Listing 1**.

Timing

The DDS is clocked at $f_{DDSCLK} = 100$ kHz. To generate a desired output frequency f the required phase increment is calculated using

$$DDSd = 2^n \times f / f_{DDSCLK}$$

and so for $f = 1$ kHz we have

$$DDSd = 2^{32} \times 1 \text{ kHz} / 100 \text{ kHz} = 42949673$$

and you will find precisely this value in the C source code where $DDSd$ is initialised.

We can see from this formula that the higher the sample rate the higher the frequencies we can generate. Why did we choose 100 kHz? The answer lies in the timing of the interrupt service rou-

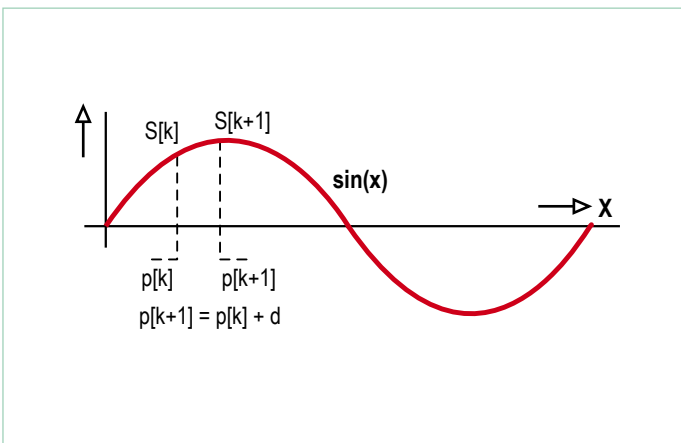


Figure 3. Sampling a sinewave.

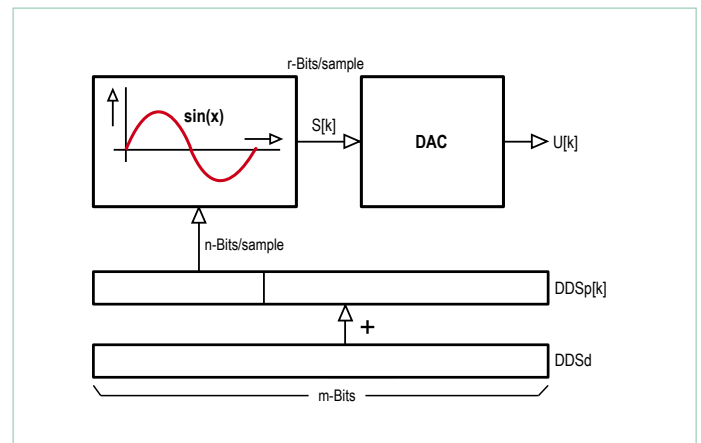


Figure 4. How the DDS sinewave generator works.



Figure 5. Testing the sinewave generator.



Figure 6. Measuring the execution time of the interrupt service routine.

time. As you can see from the code snippet above, we have bracketed the calculation with commands to set and clear port pin PD.4. This allows us to observe the execution time of the calculation using an oscilloscope: in this case we see a total time of around 2.2 μ s. However, we must be careful as this does not include other contributions to the total time needed to service the interrupt: for example, the time to save and restore processor registers will not be counted. However, with a relatively simple experiment we can determine these times as well.

Simply set up the main program as an infinite loop in which a port pin (say PD.5) is toggled as quickly as possible. We can then observe on the oscilloscope the periods when the toggling stops, which is when the interrupt service routine is active: see **Figure 6**.

In our experiment we measured the total time needed to process an interrupt at about 5.4 μ s. The maximum allowable interrupt rate is therefore 180 kHz. Adding a safety margin, we arrive at our figure of 100 kHz.

As the output frequency f approaches f_{DDSCCLK} we start to observe undesirable artefacts such as jitter, noise and aliases in the output spectrum. With a sample rate of 100 kHz it is best to keep f below about 10 kHz. Perhaps we can do a bit better if we use assembly code?

A faster DDS sinewave generator

In order to make our sinewave generator capable of higher frequencies we need to rewrite the DDS routine in assembler. With the help of a cunning arrangement of variables in registers we can manage to get the sample rate of the 32-bit DDS as high as 2 MHz. The code (**Listing 2**) uses the T flag to allow it to break out of its loop.

Our project now consists of a mixture of C and assembler code, and we need to store the sinewave table at a fixed address in memory. Configuring the project within WinAVR to achieve this is not a task for the beginner. If you do not plan to make any changes to the code it is probably best to program the ready-compiled hex file into the

Listing 1

```
ISR(TIMER1_OVF_vect) {
    PORTD |= _BV(4); // set sample timing flag
    PORTB=pgm_read_byte( SIN8+(DDSp>>24)); // fetch and output sine sample
    DDSp += DDSd; // advance DDS phase DDSp by DDSd
    PORTD &= ~ _BV(4); // clear sample timing flag
}
```

Listing 2

```
loop:
    add DDSphase0,DDSdelta0 // 1 LSB of 32 bit DDS adder
    adc DDSphase1,DDSdelta1 // 1
    adc DDSphase2,DDSdelta2 // 1
    adc ZL, DDSdelta3 // 1 MSB is in ZL as pointer
    lpm R0,Z // 3 access sine table
    out PORTB,R0 // 1 out to R-2R DAC on PORTB
    brtc loop // 2 (1) loop until T flag set by interrupt routine
// 10 cycles in total for one loop
```

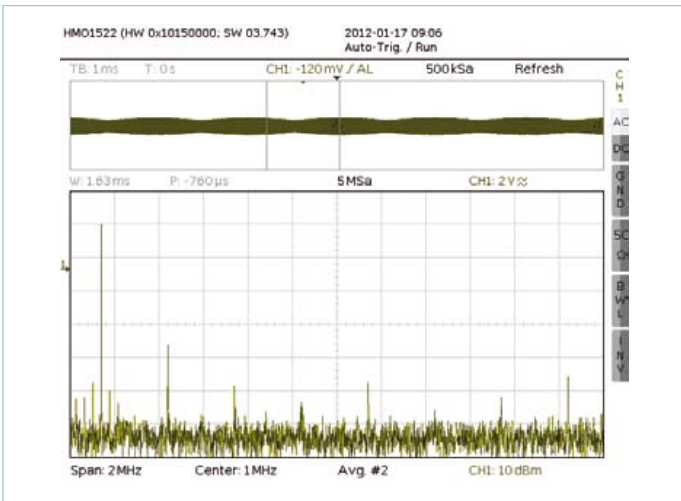


Figure 7. Spectrum of the generated signal.

frequency followed by CR and LF. The maximum usable signal frequency is about 200 kHz. The theoretical frequency resolution is given by

$$f_{\text{DSDCLK}} / 2^n = 2 \text{ MHz} / 2^{32} = 0.00046... \text{ Hz.}$$

To take advantage of this resolution the software allows you to enter a frequency with up to three digits after the decimal point, for example as '1000.045' (followed by CR and LF). The internal calculations required to turn the entered frequency into a suitable parameter value for the DDS need to be carried out very accurately. To this end the author has written special-purpose arithmetic routines, including one for fixed-point division.

processor (paying attention to the fuse bit settings). The project is called EXP-SinusGenerator-DDS-ASM-C-V01. To make the sinewave generator more flexible it includes the ability to be configured over the UART interface (19200 baud, 8N1 data format). Using a terminal program, simply type in the desired

Figure 7 shows the spectrum of the sinewave output signal at frequency $f = 125.123 \text{ kHz}$ over the range from 0 Hz to 2 MHz. As you can see, there are harmonics present, but all at more than 30 dB below the desired signal. A low level of wideband noise is also visible: this is a by-product of the DDS technique.

Advertisement

Listing 3

```

uint32_t DDS24 ; // DDS phase, 24 bits used
volatile uint32_t dDDS24 ; // delta for DDS
uint16_t TOP1 ; // integer part of divider for PWM

ISR(TIMER1_OVF_vect) {
    PORTD |= _BV(4) ;
    DDS24 += dDDS24 ; // advance DDS phase
    if (DDS24 & 0x1000000UL) { // check bit 24 for overflow
        ICR1 =TOP1 ; // on overflow PWM width = TOP1+1
    }
    else {
        ICR1 =TOP1-1 ; // else PWM width = TOP1
    } ;
    DDS24 &= 0xffffffffUL ; // make DDS24 24 bits again
    PORTD &= ~ _BV(4) ;
}

```

If an ordinary crystal oscillator is used the overall accuracy of the system will be in the region of plus or minus 100 ppm. In that context it hardly makes sense to claim that the generator can produce an output frequency of say 100.00005 kHz. For this reason the generator can accept an external 20 MHz clock signal, and in a later instalment of this series we will see how such a clock can be derived from a transmitted reference. This combination allows the generation of sinewaves with excellent frequency accuracy.

Trimming resonant circuits

Later in this series we will use an AVR microcontroller to receive and process longwave transmissions such as the DCF time code on 77.5 kHz, France Inter on 162 kHz and BBC Radio 4 on 198 kHz. Usually a ferrite antenna will be used, and adjusting such an antenna can be made much simpler using our sinewave generator: connect the circuit up as shown in **Figure 8** and adjust the trimmer capacitor for maximum amplitude.

It is possible to use the phase relationship between the output voltage U_{OUT} and the input voltage U_{IN} to determine whether the resonant frequency of the circuit is higher or lower than that of the input sinewave. If the phase of U_{OUT} leads U_{IN} then the sinewave frequency is lower than the resonant frequency; if U_{OUT} lags U_{IN} the signal frequency is higher than the resonant frequency. When the frequencies agree U_{IN} and U_{OUT} are in phase.

The example circuit shows component values for a resonant frequency of 125 kHz; coil L1 is a small pot core inductor. We will use this circuit later to generate test signals at a frequency of 125 kHz. The trimmer capacitor can be adjusted to bring the resonant frequency of the circuit to exactly 125 kHz: it is possible to use either the signal from the R-2R ladder (K3) or the squarewave from the PWM output (K4) to do this.

PWM squarewave with a fractional divider

We will now look at another application of the DDS principle. If we use a timer with a PWM output to generate a squarewave, we are normally limited to producing frequencies that exactly divide the frequency at which the timer is clocked. If N is the division ratio and f_{CLK} the timer's clock frequency then the output frequency will be $f = f_{CLK}/N$. However, if we adjust the division ratio on the fly (say between N and $N+1$) then we can also produce intermediate fre-

quencies. So for example if we alternate between using a divisor of N and a divisor of $N+1$ then the overall effective division ratio will be $N+0.5$. If a division ratio of 10.333... is wanted, then this can be achieved by using a division ratio of 11 with 'probability' 0.333... and a division ratio of 10 the rest of the time.

How can we use this in practice? We need an algorithm that will tell us, given the desired division ratio, when to divide by N and when by $N+1$.

Again, the m -bit DDS synthesiser comes to the rescue, with a sufficiently large value of m to achieve the desired precision. In this case we make use of the overflow of the phase accumulator. The rate p at which an m -bit phase accumulator overflows is just

$$p = \text{DDSd} / 2^m$$

and so we can control this rate as precisely as we need using the variable DDSd. Whenever the accumulator overflows the timer is instructed to divide by $N+1$ rather than N on the next cycle.

So, for example, if we wish to generate a 77.5 kHz squarewave from a 20 MHz master clock, the required overall division ratio is

$$20000 / 77.5 = 258.0645161\dots$$

and so we need to divide by either $N=258$ or $N=259$ on each cycle. The 'probability' of selecting $N=259$ will be $p = 0.0645161\dots$, which with a 24-bit DDS means that $\text{DDSd} = p \times 2^{24} = 1082401$. **Listing 3** shows an interrupt service routine embodying this idea.

The squarewave produced by this code does of course suffer from a small amount of short-term jitter, but in the long term the agreement with the ideal frequency is excellent.

The whole routine, including interrupt overheads, has an execution time of about 6 μs , which means that we can use the technique to generate frequencies of up to about 160 kHz. Rewriting the routine in assembler should allow considerably higher frequencies.

For convenience the squarewave generator program can also be controlled using a terminal. The source code is called EXP-SquareGenerator-DDS-T1INT-V01.c in the downloadable zip archive [3]. The fractional divider has many other applications. For example, it can be used to generate a signal with any desired sample rate, or form part of a digital PLL.

FM generator

On its own the squarewave generator is perhaps not particularly exciting. However, since controlling the PWM output does not occupy all the processor's time, we have some computing power left over to change the frequency dynamically to make an FM generator.

The German meteorological service [4] transmits weather information on 147.3 kHz using frequency shift keying (FSK) in radioteletype (RTTY) format. Later in this series we will build a receiver capable of decoding these messages. To adjust and test the receiver it is helpful to have a test signal. Using a fractional divider and the PWM output this is easy: we just use a stream of data bits to control the output frequency.

We will first program our test signal generator to work with a carrier frequency of $f = 125$ kHz. A circuit along the lines of Figure 8 is used to turn the squarewave output into a sinewave. We have already seen the interrupt service routine for the fractional divider; the new routine, 'SendBit', is responsible for sending a single bit (Listing 4).

First we wait for Timer 0 to run through COUNT2 cycles: in other words, the bit rate is the Timer 0 overflow rate divided by COUNT2. Then, depending on the value of the bit to be sent, the values of deltaDDS24 and TOP1 are set. This is where the frequency modulation occurs. Note that the commands that set these values are enclosed between a cli() and sli() pair. If this is not done it is possible that the interrupt service routine could be invoked when one parameter has been changed but not the other, with potentially unpredictable results. The routines can be found in the program EXP-SQTX-FM-RTTY-V01.c. With further auxiliary routines we can transmit characters using the Baudot [5] code, emulating the meteorological service transmissions.

Figure 9 shows the spectrum of the FM RTTY signal. There are two narrow peaks adjacent to one another, at frequencies of $125 \text{ kHz} \pm 50 \text{ Hz}$. The spectrum is continuous, falling off rapidly beyond $\pm 1 \text{ kHz}$.

Having built such a transmitter, it is natural to want to test it to check that the modulation is correct. For that reason the next instalment in this series will start to look at the components that comprise a digital receiver.

(100180)

Internet Links

- [1] www.elektor.com/110553
- [2] www.elektor.com/080083
- [3] www.elektor.com/100180
- [4] www.hfunderground.com/wiki/RTTY_maritime_weather_transmissions
- [5] http://en.wikipedia.org/wiki/Baudot_code

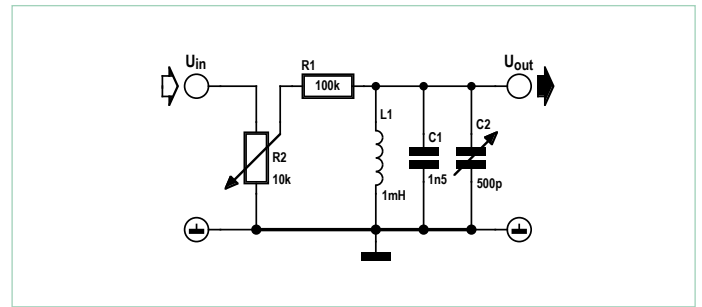


Figure 8. Trimming a resonant circuit.

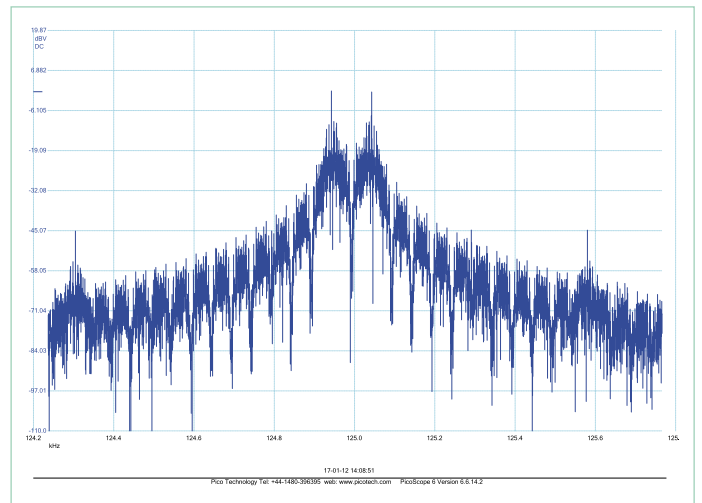


Figure 9. Spectrum of a frequency-modulated signal at $125 \text{ kHz} \pm 50 \text{ Hz}$.

Listing 4

```
void SendBit(uint8_t theBit) {
    uint8_t k ;
    for (k=0 ; k<COUNT2 ; k++){
        while( ( TIFR & (1 << TOV0) ) == 0 ) { }
        TIFR |= (1 << TOV0) ;
    }
    if ( theBit==MARK ) {
        cli() ;
        deltaDDS24=MARK_deltaDDS24 ;
        TOP1=MARK_TOP1 ;
        sei() ;
    }
    else{
        cli() ;
        deltaDDS24=SPACE_deltaDDS24 ;
        TOP1=SPACE_TOP1 ;
        sei() ;
    }
}
```

Android Switch Interface

Using an Android smartphone as a remote control or user interface for your microcontroller projects



A low-cost Android phone forms an excellent basis for a high-end user interface or remote control for a microcontroller circuit. In this article we tell you how to implement various wireless sensing and switching functions using an Arduino board with a Bluetooth shield. We also describe how you can program your own Android app for this purpose and what (free) PC software you need for this.

By Jos van Kempen (The Netherlands)

Nearly all microcontroller circuits intended to control something have some form of user interface. This often involves several buttons, knobs, LEDs and an LCD module. If you want something really nice, you might even use a touchscreen. A remote control can also be handy – preferably one that does not have to be aimed so precisely. However, ‘luxury’ interfaces of this sort usually cost more and are more difficult to implement. This doesn’t have to be the case. As you will see, an attractive user interface with remote control does not need to be expensive, and the programming is reasonably easy. There’s a good chance that your have a smartphone with Bluetooth

capability, which can be used as a fancy remote control unit. Some microcontroller boards have built-in Bluetooth capability, and with others you can buy adapters or shields to add this capability. In this article we use a small Arduino board with a Bluetooth shield (Itead, priced at around 15 pounds) augmented by a small I/O shield specifically designed for this article.

Articles about adding Bluetooth functionality to your own circuits have been published in previous issues of Elektor (September 2004 and February 2010). Elektor also published an article about programming Android apps in the June 2011 edition, and an Android controller using the audio output was described in December 2011.

Many of you are familiar with microcontroller programming, although you may not have any experience with programming a

mobile phone, so we devote only minimum attention to microcontroller programming in this article. However, we discuss smartphone programming in detail, from downloading and installing the software to programming the various components of the desired user interface. We hope this information will be sufficient to enable you to program a user interface yourself for your own application with your own microcontroller. The source code of the application described here (for both the microcontroller and the smartphone) can of course be downloaded from the Elektor website [1].

Hardware and microcontroller software

A Bluetooth shield is fitted on the Arduino board. It communicates over the UART interface. In Bascom you can use the 'input' and 'print' instructions to receive commands and send data from and to the Arduino board.

For this project we also developed a simple shield consisting of a pair of relays with LEDs (digital outputs), a PWM output with a FET ('analogue' output), a switch (digital input) and an NTC thermistor as an analogue sensor (analogue input). The schematic diagram is shown in **Figure 1**, and the PCB layout designed for this circuit is shown in **Figure 2**. This shield can be used to experiment with all of the options for communication between the Android smartphone and the Arduino board using Bluetooth. All relay, switch and sensor lines are available on connector K9.

The microcontroller software (written in Bascom) works as follows (see **Listing 1**).

A loop checks whether a character has been received. If the Arduino receives an 'R', output D11 is activated (LED1 lights up and relay RE1 is energised). If it receives an 'r', LED1 and RE1 are de-energised. Output D13 and the second LED and relay behave in a similar manner, but they respond to the characters 'O' and 'o'. If

a 'P' (short for 'PWM') is received, the routine 'Input' waits for a value (the smartphone must send the character string '\r\n' after the value), and this value is used to drive the PWM output. FET T3

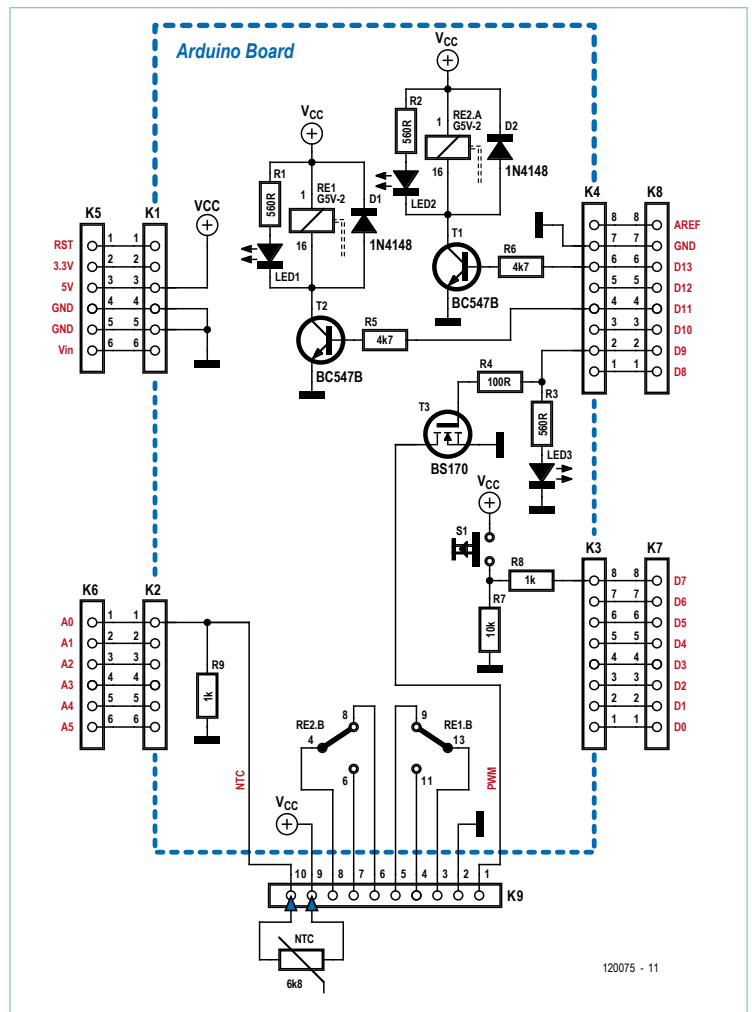


Figure 1. Schematic diagram of the I/O shield consisting of a pair of LEDs and relays, an NTC thermistor, a pushbutton, and a PWM output with a FET and an indicator LED.

COMPONENT LIST

Resistors

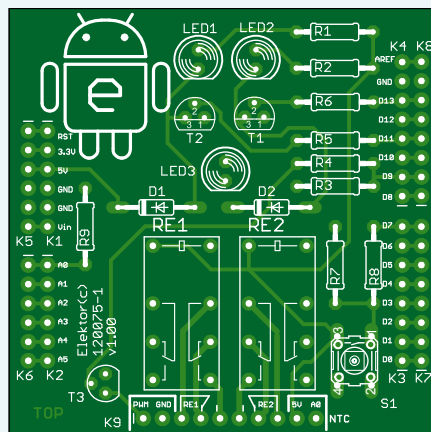
- R1,R2,R3 = 560Ω
- R4 = 100Ω
- R5,R6 = 4.7kΩ
- R7 = 10kΩ
- R8,R9 = 1kΩ

Semiconductors

- D1,D2 = 1N4148
- LED1,LED2,LED3 = LED, red, 5 mm
- T1,T2 = BC547
- T3 = BS170

Miscellaneous

- K1,K2 = 6-pin pinheader
- K3,K4 = 8-pin pinheader
- K5,K6 = 6-way pinheader socket



- K7,K8 = 8-way pinheader socket
- K9 = 10-way pinheader socket
- RE1,RE2 = 5V miniature relay (e.g. TE Connectivity type MT2-C93401 or OMRON type G5V-2-H1)
- S1 = pushbutton with make contact (e.g. B3F-1000)
- PCB # 120075-1 (see [1])

Figure 2. The PCB layout is designed such that the connectors mate with a standard Arduino board.

Listing 1. The microcontroller program (written in Bascom).

```

$baud = 9600 : Ucsr0a = &H00                                     'Thanks to J.F. Theinert
'Software under CC-BY-NC-SA licence by Jos van Kempen.
Config Adc = Single , Prescaler = Auto , Reference = Avcc      'ADC (analog) input initialize
Start Adc                                                     'PWM (analog) output initialize
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Down , Prescale = 8
Config Pinb.1 = Output                                       'PB1 =digpin9=pwm1a
Pwm1a = 0
Dim Pwm_str As String * 5                                     'text 0-255
Dim Pwm_b As Byte
Dim Value As Integer
D13 Alias Portb.5 : Config D13 = Output                       'Dig13 no resistance needed for LED
D11 Alias Portb.3 : Config D11 = Output                       'Dig11
D7 Alias Pind.7 : Config D7 = Input                           'Initialize DigInput Dig7
Declare Sub Set_pwm
Dim B As Byte
Do
  B = Ischarwaiting()
  Print B
  If B = 1 Then                                               'if incoming command
    B = Waitkey()
    Select Case B
      Case "R" : D11 = 1
      Case "r" : D11 = 0
      Case "O" : D13 = 1
      Case "o" : D13 = 0
      Case "P" : Call Set_pwm
    End Select
  End If
  Waitms 300
  Pwm_b = Pwm_b + 3
  Value = Getadc(0)                                           'A0
  Print "T" ; Value ; "t"
  Waitms 40
  If D7 = 1 Then Print "G" Else Print "g"
  Waitms 30
Loop
End
Sub Set_pwm
  Input Pwm_str Noecho
  Pwm_b = Val(pwm_str) : Waitms 30
  Pwm1a = Pwm_b
  Print "*" ; Pwm_b : Waitms 30
End Sub

```

has a maximum rating of 60 V / 0.5 A. LED3 provides an indication of the PWM level.

The measured value of the NTC thermistor sensor R9 is also printed with the format 'T;adc(0);t', and the signal level at input D7 to which pushbutton S1 is connected is printed as 'G' if it is high ('1') or 'g' if it is low ('0'). The interface is configured with standard data rate of 9600 baud, which means that a small waiting time (around 30 to 40 ms) is necessary after a value has been sent.

Required software

The following software that you need in order to program apps for Android smartphones can be downloaded for free:

1. The Java language is used for programming. The Java Development Kit (JDK) can be downloaded from the Oracle website [2].
2. The Android Software Development Kit (SDK) is available at [3]. After installing the SDK, you should have a look at the rest of the site, where you can find a lot of information about programming, help files and USB drivers for your smartphone (which you will need later for downloading your app). The USB driver may already be installed on your PC if you use USB to exchange photos, music and so on between your PC and your smartphone.
3. Download the ADT plugin for Eclipse from the same site, and make note of the folder it goes into.
4. We used Eclipse Classic 3.6.2 as the integrated development environment (IDE). It can be downloaded from [4]. (Note: the latest Android plugin at the time of writing was ADT12, which is not compatible with Eclipse 3.7.)
5. After installing Eclipse, you have to install the Android plugin (under *Help | Install new software | Archive*). Select the ADT zip

file and enter the name 'Android'. Then go to *Windows | Preferences* and look for the folder with the extracted Android SDK (in lower-level folders, including 'Tools').

Go to *Windows | Android SDK and AVD Manager*, select *New (Virtual Device)*, and select Gingerbread for version 2.3.3 (or Samsung_GIO or something similar) so that you can simulate the program on the PC without a smartphone if necessary.

- Download the folder containing the BluetoothInterface example project from [1], but do not put it in the Eclipse workspace.

The BluetoothInterface project

The software that allows a Bluetooth device to find other Bluetooth devices, establish contact and exchange messages is complicated, but fortunately an example is included with the Android SDK. Unfortunately, it apparently does not work properly with many Bluetooth devices. A replacement for one of the files (*BluetoothRfcommClient.java*) can be found on the Internet. After the declarations are modified, it solves this problem (see [5] and [6]). Now you can use the project to make your own interface.

After starting Eclipse, you can select *File | New | Android project | to create a new project* (perhaps based on an example), but in this case you already have a project that you want to import into the Eclipse workspace. To do this, select *File | Import.. | Existing Projects in Workspace*. Be sure to tick *Copy projects into workspace*, as otherwise the original project will be overwritten. Browse to the folder where you placed the downloaded BluetoothInterface project.

Various .java and .xml files are visible in the project file structure (Figure 3). You may have to click the relevant folder to see them. The ones we are interested in are *BluetoothChat.java* (main program), *main.xml* (smartphone screen interface) and *strings.xml* (declaration of the variables used in the interface).

Interface implementation

Double-click *main.xml* to display the smartphone screen interface (see Figure 4). The tabs below the screen layout can be used to switch between the graphical layout and the automatically generated code.

A horizontal layout is already defined on the interface, including a *ListView* for the detected Bluetooth devices, an *EditText* for the entering the text you want to send during a chat session, and a button for sending the text.

In many cases the easiest way to place a component is to drag it to the *Outline* pane. An error may occur when you drag the size on the screen if your PC is configured for a comma decimal marker. In this case you must replace the ',' in the file *main.xml* with a '.'.

When you place a button or a radio button, you can define the procedure that will be executed if it is clicked by entering the name of the procedure for the *OnClick* property.

The LEDs and relays (digital outputs) on the shield are operated by a set of buttons or checkboxes. For this purpose, place a linear layout of checkboxes or (toggle) buttons on the screen or in the outline (which is often easier).

The switch state (digital input) is indicated on the interface (for a

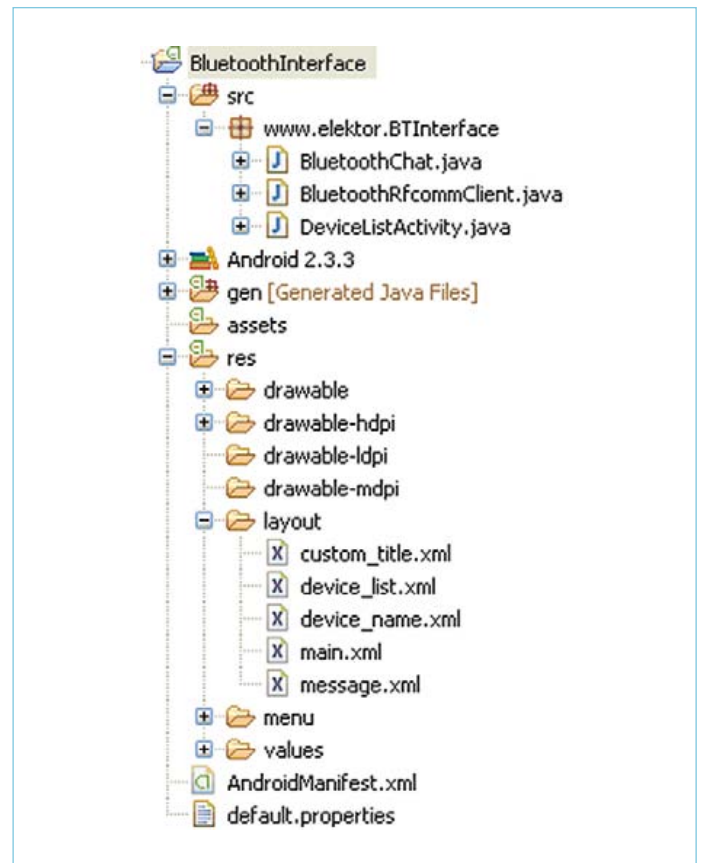


Figure 3. The file structure of the BluetoothInterface project.

change) by two radio buttons labelled 'On' and 'Off' (*radioGroup orientation horizontal*).

The signal level on the analogue input is indicated by a horizontal *progressBar* (style: *horizontal*), a number in a *textView* and a chart in an *imageView*.

The PWM output with the FET is set by a slider on a *seekBar*.

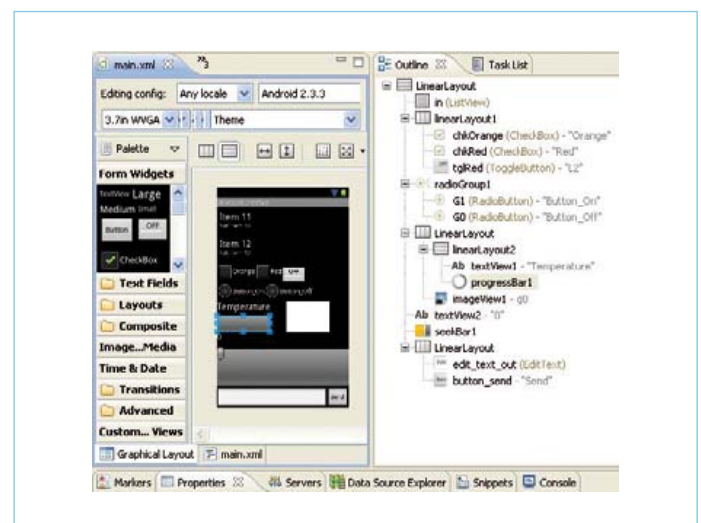


Figure 4. The interface screen layout. It's often easiest to place the controls in the Outline pane.

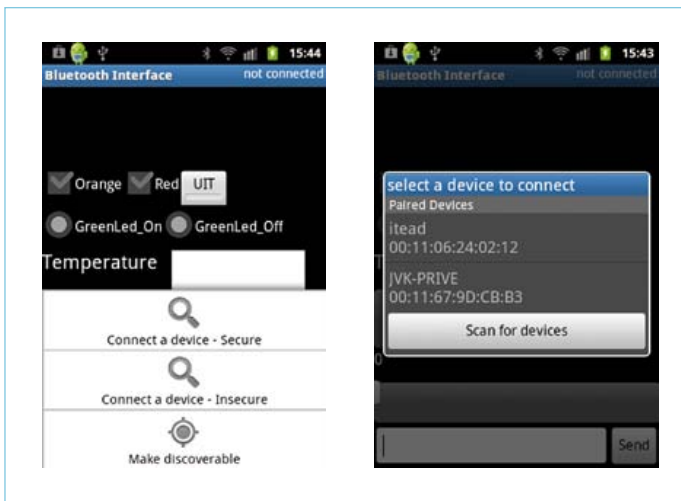


Figure 5. Select ‘Connect a device – Secure’ and then select one of the already paired devices or scan for devices that are configured as ‘discoverable’. You have to enter a code for (initial) pairing; ‘0000’ and ‘1234’ are common codes.

You haven’t programmed anything yet, but you can already see what the interface will look like when it’s done. You can also run a simulation on the PC with a virtual smartphone displayed on the monitor, but it isn’t possible to simulate a Bluetooth device with this. Connect your smartphone to the PC with a USB cable and then select the associated project folder. Right-click the folder and select *Run As.. | Android Application*. The app will be compiled and downloaded to your phone. If Bluetooth reception is enabled, you can now receive data and display it on your phone.

If you wish, you can also test the app by communicating with a Bluetooth USB stick plugged into the PC, in combination with a terminal emulator program for sending and receiving data. Of course,

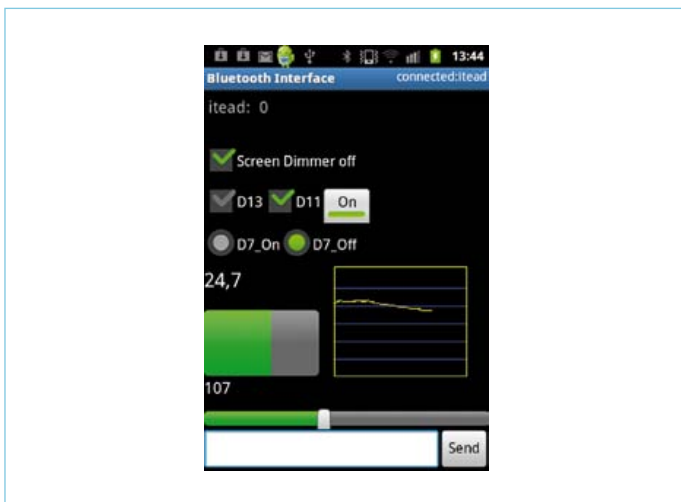


Figure 6. The interface in use. The messages sent by lthead (the Bluetooth shield) or Me (the phone) are visible at the top. Commands to be sent can be entered manually at the bottom. These features are handy for troubleshooting, but you may want to remove them for other applications.

you must first select the right COM port in the terminal emulator (e.g. Hyperterminal or Advanced Serial Port Terminal) and establish a connection.

After the app has been downloaded and tested, establish a connection by clicking the menu button at the lower left and choosing the desired Bluetooth device (select *connect to device –secure*). If everything goes right, you should see the message ‘connecting’ followed a short while later by ‘connected: <device name>’ (Figure 5). The code in the BluetoothChat file may look like this:

```
private CheckBox chkD13, chkD11,chkDIM;
    (The variable must be declared if the checkbox
    properties are read (or written) in the program)

chkD13 = (CheckBox) findViewById(R.id.chkD13);
    (The link between the name in main.xml (the screen
    layout) and the name in the program is created in
    the procedure 'onCreate')

public void chkD13Click(View view){
    if (chkD13.isChecked()==true)
        sendMessage("O"); else sendMessage("o");
}
```

The procedure itself is very simple. Either an ‘O’ or an ‘o’ is sent, depending on whether the checkbox is ticked or cleared. These are the commands that cause the microcontroller to activate or deactivate LED 2 and relay 2. The code for sending a value determined by the position of the slider on the seek bar when it is released is a bit more complicated. However, if you start typing the ‘quick fix’ assistant will point out your mistakes and generate most of the code automatically. It also automatically adds *implements OnSeekBarChangeListener* in the class declaration.

```
public void onProgressChanged(SeekBar seekBar, int
progress, boolean fromUser) {
    // TODO Auto-generated method stub
    textView2.setText(Integer.toString(progress));
}

public void onStartTrackingTouch(SeekBar seekBar) {
    // TODO Auto-generated method stub
    textView2.setTextColor(Color.rgb(255, 48, 48));
}

public void onStopTrackingTouch(SeekBar seekBar) {
    // TODO Auto-generated method stub
    textView2.setTextColor(originalTextColor);
    sendMessage("P");
    try{
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```



```

    }
    sendMessage (seekBar1.getProgress()+
"\r\n");
}

```

The only things you have to add here are code to change the displayed text value when the slider in the seek bar moves, code to change the colour of the text when you are moving the slider, and code to cause the phone to send a 'P' (which is the signal for the microcontroller to call a procedure in which the desired PWM value is read using *Input Pwm_str*) followed by the character string '\r\n' to mark the end of the message.

There is also already a procedure that receives messages from the microcontroller and places them in the text box. The following code is important for this:

```

case MESSAGE_READ:
    byte[] readBuf = (byte[]) msg.obj;
    // construct a string from the valid bytes
    in the buffer
    String readMessage = new String(readBuf, 0,
        msg.arg1);
    mConversationArrayAdapter.
add(mConnectedDeviceName+
    ": " + readMessage);
break;

```

Now all you have to do is to 'capture' the message and assign the right values to the radio buttons or the progress bar according to the message content.

The radio buttons are controlled by the characters 'G' (on) and 'g' (off):

```

    if (readMessage.contains("G")==true)
G1.setChecked(true);
    if (readMessage.contains("g")==true)
G0.setChecked(true);

```

For the progress bar, the code first checks whether the string 'Txxxxt' has been received. Here 'xxxx' is a number in the range of 0 to 1023, which is the digitised value of the signal level on the analogue input. The entire string is evaluated because it sometimes arrives in two pieces, which should not lead to an incorrect value:

```

if (readMessage.charAt(0)=='T')
    if (readMessage.contains("t")==true) {
        Tempoud=Temp;Temp=readMessage.indexOf("t");
        OldMessage=readMessage.substring(1,Temp);
        Temp=Integer.parseInt(OldMessage);
    }

```

The following code handles the value shown as text (the current temperature); for the progress bar and the chart it is sometimes necessary to recalculate and format the values due to the scale:

```

RTemp=Temp;RTemp=RTemp/6;// 0-255 naar grdC
// nooit hoger dan maximum progressbar
A0.setProgress(Temp*4); //schaal progressbar
Temp=Temp/2; //schaal grafiek
DecimalFormat formatter = new DecimalFormat("#.##");
textView1.setText(formatter.format(RTemp));

```

The measured value is also plotted on the chart. The chart is erased when the number of measured values reaches 150, after which the outline with scale lines is drawn again:

```

xcoordoud=xcoord;xcoord+=1;
if (xcoord==1) { //tekenen kader en hulplijnen
    paint.setColor(Color.BLUE);
    canvas.drawLine(0,105,150,105,paint); //10 grd
    canvas.drawLine(0,85,150,85,paint); //15 grd
    canvas.drawLine(0,65,150,65,paint); //20 grd
    canvas.drawLine(0,45,150,45,paint); //25 grd
    canvas.drawLine(0,25,150,25,paint); //30 grd
    paint.setColor(Color.YELLOW);
    canvas.drawRect(1, 1, 149, 124, paint);
}
if (xcoord==150) {xcoord=0;
    canvas.drawColor(Color.BLACK);}
else
{canvas.drawLine(xcoordoud,
    125-Tempoud,xcoord,125-Temp,paint);
}
OldMessage=readMessage;

```

To prevent the application from starting up with a keyboard, *Androidmanifest | Application | Window soft input mode* is set to *stateHidden*.

This completes the discussion of the main points.

Of course, you will want to tailor the application to your specific needs. We hope that you now have enough information to be able to do this yourself. The download on the Elektor website includes a complete application file (.apk) that you can copy directly to your smartphone and try out all of the features described here after installing it.

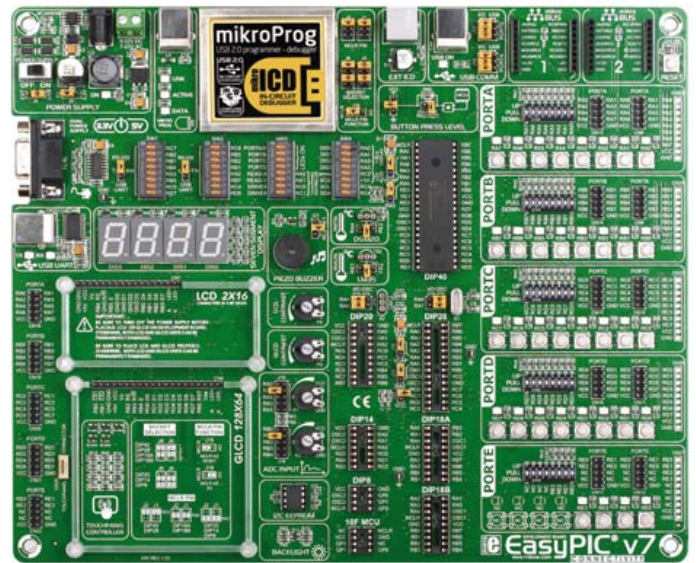
(120075-1)

Internet Links

- [1] www.elektor.com/120075
- [2] <http://www.oracle.com/technetwork/java/javase/downloads/>
- [3] <http://developer.android.com/sdk/index.html>
- [4] <http://www.eclipse.org/downloads/>
- [5] <http://projectproto.blogspot.com/2010/09/android-bluetooth-oscilloscope.html>
- [6] <http://code.google.com/p/android-bluetooth-oscilloscope/>

EasyPIC v7

A PIC MCU development board born under the sign of Connectivity



When it comes to MCU development boards the Serbian manufacturer MikroElektronika (mE) has a reputation to keep up. Not only do they create and launch a wide range of such boards, they also keep improving them. Recently the seventh (!) generation of their EasyPIC development board for Microchip PIC microcontrollers was introduced. We tried it out for you.

By Clemens Valens (Elektor UK/US Editorial)

The EasyPIC board is a product within MikroElektronika's 'Easy' line of development boards like EasyAVR, EasydsPIC, EasyPSoC, and so on. The EasyPIC targets, as you might expect, 8-bit PIC microcontrollers from Microchip. My first contact with the EasyPIC series was a few years ago with the EasyPIC4. This was already a comprehensive and useful board, so you might wonder how it could be improved to get as far as a seventh generation.

The two boards look rather different, the '4' is much smaller than the '7' (26.5 x 22 cm) and it was laid out quite differently, but when you look closely they appear highly similar. Both have eight sockets catering for all the different PIC DIP devices from 8-pin to 40-pin; they also have in common a 4-digit 7-segment LED display, space for a 2 x 16 alphanumerical LCD, space for a 128 x 64 graphical LCD, pushbuttons, LEDs, pullup & pulldown resistors on every I/O pin with an extension connector for every port, an on-board program-

mer and in-circuit debugger, a power supply, RS-232 and USB ports and a few other things you might expect from a development board. Tracing the evolution of the EasyPIC, starting at generation 4, apparently the '5' added mainly a touch panel controller. The '6' took a bigger step by introducing SMD technology and DIP switches for pullup/down selection per GPIO, replacing the 7-segment display by a chip-on-glass (COG) LCD and adding a port expander, a 4 x 4 keypad, a 6-key menu keypad and a connector compatible with Microchip's ICD programmers/debuggers. Compared to the '6', the '7' seems to go back to its roots by replacing the COG display by the good old 7-segment LED display. The most obvious change however is the new component layout which I find much clearer. Where on previous generations all the GPIO LEDs, pushbuttons and pullup/down resistors were grouped by component type (i.e. all the LEDs together), they are now grouped by port.

The 'theme' of the '7' is Connectivity, as printed in the lower right-hand corner of the board. All ports now have three extension connectors (instead of one, two on the right side of the board and

EasyPIC v7 (\$ 149)

- supports over 250 8-bit PIC microcontrollers
- on-board mikroProg programmer/in-circuit debugger
- dual power supply (5 V & 3.3 V)
- 3 extension connectors per port + row of holes
- pushbutton, LED, pullup/down resistor per port pin
- 2 mikroBus slots
- RS-232 or serial-to-USB port
- USB, ICD, buzzer, I²C EEPROM
- 4-digit 7-segment display
- optional 2 x 16 alphanumerical LCD
- optional 128 x 64 graphical LCD with touch panel

www.mikroe.com/eng/products/view/757/easypicv7-development-system/

one on the left side) plus a row of extension holes. The connectivity theme is further illustrated by two slots (kind of) for add-on ('Click') boards compatible to MikroElektronika's mikroBus form factor (**photo 1**). Well over a dozen of these Click boards are currently available offering functions like Ethernet, Bluetooth, GPS, MP3, SD-card, etc. The PS/2 port has finally been replaced by a USB-to-serial interface (FTDI based).

Other changes can be found in the power supply which is now dual delivering 5 V and 3.3 V (one of which can be used for the board, selectable by a jumper), and in the mikroLCD programmer/in-circuit debugger that's now hidden by a metal cover. Removing it is easy enough and reveals a completely redesigned circuit (**photo 2**; doing so will obviously void your warranty). How exactly the MCU voltage is derived from the board voltage is not clear from the schematic.

Additions include a second thermometer connection — besides the traditional DS1820 it is now also possible to easily connect an LM35 sensor —, a buzzer, a type 24C08 8 Kbit I²C EEPROM and, verging on the silly but highly essential as I see it: a 5 mm mounting hole in each (rounded) corner of the board.

The build quality of the board is excellent. The PCB is extra thick (2.7 mm) so it will not easily bend and it has very clear component print on both sides. Super Duper mechanical supports for the LCDs are available and the variable resistors are spaced so wide apart that even thick fingers can adjust them comfortably.

What's in the box?

One thing I like about mE's products is their packaging. All of their products come in sturdy cardboard boxes that allow for easy storage when you don't use them, without kindergarten puzzle solving to get everything back in the box. The EasyPIC v7 box is rather large. In it you will find the board in an antistatic bag, a (red) USB cable, a DVD with software and documentation, a printed User's Guide, a manual for the mikroProg Suite for PIC, another manual for the on-board mikroLCD in-circuit debugger and the

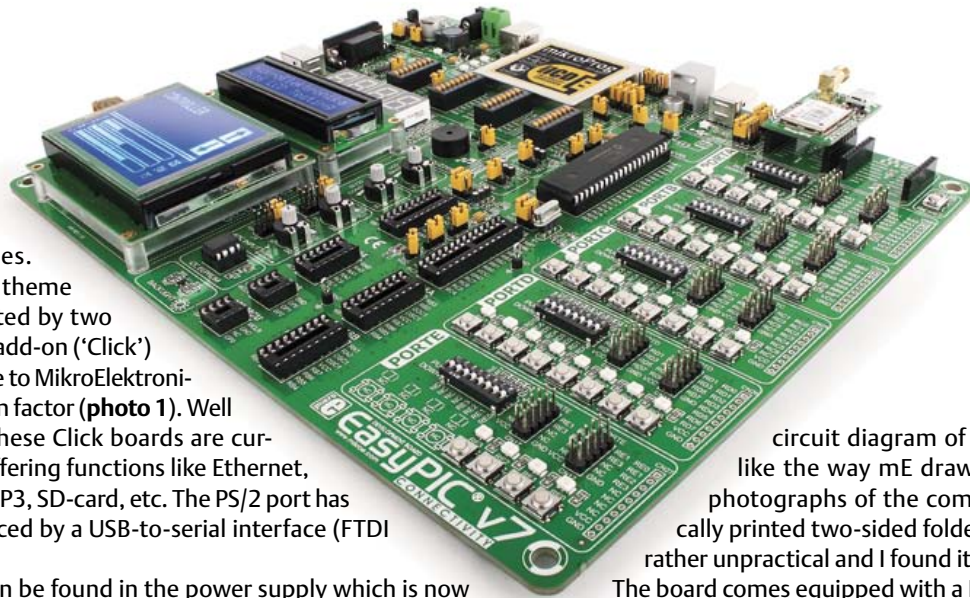


Photo 1. A mikroBus EasyGPS 'Click' board positioned on slot #1.

circuit diagram of the board. Usually I do like the way mE draw their schematics with photographs of the components, but the vertically printed two-sided folded sheet supplied now is rather unpractical and I found it difficult to read.

The board comes equipped with a PIC18F45K22, a 16 MIPS CPU with 32 KB of Flash memory and 1.5 KB of RAM. The board supports more than 250 PIC MCUs, so if this one doesn't suit you, just pick another.

Note that the LCDs and temperature sensors have to be ordered separately.

First power-on

The board has three power connection options: a 6.3 mm barrel socket (centre is positive), a screw terminal and USB. Although there are three USB sockets on the board, it can only be powered from the mikroProg USB socket. Since the board comes with a USB cable it is easiest to power the board over USB. Make sure jumper J6 is on USB, connect the USB cable to the board and to the PC and put the power supply switch in the ON position. The 7-segment display now shows a simple animation with the text "EP7", and then starts counting at a rate of about 20 Hz. The LEDs on port C scroll from right to left.

The board should also be detected by the PC's operating system. In order for the board to be recognised as well, a driver is required that has to be installed first. Note that (certified!) drivers for Windows Vista, XP and 7 are available only.

Software installation

Now that you know the board is working you can switch it off to install the software. Note that you can also use the board with non-mE compilers. Although a DVD is provided in the box you might as well download the latest version of the compiler of your choice from mE's website as there may be an updated version available anywhere. The installation is

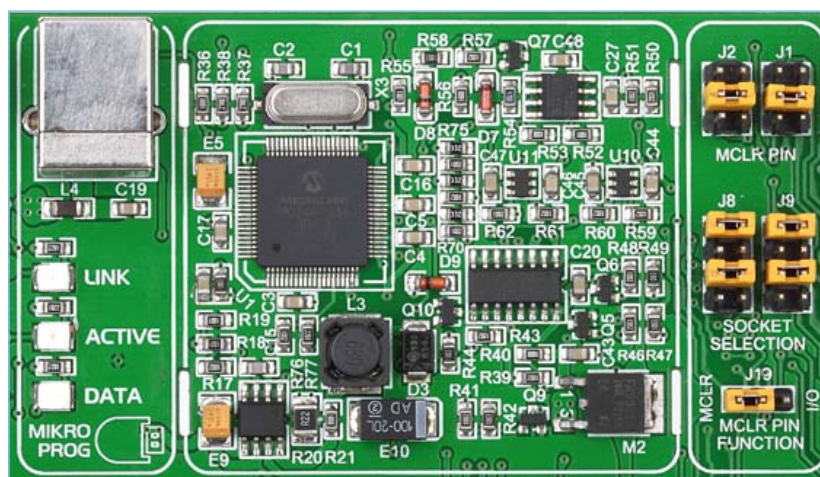


Photo 2. Inside the on-board mikroProg programmer/in-circuit debugger. Note that removing the hood will void your warranty.

straightforward and requires about 185 MB of hard disk space, two minutes and 16 mouse clicks. It creates two desktop icons, one for the compiler (mikroC Pro in my case) and one for the mikroProg suite for PIC.

When you launch mikroC you can quickly navigate to the examples using the Project menu and then Open Examples Folder. You can then select the Development Systems folder, which contains an EASYPIC7 folder, again containing example folders. From the LED Blinking folder load the project file LedBlinking.mcppi (sic), a simple 'Hello World' project. One mouse click is enough to compile it and start programming the executable in the MCU. Or so it should, except that programming doesn't start because you have to install the mikroProg driver first. The driver was copied to hard disk during installation, but it is not installed automatically for practical reasons and you have to do this manually. If, like me, you closed the Explorer window that opened up after the installation and you don't know where the driver is, in Windows XP, go to C:\Program Files\Mikroelektronika\mikroC PRO for PIC\mikroProg Suite installer\Drivers (replacing "mikroC PRO for PIC" by the name of your compiler). In case you have no idea how to install a driver manually, you will also find a PDF document with clear instructions right there.

With the drivers in place compiling and programming the LedBlinking example takes less than 15 seconds. All LEDs on port C and all segments of the 7-segment display should start blinking at a rate of 0.5 Hz.

All in all the EasyPIC v7 with mikroC PRO for PIC scores a helloWorld benchmark^[1] value of 945, which is actually pretty good.

When you got this far it's time to start sampling the other examples and of course begin developing your own applications (see the 'MyFirstProgram' inset). Note that some of the examples will not compile in the code-size-limited compiler demo versions, but precompiled versions are included for all examples, which is sure to meet with appreciation from newcomers.

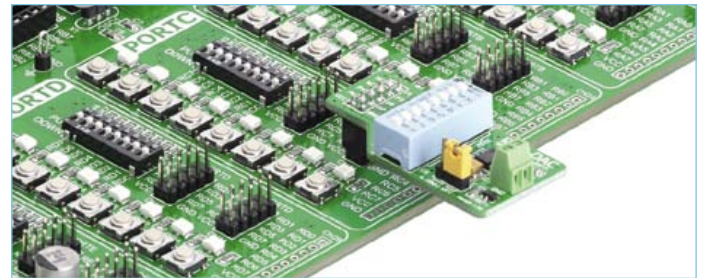


Photo 3. A DAC accessory board connected to port C.

Conclusion

The EasyPIC v7 is an attractive development board for 8-bit PIC microcontrollers offering all the peripherals you might expect from such a product. Many extension options allow you to easily add your own interface circuitry, and a choice of ready-made add-on modules is available from mE that can be plugged either on the mikroBus connectors or on the port extension connectors (**photo 3**). Build quality of the board is excellent and the mounting holes in the board allow for easy securing in out-of-lab situations. Its 3.3 V capability makes it compatible with low voltage parts and peripherals so widely available nowadays. Using the board with programming tools from other manufacturers is within easy reach thanks to a Microchip-compatible ICD programming connector. If you are serious about PIC-based project creation and development, this board is a must-have.

(120093)

[1] A Benchmark for Microcontroller Development Kits, Elektor February 2012, www.elektor.com/120096

MyFirstProgram

The review board came with an EasyGPS Click module (**photo 1**) so I decided to write a little program to use it. From mE's website I downloaded the EasyGPS demo code, but it turned out to be for their SmartGPS board as well as targeted at a PIC18F4520 on an EasyPIC6 board (issue corrected since). This meant that I had to change the target MCU (in the Project Settings window accessible from the View menu), enable the PLL x 4 fuse bit (Project -> Edit Project) and adapt the definitions of the graphical LCD interface (copied from another example). This made the program work on the EasyPIC v7 with graphical LCD (GLCD), except for the serial input. After some re-

searching I discovered that you have to disable the analogue inputs of the PIC18F45K22 to enable the serial input (add `ANSEL = 0` just before `UART1_Init(9600)`).

Although the EasyGPS module runs from 3.3 V the program worked fine with the PIC18F45K22 running from either 5 V or 3.3 V. For those who use the demo version of the compiler or who do not have a GLCD I also added some code for a 2 x 16 alphanumerical LCD. You can download my program from www.elektor.com/120093; it includes two precompiled executables, one for the 2 x 16 LCD and one for the GLCD.



Coming Soon! The **RL78** Green Energy Challenge

\$20,000
in cash
prizes

Competition
starts
March 26,
2012

With an incredible ecosystem of hardware, software and third-party vendors, Renesas' family of RL78 MCUs are optimized for efficient power consumption and deliver up to 41 DMIPS at 32MHz. These versatile MCUs offer a true low-power platform for the most demanding 8- and 16-bit embedded applications.

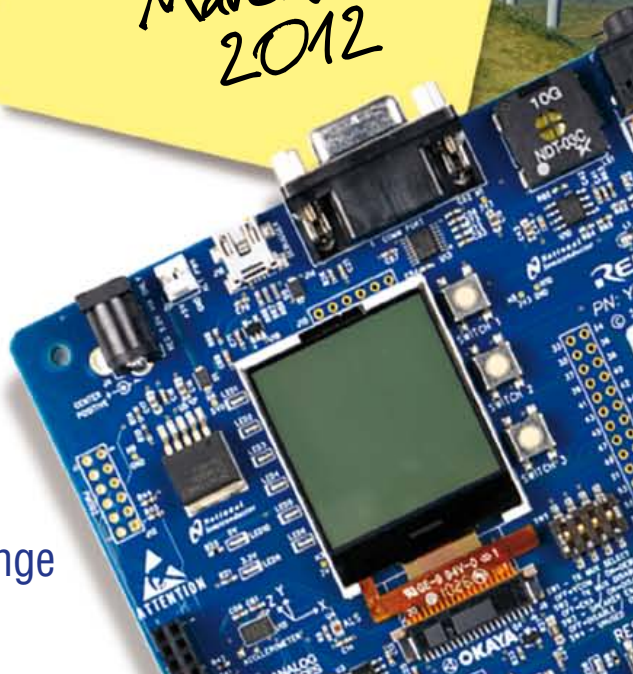
Renesas, along with *Circuit Cellar* and *Elektor*, invites you to experience true low power by developing a green energy application using the RL78 MCU and IAR tool chain. Succeed and win a share of \$20,000 in cash prizes!



In association with *Elektor* and *Circuit Cellar*

Be the green you see in the world and get ready for the RL78 Green Energy Challenge.

www.circuitcellar.com/RenesasRL78Challenge



Hard Disk Activity Monitor

A 10-LED indicator to show how hard it's working

By Karsten Böhme (Germany)

Almost every PC case is fitted with an LED on the front panel which flickers whenever the hard disk(s) is accessed. A single flickering light is better than nothing at all but a 10-LED linear scale showing hard disk loading as a percentage would be really handy!



Features

- A display using 10 LEDs indicates hard drive activity
- Displays hard drive activity in 10 % steps
- Crash proof operation (independent of PC software)
- Neat compact module fits easily inside a PC case

It can sometimes happen, even using the most up to date PC with super fast hard disks and a powerful processor, that while some complex application is running you find yourself staring at the monitor, not quite knowing why the machine is performing slowly or has indeed decided to stop responding altogether. Sound familiar? You cannot be sure if the program has truly got itself into a pickle or intense hard disk activity is responsible for the sluggish behaviour. In such situations it would be helpful if you had a piece of add-on independent 'hardware' which gave some insight into what was happening inside your PC.

This design idea shows real-time hard disk activity in the form of a 10-bit linear display. One advantage of the design is that it runs independently of the PC software so it won't place any additional load on the PC's proces-

sor. It is also unaffected by internet-borne bugs or viruses (assuming there are none in the microcontroller firmware to start off with) and will function even if an application running on the PC has crashed. Despite the market for PC case modding and customisation being relatively large there doesn't appear to be any similar hard disk indicator available, but on second thoughts I guess that is what we have Elektor for!

More LEDs!

The average PC usually has just a single flickering LED giving you a rough indication that the hard disk(s) is being accessed. The modification described here shows hard disk activity in a much more informative linear display using ten LEDs (each step represents 10 %). The display is refreshed five times per second, giving a good responsive indication.

The question is how has the author managed to gain access to this information because it is not output from the hard disk controller in either analogue or digital form. His solution is to treat the on/off signal to the hard disk LED as a pulse-width modulated signal, simply measuring the on to off ratio of this signal gives an indication of how hard the disk is being used. This approach works surprisingly well. The wire from the motherboard normally connecting to the hard disk LED is instead connected to the

input on the hard disk monitor board. A microcontroller then reads the state of this signal during 200 ms windows to determine the amount of time that this signal is high during each period. A high for the entire period will result in all ten LEDs lit while a low for the entire period will turn them all off. The photo of the author's PC shows all ten LEDs lit. The first seven (the range from 1 to 70 %) uses green LEDs, the next two (71 to 90 %) are yellow and the final position is red (91 to 100 %).

Decoder and LED driver

From the functional description and circuit diagram you will be aware that the circuit uses a microcontroller. A low cost Atmel ATtiny2313 microcontroller together with an optocoupler and LEDs are connected as shown in the diagram in **Figure 1**. The wire which originally connected the motherboard to the hard disk indicator LED is instead connected to the monitor board input where it now drives an optocoupler. Its LED input stage forms a good substitute for the indicator LED and reduces the possibility of any signal mismatch. The original hard disk indicator LED does not go unused because the circuit provides an output to drive it in the same way that the signal from the motherboard did originally. IC1 drives the indicator scale consisting of ten LEDs driven via 470 Ω series resistors.

Note. Readers' Projects are reproduced based on information supplied by the author(s) only. The use of Elektor style schematics and other illustrations in this article or the availability of project (software) downloads from the Elektor website does not imply the project having passed Elektor Labs for replication to verify claimed operation.

To increase the number of LEDs it would be necessary to use a larger controller with corresponding firmware. It is important to ensure that not only the maximum current output from each pin is within spec (10mA here) but also that the current drawn by the complete IC (60 mA for this device) is not exceeded. If a brighter display is required (which many users find unnecessarily distracting) more efficient low-current LEDs can be used.

The complete circuit is powered from a 5 V voltage source (use the 5 V from a spare power connector in the PC normally used to power a hard or floppy drive). A 10-way pin header is included on the PCB to allow in-circuit programming of the microcontroller.

Construction

For this project the author has designed a PCB for mounting the entire circuit including display LEDs. The PCB layouts for the component side (file: PCB_top.jpg [1]) and the underside (file: PCB_bottom.jpg [1]) of the board show that despite the use of SMD components the layout is quite well spaced out. The author has specified 1206 outline SMD resistors which are relatively easy to fit by hand. It is not anticipated that mounting any of the other components will present any great difficulty. All of the components except for the LEDs are fitted to the component side. The LEDs are mounted on the PCB underside at a suitable stand-off so that the LEDs fit snugly into the front panel holes when the PCB is fixed (on spacers) behind the front panel. A 1:1 scale photocopy of the PCB layout can be cut out and used as a template to drill the ten LED holes in the PC front panel. The LEDs can also be arranged as in the author's prototype board (see Prototype.jpg at [1]).

It is important to specify that the microcontroller is clocked by a 4 MHz internal clock when the firmware is flashed to memory. Without this step the controller runs noticeably slower and affects the display refresh rate. The screen shot in Figure 2 shows how the clock fuses are configured in the Atmel AVR Studio 4 development environment. The author has written the software in C using the CodeVision compiler. The source code is well documented to allow simple modification should you be tempted to

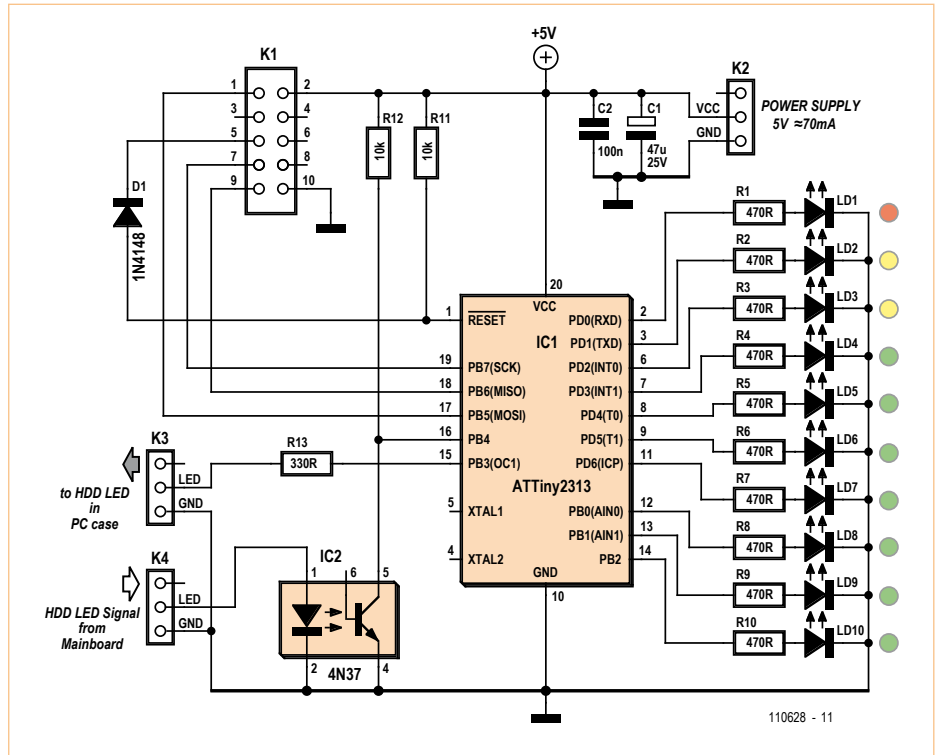


Figure 1. The hard disk monitor circuit diagram is quite simple.

add your own improvements. For the less adventurous there is also a complete Hex file available which can simply be flashed to the microcontroller memory. Both files form part of the free downloads for this article [1] together with the PCB design files in Eagle format. To see the display in action click on the video in the link below.

(110628)

Internet Link

[1] Video, software and author's PCB artwork files: www.elektor.com/110628

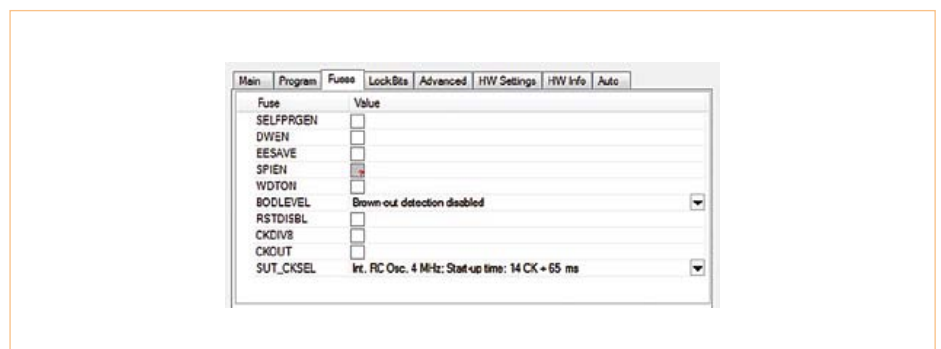
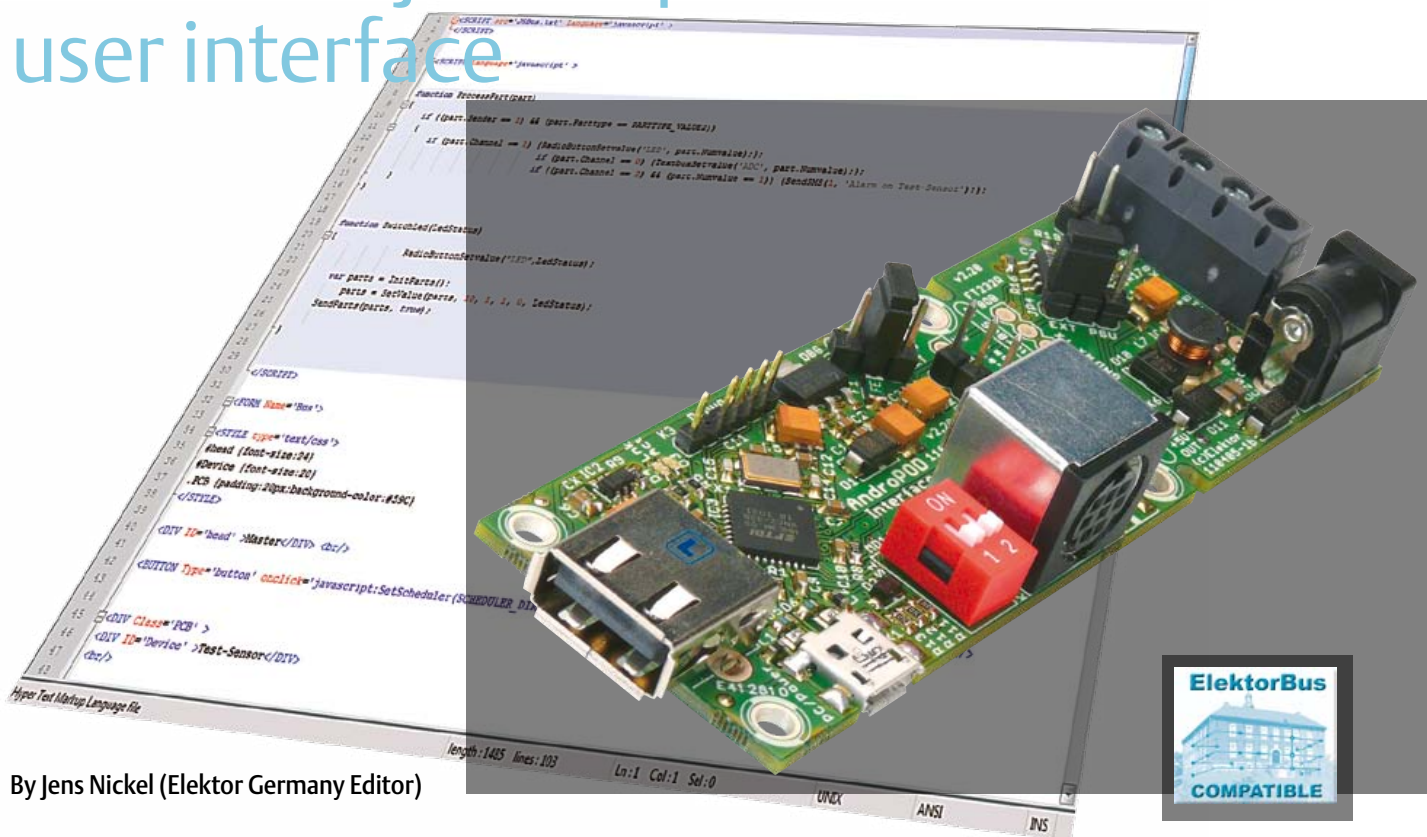


Figure 2. The fuse settings in AVR Studio 4 to give an internal 4 MHz clock.

AndroPod (2)

HTML and Java implement a tailored user interface



By Jens Nickel (Elektor Germany Editor)

In the first article of this series we saw how easy it is to connect your own circuitry to an Android smartphone using the AndroPod interface. Our aim there was to make the software aspects as simple as possible for all users. Building a user interface for your own project with HTML is very easy if you use our free Android app. As always with Elektor, the app software can be downloaded free of charge, and you can modify it if necessary because everything is open source.

To control your own circuitry, you basically need an Android app that includes a tailored user interface. The obvious approach is to use the powerful Android framework and the Java programming language to implement this user interface. However, programming in the Android environment has a very steep learning curve for beginners, although it may be worth the effort because more and more projects using Android are being published in Elektor. Bernhard Wörndl-Aichriedler [1], a student at the Hagenberg campus of the Upper Austria University of Applied Sciences and the developer of the AndroPod Interface, has put together a small Android tutorial in German and English (see [2] for download). A user guide in English is also available on the Android developers site [3]. Among other things, it explains how to install the Eclipse development environment [4] and how to extend it for Android programming.

After starting a new Android project, the first task is to implement the basic functions (receiving and transmitting bytes over the

serial interface). The AndroPod developers provide the Java class `AndroPodConnection` for this purpose (download at [5]). It can be linked into your own Android projects as described in the tutorial. The next task is to build the user interface, which consists of controls such as text boxes, buttons and so on. The structure of a complete AndroPod app is described in the inset. The source code for this app can be downloaded and used as a starting point for developing your own projects.

Same control interface for smartphones and PCs

There is also another option for generating the user interface: using HTML and Javascript. Even for beginners, it's easy to learn how to build an HTML page with the necessary controls and program the control logic in Java. Furthermore, the result is fundamentally independent of the operating system that is used. You generate the user interface only once, after which you can use it on an Android smartphone, a PC or another computer platform, as long as a suitable

serial interface is available. For PCs, Elektor offers a USB to RS485 converter [6].

In theory the generated HTML pages can be displayed directly on the smartphone in the web browser present on every Android device. However, for security reasons it is not possible to persuade a normal Web browser to send data to the AndroPod or receive data from the AndroPod. For this we need our own Android app, which is a sort of special browser. It provides the HTML user interface and implements communication over the serial interface (Figure 1). The advantage of this is that this app can be provided ready for use and does not need to be modified by the AndroPod user. The Elektor-BusBrowserForAndroPod app is available on the Google Android Market site for free download. This makes downloading and installation on the smartphone or tablet especially easy. The app is also available on the Elektor website in the form of an .apk file (roughly equivalent to an .exe file in the PC world) and as source code [5]. If you have completed the hardware test described in the first article in this series, this app is already installed [7], but you should update it to the latest version.

All you need to do after this is to download your HTML pages to the smartphone, which is not difficult. However, let's first look at how these HTML pages should be put together.

Simple messages

Our Android app receives and sends 16-byte messages based on the Elektor Message Protocol [6]. The data rate is 9600 baud, which automatically makes it compatible with many upcoming Elektor projects. This protocol was developed for the ElektorBus [8], but it is by no means limited to ElektorBus hardware or the RS485 bus. The app works equally well if the data is received and transmitted over the mini DIN connector or the pin header. In this regard it is important to note that in this case the RS485 driver must always be disabled by removing the jumper at JP4.

In the simplest case an ElektorBus message has the following structure (see also Figure 2):

Byte 0	170 = AA _{hex}
Byte 1	0
Byte 2	0
Byte 3	Receiver address (1-127)
Byte 4	0
Byte 5	Sender address (1-127)
Bytes 6-13	User data
Bytes 14-15	Optional; may be used for a checksum or CRC

The default address of the Android smartphone is 10, but this can be changed in the Settings menu of the app. In the simplest case you communicate with just one other node, such as your own circuitry, which is fitted with an AVR microcontroller or similar device. If you assign this microcontroller the address '1', messages from the smartphone to the microcontroller have the sequence '170, 0, 0, 1, 0, 10, ...'. In the other direction this is '170, 0, 0, 10, 0, 1, ...'.

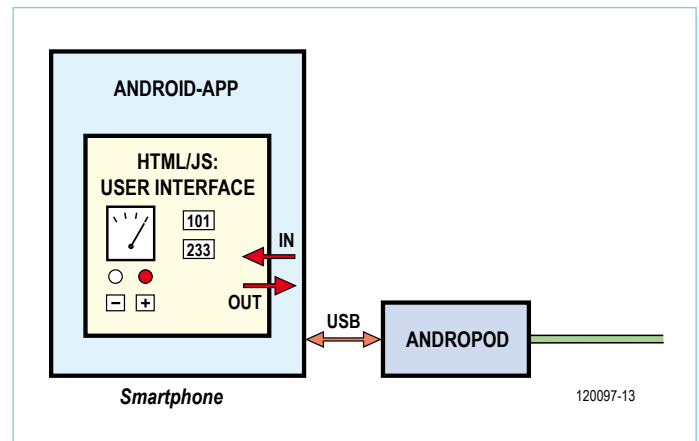


Figure 1. Our Android app includes HTML pages that can be tailored by users for their own projects.

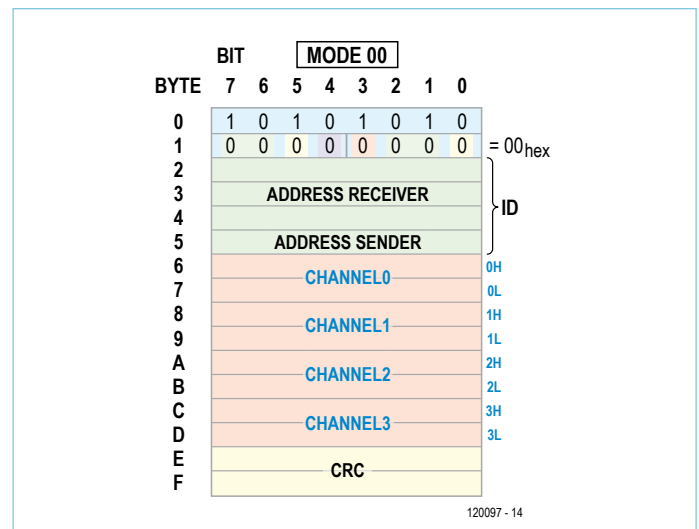


Figure 2. Structure of an ElektorBus message, which has a fixed length of 16 bytes.

The user data bytes are subject to the restriction that the value '170' is not allowed because it is used as the start byte. If you are not happy with this (or something else, such as the fixed message length), you are free to develop your own protocol and modify the code of the Android app accordingly. Detailed Android expertise is not necessary for modifications such as changing the message length, as long as you do not alter the basic structure of the app (see inset).

Receiving data

After the Android app has received 16 bytes, it first parses the message into three parts: sender, recipient and user data (8 bytes). The user data is then passed to the HTML page of the AndroPod user. The application developer does not need to worry about the details because there is a Javascript library called 'JSBus' available for linking into your DIY HTML pages. It receives and decodes the 8-byte data packets (see Figure 3), under the assumption that the data is structured according to the Elektor Application Protocol. This protocol is ideal for transmitting measurement values (in integer or

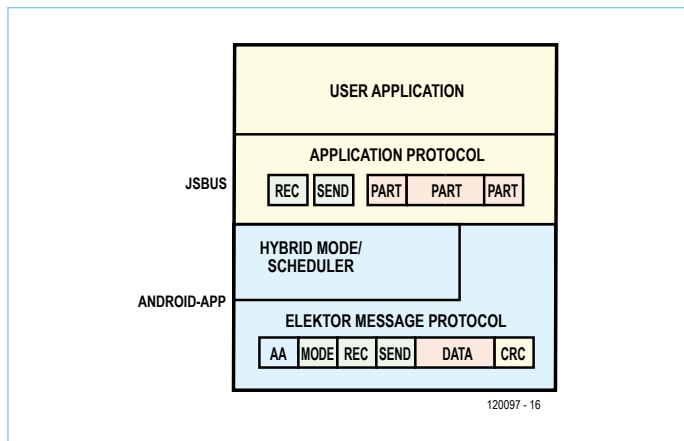


Figure 3. The Android app parses the messages, and Javascript decodes the user data.

```
<SCRIPT src='JSBus.txt' Language='javascript'
></SCRIPT>
<SCRIPT Language='javascript' >

function ProcessPart(part)
{
    // own code
}

</SCRIPT>
```

You can then use your own code to process the measurement values, for example by placing them in an HTML text box for display. Here as well the JSBus library offers assistance. Suppose you want to have the first measurement value (channel 0) displayed in a text box named 'MyTextbox'. Simply enter the following line in your code:

```
if (part.Channel == 0)
{TextboxSetvalue('MyTextbox', part.Numvalue);};
```

The received measurement value (range -1023 to +1023) is always held in the variable `part.Numvalue`. A summary of all of the functions of the JSBus library and much more information is available at [9].

Various demo programs for AVR microcontrollers can be downloaded from the ElektorBus project site [8]. The microcontroller site also provides ready-made code that you can modify for your own purpose, so you don't have to reinvent the wheel. Additional microcontroller software compatible with the protocol described here will become available in the course of the year.

Sending data

The following Javascript code can be used to send a control message from the smartphone, for example to switch a pair of LEDs on or off (via channels 1 and 2):

```
function SwitchLed(LedStatus1,LedStatus2)
{
    var parts = InitParts();
    parts = SetValue(parts, 10, 1, 1, 0,
    LedStatus1);
    parts = SetValue(parts, 10, 1, 2, 0,
    LedStatus2);
    SendParts(parts, true);
}
```

Here `LEDStatus1` and `LEDStatus2` must be set to 0 for 'off' or to 1 for 'on'. This code first generates an empty array of parts (message units). Then two parts are added, each of which contains a setting to be transmitted to the LEDs over the two channels in order to switch them on or off. Finally, the function `SendParts(...)` transmits both parts in a single message.

The function `SwitchLed(...)` can be invoked by pressing an HTML button, in response to an incoming message, or under time control.

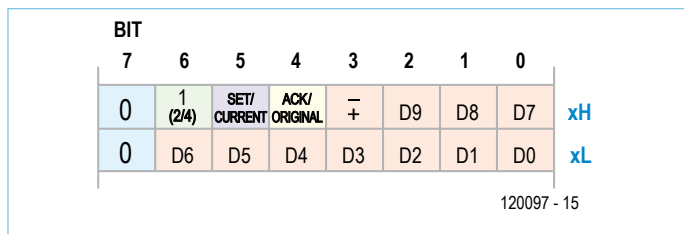


Figure 4. Two bytes allow values in the range of -1023 to +1023 to be represented, which is sufficient for many applications.

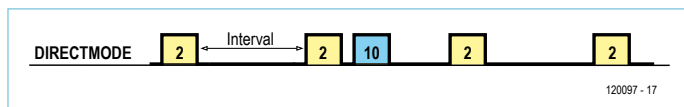


Figure 5. The simple Direct mode avoids collisions in one-to-one communication.

floating point representation), changing the unit and scale factor (e.g. from volts to millivolts), configuring setpoint values, signalling alarms (values outside specified limits), and quite a few other things in realm of instrumentation and control. This protocol will also be used in future Elektor projects (one thing we have in mind is a multimeter), and it can of course be used to control ElektorBus hardware. For example, an AC power switch board will be described in the next edition.

Various types of data can be transmitted in the messages. For example, up to four measurement values (range -1023 to +1023) for four channels could be transmitted in a single message, using two bytes for each value (see Figure 4). When the Javascript library receives an 8-byte data packet, it splits it into four parts corresponding to four measurement values. For each received part it calls the Javascript routine `ProcessPart {...}`, which must be integrated in your own HTML pages.

This is not as difficult as it sounds. Simply include the following lines in your HTML file:

Examples of all three options are described in articles [9] and [10], and the corresponding HTML files are available for download on the associated Elektor website pages. We recommend that beginners have a look at these code segments (most of which are very short) and adapt them to their own purposes.

Collision avoidance

If you use the RS485 extension, you need to give some thought to avoiding message collisions on the bus, since all messages on the bus travel over a single pair of data lines (half duplex). If the sender and the receiver try to use the bus at the same time, the result will be digital chaos. Once again, the Android app looks after most of this for you. It basically supports two modes: Direct mode and Hybrid mode.

Direct mode is particularly suitable for the type of one-to-one communication described above (for example between the smartphone and a sensor). It presumes that the external circuitry, such as an instrument or sensor, sends measurement values or status messages at fixed intervals. If a control message must be sent from the smartphone to the external circuitry, it is sent immediately after a measurement data message has been received (see **Figure 5**). The application protocol also provides the option of designating the sensor that sends measurement values in a particular interval [10]. With a data rate of 9600 baud, a measurement interval of 100 ms does not present any problems.

Use the following line of Javascript code on your HTML page to enable Direct mode:

```
SetScheduler(SCHEDULER_
DIRECTMODE,0,0,0,0,0,0,0,0);
```

This command can also be linked directly to an on-click event for an HTML button, which for example could be labelled 'Direct Mode On':

```
<BUTTON Type='button' onclick='javascript:
SetScheduler(SCHEDULER_
DIRECTMODE,0,0,0,0,0,0,0,0) ` >Direct Mode On</
BUTTON>
```

The syntax of HTML and Javascript is fairly easy to learn. A brief overview for beginners is available at [9].

In contrast to Direct mode, Hybrid mode is suitable for communication with several nodes on a shared bus. This is handled by a scheduler whose only task is to allocate send time to the bus nodes. The Android app includes a suitable scheduler. When you start it, you tell it the addresses of all the nodes that must normally be taken into account. For example, you can start the scheduler with the following command if the AndroPod interface is connected to an RS485 bus with two nodes at addresses 1 and 2:

```
SetScheduler(SCHEDULER_ON,2,1,10,0,0,0,0,0);
```

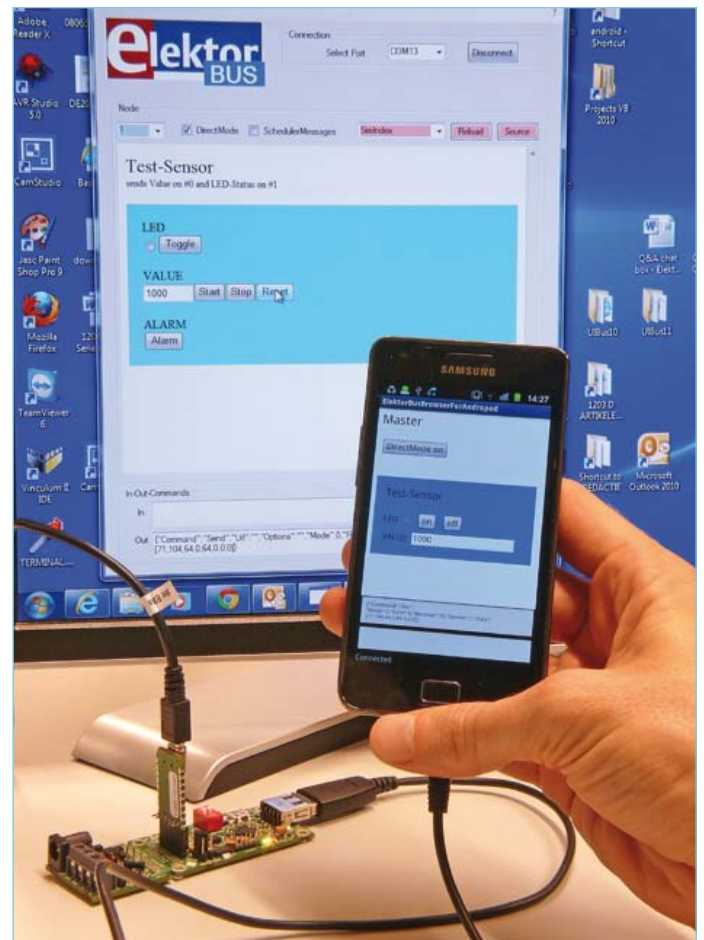


Figure 6. Initial testing can be performed by connecting the AndroPod to a PC (for example over the BOB FT232 USB to TTL converter).

Up to eight node addresses may be stated as parameters. Be sure to include '10' if the Android smartphone occasionally has something to say.

Demo

Up to now we've been concentrating on the theory; now it's time to get practical with a small demo application. Microcontroller boards such as those used in the ElektorBus project [6] are naturally ideal for a simple test. However, this can also be done using a PC, which can be connected to the AndroPod in various ways as described in the previous article in this series – such as with the BOB FT232 USB to TTL converter [5] as illustrated in **Figure 6**. The downloadable zip file includes an executable file named ElektorBusElectronics-Simulator.exe [5]. This is a PC program that can send and receive messages. You can use it to simulate a suitable opposite party for the smartphone, such as a sensor. An HTML page is used to represent this simulated hardware. To enable the PC program to find this page, the UIBus folder in the download folder [5] must be dragged onto the desktop of the PC.

The HTML page to be displayed on the smartphone must also be installed on the smartphone, and the JSBus.txt library must be present on the smartphone. By default the app looks for these two files in a folder called ElektorBusBrowser on the SD card; this can be changed in the Settings menu. But how can you get the files into

this folder?

The first option is to transfer the files from the PC to the smartphones in the conventional manner, such as by using a suitable USB cable or the Bluetooth link. If you use a cable, you must disable USB debugging before the transfer and enable it again after the transfer (see the description in the first article of this series [7]). You also have to create the folder for these files yourself.

However, there is a more convenient manner that entirely eliminates the need to mess about with cables. As in the previous article in this series, you use the AdifController software from the AndroPod developers for this, but this time you select the 'Files' tab. You should see the smartphone under 'Detected Phones' on the left; you may have to briefly switch off power to the AndroPod interface to achieve this. Then click 'Browse' and select the 'UIBus' folder on the desktop. After this, press the 'Upload' button.

You should use the file manager to verify that the files have been

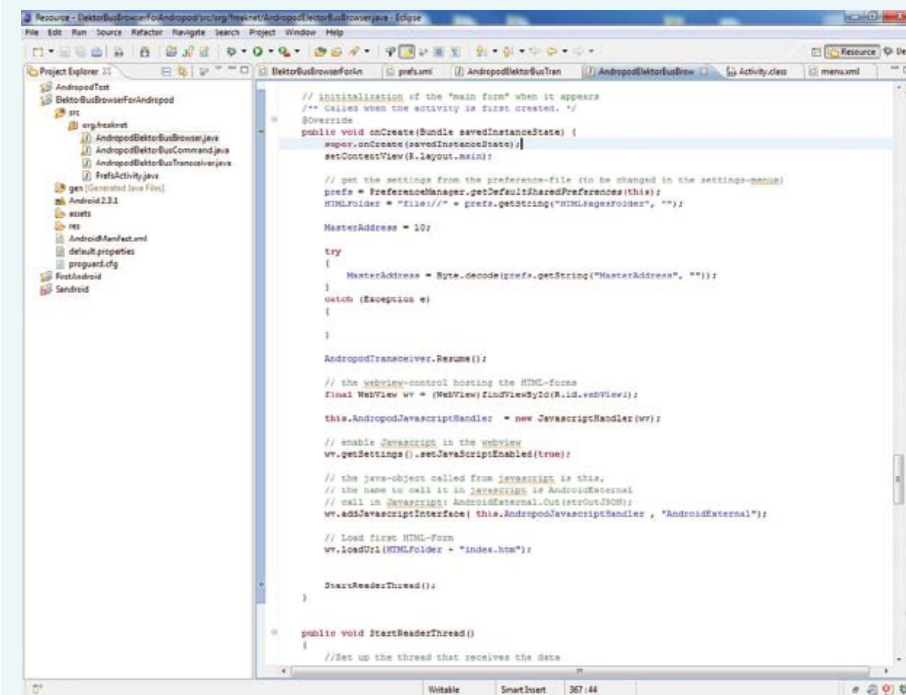
correctly stored on the memory card of the smartphone.

Testing

Now start the app on the smartphone and the program on the PC. The simulated measurement device appears in the large window of the PC program (Figure 7). First configure the appropriate COM port (or virtual port) at the top, and then press the 'Connect' button.

The PC simulates a sensor that sends data periodically, such as an electricity meter in the cellar. On the HTML interface, start data transmission by pressing the 'Start' button. You can also use the 'Toggle' button to simulate switching an indicator LED on or off. Now the values should be visible on the smartphone displaying the 'master' interface (Figure 8). Note the slight delay in the indicator LED display. This results from the fact that a LED status change is

AndroPod app structure



thread programming and network programming in a good Java book. This should enable you to understand the rough structure of the app and to modify it for your own purposes. If you want to program a native user interface (implemented directly in Android), you also need to have or acquire Android expertise. Naturally, the Android Developer's Guide [11] is the first point of reference for this. The author can recommend [12] as a good book.

Our Android app uses three classes. `AndroPodElektorBusTransceiver` initialises a TCP connection on port 1337, closes it, and opens it again as necessary. The methods `TransmitPacket` and `ReceivePacket` send and receive (respectively) 16-byte data packets, with the start of each packet (message) marked by the value 170 (AA_{hex}). If you wish to use a different message length, you must modify the code here.

The best way to study the structure of an app is to examine an example such as *ElektorBusBrowserForAndroPod*, which can be downloaded from the Elektor website [5] as an Android project (for Eclipse). The source code is located in the project subfolder 'src'. As already mentioned several times, the Android framework is based on the Java programming language and uses the same syntax as well as major libraries. Basic (or more than basic) knowledge of Java is therefore very helpful. It is also worthwhile to read through the discussion of

Objects of the class `AndroPodElektorBusCommand` serve as containers for message data to be sent or received. Each object contains a byte array named `Rawdata`, which holds all 16 bytes of one message. The method `parseRawData` parses the message packet. It populates the variables `Mode`, `Receiver`, `Sender` and `Data`, where `Data` is again an array that holds the user data (8 bytes) of the message. In the other direction, the 16-byte

not transmitted immediately from the PC, but instead waits until the transmission of the next measurement value. The LED status and measurement value are transmitted in different channels in the message.

Now it's time to send something from the master to the sensor. You can use the 'On' and 'Off' buttons on the master interface to set or reset the indicator LED on the sensor. However, you must first enable Direct mode to allow message transmission in the opposite direction. As the smartphone display has lower resolution than the PC monitor, the button for this is located below the HTML interface. Note how the LED indicator of the smartphone responds when the 'On' or 'Off' button is pressed. If you have been paying attention, you know the reason: the smartphone waits until the next message from the PC (with a measurement value and the old LED status) has come lumbering in from the PC before it sends the new LED status, because only then is the bus guaranteed to be free.



Figure 7. The PC program simulates an external instrument or sensor.

array `Rawdata` is populated when the values `Mode`, `Receiver`, `Sender` and `Data` are passed to the constructor of `AndroPodElektorBusCommand`. The class also packs these values into a special string so that the decoded message can be passed to the already displayed HTML/Javascript page (see [9] for more information on the structure of these In and Out commands). Any developers who wish to divide the bytes inside the message in a different manner may modify this class accordingly.

The class `AndroPodElektorBusBrowser` looks after displaying the received data and accepts user entries that trigger the message transmission. Both of these activities are performed inside an HTML page to obtain the platform independence mentioned in the main article. The only function of `AndroPodElektorBusBrowser` is to cause this HTML page to be displayed in an Android control of type `WebView`. Received messages (or more precisely, the strings generated from them) are passed to the HTML/Javascript by the method `In (...)`. In the other direction, the Javascript embedded in the HTML page passes a string to the method `Out (...)` when a message needs to be sent. The scheduler is also located in this class. The method `Out (...)` therefore does not send the message immediately, but instead buffers it. The message is sent only when the scheduler allows it to be sent (except in Direct mode). Part of the class is tailored for the ElektorBus protocol; the code starting with the method `onCreate` will be of interest to readers who wish to write their own apps. This class is an Android activity that is called when the app is launched (in a manner of speaking, it is the main window of the app). Android experts will know that along with `onCreate` (initialisation) the methods `onStop`, `onPause` and `onResume` should be overwritten; they contain code that must be executed when stopping the

application, when switching to a different activity, and on resuming the activity. The method `StartReaderThread` (described below) is also important for the `AndroPod`.

When the application is launched, `onCreate` is first called by `AndroPodElektorBusBrowser` or by a user-programmed activity. The command `AndroPodTransceiver.Resume` activates the 'transceiver' in the class `AndroPodElektorBusTransceiver`. It first uses the method `Resume (...)` to open a server socket on port 1337. After this an extra thread is started and the code of the method `run (...)` in `AndroPodElektorBusTransceiver` is executed. As you can easily see, this code is contained in an endless loop. This loop continuously checks whether a connection to the `AndroPod` interfaces already exists; if it does not, a connection is established. This thread is what allows the interface to be connected while the app is running.

At the end of the initialisation routine `onCreate` the actual read thread is started by the command `StartReaderThread()`. The associated code is located a few lines below, again inside a `run()` method. The `while` loop there is iterated as long as the connection to the interface exists. The line `ReceivedMessage = AndroPodTransceiver.ReceivePacket();` reads in the next 16 bytes (beginning with the start byte `AAhex`). After this the message packet can be processed. For the sake of completeness it should be mentioned here that the scheduler also runs in its own thread.

The effort for sending a message is not especially large: you simply call the method `AndroPodTransceiver.TransmitPacket (...)` with an `AndroPod ElektorBus` command as a parameter.



Figure 8. Screenshot of the control interface on the smartphone.

If you have a second mobile phone available, you can also test the text messaging function. Before doing this you must tell the app the number of the mobile phone to which it should send a text message in the event of an alarm. This is done in the Settings menu of the app, which was described in the first article of this series [7]. Three parameters can be configured after you press the menu button of the mobile phone at the bottom left and then click 'Settings' :

- The file where the app looks for the HTML and Javascript files
- The default text messaging number
- The ElektorMessage address of the android smartphone (default value '10')

In rare cases the app may freeze after the settings are changed. If this happens, end the app and then restart it.

After you enter the number, you can test the text messaging function by clicking the 'Alarm' button in the PC program window. This causes a '1', which indicates a sensor alarm, to be sent in channel 2 (otherwise unused) of the next message.

The HTML page displayed on the mobile phone then uses issues the following Javascript command to cause a text message to be sent:

```
SendsSMS("1", "Alarm on test sensor");
```

The '1' in the first parameter field means that the text message should be sent to the number configured in the app settings. Alternatively, any desired mobile phone number can be specified directly here.

The source code of the two HTML pages used for this test can be displayed using the PC program. The page that simulates the sensor

is called 'SimIndex', while the page displayed on the smartphone is called 'Index'. Use the combo box with the red background to select the page, and then click the 'Source' button.

An acknowledgement mechanism should be built into a real application to ensure that important messages are not lost if something goes wrong. The Elektor Message Protocol and the Application Protocol provide various options for this purpose [8].

Prospects

Although the concept of the HTML-based approach may appear somewhat complex at first glance, it has enormous advantages. Programming an HTML/Javascript page is much easier and cheaper than programming a user interface in Android Java, and the HTML version is platform independent (the same interface can also be used on the PC). Furthermore, HTML is presently on a roll, and it is reasonable to assume that more and more PC software for Elektor projects will run in a browser environment.

In the next edition we will describe a 'utility board' with two relays and two inputs, along with associated firmware compatible with ElektorBus. This allows lamps and other AC powered loads to be switched using an Android smartphone, and if you wish you can send status indications by text messaging. Other projects, including instrumentation projects, are also in the works.

(120097-1)

[1] www.xdevelop.at
 [2] www.xdevelop.at/#category=projects&subcategory=1&anchor=6
 [3] <http://developer.android.com/sdk/installing.html>
 [4] www.eclipse.org/downloads/
 [5] www.elektor.com/120097
 [6] www.elektor.com/110258
 [7] www.elektor.com/110405
 [8] www.elektor.com/elektorbus
 [9] www.elektor.com/110517
 [10] www.elektor.com/110708
 [11] <http://developer.android.com/guide/index.html>
 [12] Marko Gargenta, Learning Android, O'Reilly: <http://shop.oreilly.com/product/0636920010883.do>

Elektor Products and Services

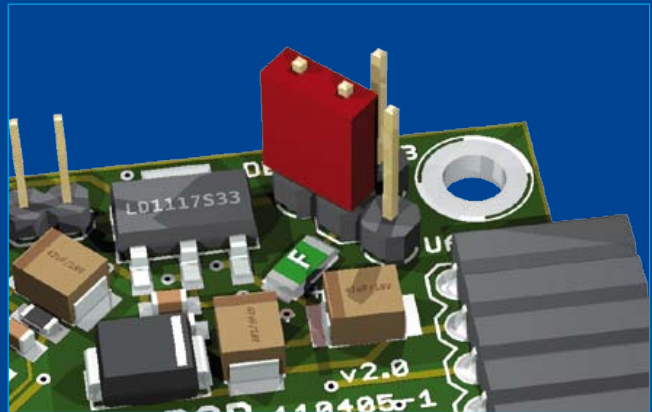
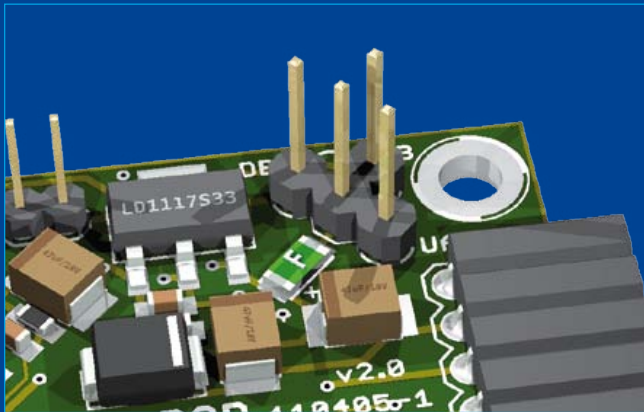
- AndroPod with RS485 extension, fully assembled and tested board: # 110405-91
- BOB FT232 USB/TTL converter, fully assembled and tested: # 110553-91

- RS485/USB converter, fully assembled and tested: # 110258-91
- USB-A to Micro-B cable
- Smartphone AC power adapter with Micro-B USB plug
- Software download (free)

All products and downloads are available via the web page for this article: www.elektor.com/120097

AndroPod's 3-way jumper

At first sight, it's actually got four, but let's take a closer look...



DEBUG position

By Antoine Authier (Elektor Labs)

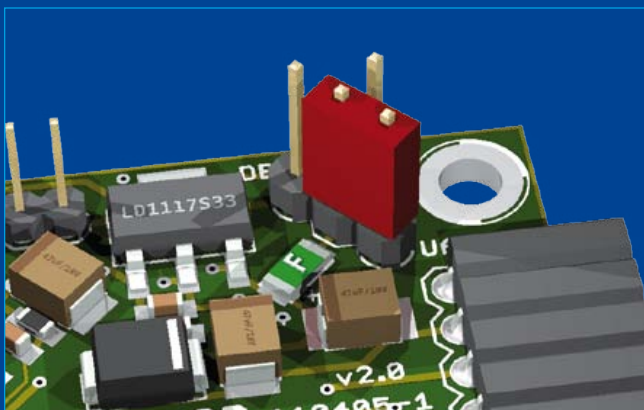
The Elektor AndroPod is our command interface to allow an Android phone to communicate with other circuits. One of its notable features is the variety of power sources possible: USB device, UART, debugging module, RS-485 module. That sort of flexibility involves something of a challenge for the designer: how to avoid power supply conflicts without resorting to expensive mechanical components? What's more, after a long time spent poring through distributors' catalogues and discussing it with my Chinese friends, I couldn't help noting that it's hard, if not impossible, to find a nice, small 4-position slide switch that can handle say 500 mA, or even 1 A.

These questions led me to ponder even more than usual when I was designing the PCB, so that the project would remain practical, easy to use, and accessible to everyone. One solution

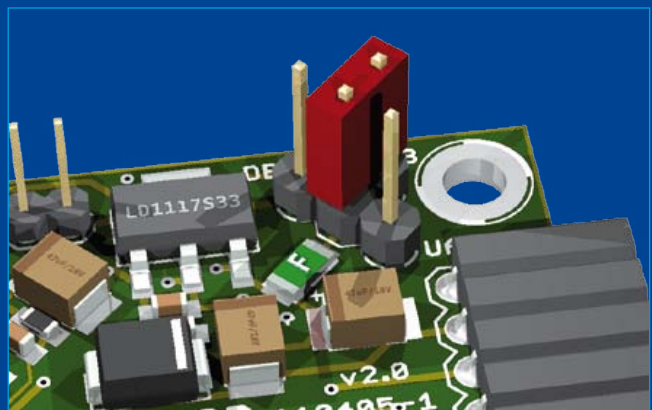
would be to use multiple jumpers... but this would be rather risky, since with the large number of configurations possible, sooner or later a lapse of concentration is going to lead to a false move... and the worst will happen. The solution came when my colleague Ton Giesberts, doubtless feeling sorry for me lost amid the maze of jumpers, ventured ingenuously "Isn't all that a bit much, though?" Without waiting for me to reply, he added "Why not arrange them as a half-star, erm, triangle...?" Eureka! Cool. Brilliant. Awesome. Elementary, my dear Ton! In point of fact, this configuration, although perhaps at first sight a bit exotic, prevents multiple connections and never allows the circuit to be powered by more than one source at a time. Have fun with AndroPod.

Antoine (thanks Ton!)

(120076)



USB position



UART position

Wanted: SMPSU DC-out filtering

By Raymond Vermeulen (Elektor Labs)

DC/DC converters! They're often quite useful, sometimes indispensable and most of the time very energy efficient. I am starting to apply them increasingly, following in the footsteps of real pros. The other day, while working on some battery powered circuits I found myself in need of a very efficient DC/DC converter to generate the voltages required by the circuit. While I was busy calculating the reservoir capacitor for the power supply, something came to mind I learned during my student days about passive filters and switch-mode power supplies.

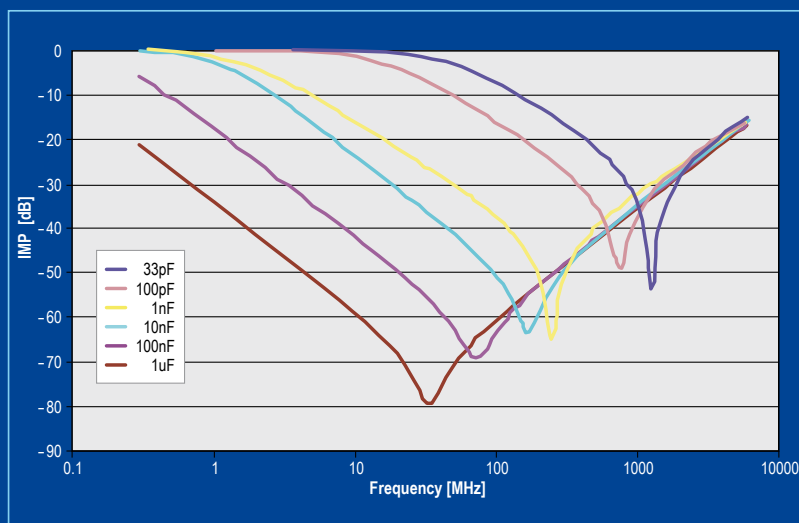
Unlike linear regulators such as the venerable LM78xx series, these switching power supplies often produce lots of ripple voltage at their outputs. Digital circuits and ICs are affected less by this than analogue circuits, so depending on the application it is necessary to smooth any ripple by using filter circuits.

Datasheets of switch-mode PSU (SMPSU) components often suggest a bare minimum of output capacitance for filtering, for example a 1- μF capacitor between the output and circuit ground. Many times you tend to go for something of the order of about 10 μF . But...

As a rule of thumb, the larger the value of the capacitor, the sooner it starts to function as an inductor when increasing the applied frequency (see **graph**). Also, inherent to its oper-

ating principle the switch-mode power supply generates ripple frequencies over a broad spectrum. This is not very helpful as components are often more easily thrown off by high frequency power supply ripple than low frequency noise on their supply rails.

To enhance the filter spectrum, several capacitors with different capacities can be paralleled. Returning to our example, instead of a single 10- μF capacitor, you could deploy a small 100-nF capacitor in parallel, preferably one with good RF specifications (for example, a low ESR ceramic X5R capacitor) in order to enhance the ripple rejection within the entire applicable bandwidth.



If this still isn't enough filtering for your application, you can also put a small inductance of about 0.1 μH in series with the supply output, between the 10 μF and 100 nF filter capacitors, effectively creating an LC filter. Furthermore it's considered good practice to implement an extra RC filter as close as possible to the supply pins of each analogue IC. All of the above measures cost a little more on the component side and need a little more PCB surface. They do however keep trouble at bay and you will probably save yourself a headache or two, three.

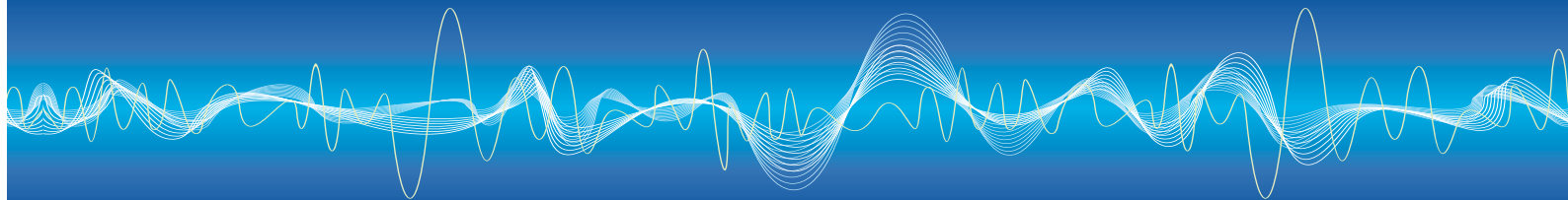
(120144)

Plug-o-(d)rama

By Thijs Beckers (Elektor Editorial)

While researching a subject for his monthly Retronics piece in Elektor my colleague Jan Buiting stumbles upon all sorts of oddball items that have existed in the world of consumer electronics, with a slight bias towards ancient Philips audio and TV equipment. A while ago Jan showed me a random collection of plugs collected over the years. Being too young to have wit-

nessed the actual use of any of these plugs, I was curious to know their intended use 'in the old days'. Let's have a quick look at what was out there many years ago and is now sadly neglected — but not forgotten. Most of these plugs or sockets are polarised, meaning they have a mechanical provision for incorrect connection with the mating part on the equipment. Let's survey the picture and see what all this primitive connectivity is all about.



1. A 1950s radio loudspeaker plug with a centre prod to prevent it being plugged in a 220 VAC power outlet. More about this under item 4.

2. A heavy duty three-pin plug with screw-on securing cap. Dimensioned for high current or high voltage use. Often used to connect loudspeakers including 100 volt systems.

3. Philips DIN-to-5-way-IEC record player adapter with five silver-plated pins (cleaned with a glass fibre brush). Rare.

4. Loudspeaker plug to connect a pair of high frequency 'satellite' loudspeakers to high-end Philips radio type B6X62A. Non-polarised; red dot applied to identify LSP phase. The brown version (1) is for connecting the bass speaker cabinet. The B6X62A radio has built in active two-way filtering with 400 Hz roll off!

5. Cable version of Philips 5-way-IEC audio plug together with its female counterpart. Used stubbornly by Philips only, this standard failed to get customer and/or industry acceptance. A special tool is required to open the connector.

6. 3-pin plug for Philips microphones or 800-ohm loudspeakers and its female counterpart. Loudspeaker connection to OTL (output transformerless) tube amplifiers (approx. years 1955–1964). Polarized. The third pin is optional for carrying the treble (>400 Hz) loudspeaker signal. Note the screened cable for a dynamic microphone on the female connector.

7. Early IEC style non-earthed AC power appliance connector. Polarized.

8. VHF band antenna plug for 240-Ω ribbon cable. With centre prod.

9. As 8, but rounded pins and no centre prod.

10. VHF/UHF band TV antenna connector for 240-Ω ribbon cable. Note dual-diameter pins. Exact application unknown.

11. As 8. Polarized (how?). Probably antenna; exact application unknown.

This is just a small selection from what must be an infinite range of plugs, receptacles and connectors having been around for decades prior to Ethernet and WiFi. And apparently there still isn't enough diversity, because the number of connector types is still growing.



Among today's most common models are USB, USB 3, Firewire (4- and 6-pin), Thunderbolt, HDMI, DisplayPort, DVI (-A, -D, -I, -D HDCP, mini-, micro-), VGA, S-video, RCA, TOSLINK (for S/

PDIF), XLR, TRS (2.5 mm, 3.5 mm jack (1/8 in), 6.35 mm jack (1/4 in), banana (a.k.a. wander plug), Sata, RJ-45 and the list goes on, not even mentioning a galaxy of power connectors.

So when will this eruption of plugs subside? Who knows? Leafing through our history books, we should be able to determine when it started.

That said, we would like to challenge you to locate the 'Iunctio Antiquissima' and send us a clear picture of your oldest piece of connector to connectorcontest@elektor.com. Quite possibly your entry makes it to a Retronics article or a laurel in one of our upcoming editions...

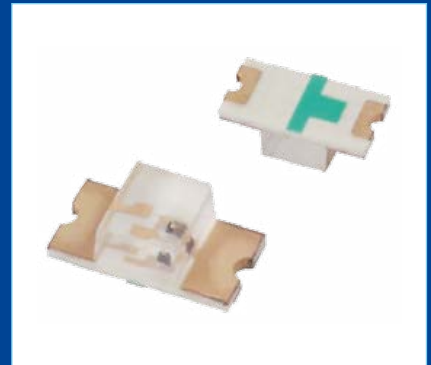
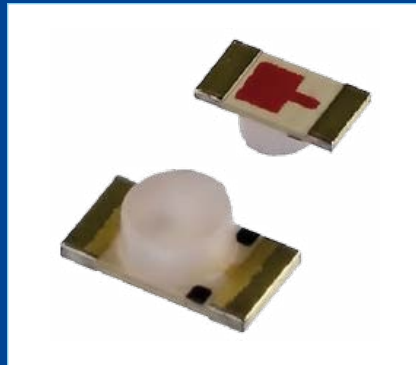
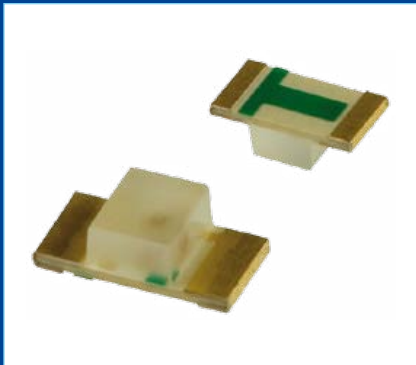
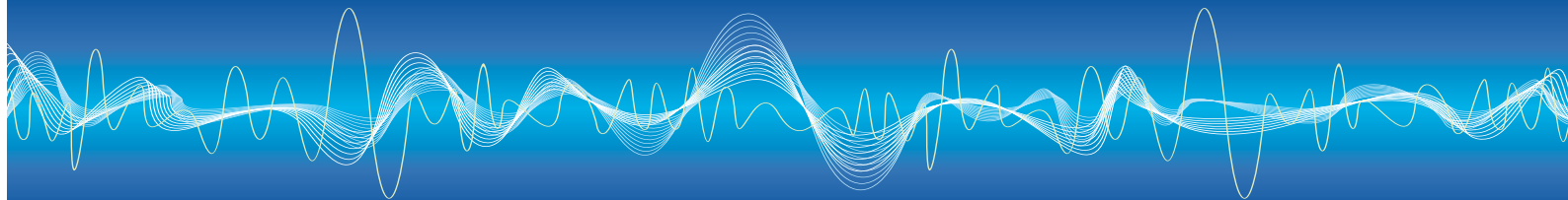
(120027)

SMD LED polarity

By Raymond Vermeulen (Elektor Labs)

Soldering polarised components always requires some extra attention. Sometimes, mostly with small SMT components, it's

not quite obvious which pin is which, so I noticed when I wanted to solder two SMT LEDs in a type '0603' housing. Besides the fact that those 0603 housings are pretty small, I also seemed unable to find any markings of the anode or cathode pin...



I decided to take pot luck and just solder them in, faintly hoping they would function instantly. I had a 50% chance of mounting them correctly, right? But of course, they were both the wrong way around. A little annoyed, I desoldered them and put them back on the board with the correct orientation.

I could have left it at that, but something just didn't feel right, — designing a circuit and not knowing every little detail about one's components. So after referring to the datasheet of the LEDs, it *does* turn out there is some sort of marking (of course there is!), even on really tiny components like 0603 LEDs. On the backside of the LED there should be a symbol of some kind to indicate anode or cathode. It's unthinkable for a standard

for this indication to exist, so every manufacturer cheerfully employs their own symbol for the purpose. The illustrations show some of the specimens we were able to locate, but this collection is far from complete. There's a wide variety of stripes, dots, rectangles and arrows that can be used, and even combinations of these.

So be sure to check the datasheet or check the polarity with a small test setup — a little lesson I learned — blatantly obvious to the pros, a little less so to newbies, but in the end making soldering these tiny components a fraction easier.

(120145)

The Bus Conductors

By Jens Nickel (Elektor Germany Editor)

Up until now development of the ElektorBus project has been confined to a benchtop with all bus components just connected by short lengths of wire, now it's getting serious. In the next edition we are planning to present an 'installation board' which allows the control of AC powered equipment. Following that we will be installing a pilot bus in our publishing HQ. While here in Elektor Castle we need to order quite a large amount of cable to interconnect the offices on all the floors. High time we put some thought into the type of cable we will be using.

For test purposes we took a length of cable and hooked up an experimental node described in [1] and an AndroPod Android bridge board [2] connected to a smartphone. With the Elektor BusBrowser App running it is possible to control the remote experimental node and continuously display values of ambient light measured by its light sensor.

In the store room we uncovered a spool of likely looking cable (KROSCHU Schaltflex CY Style 2571 [3]). The cable has 10 conductors, two of which are a twisted pair. Firstly we used four individual conductors to wire the data A and B, 12 V and GND then secondly in accordance with RS485 standard we used the twisted pair to convey signals A and B. In a typical office environment with a cable length of just over 100 feet (we would have

gone further but we ran out of cable) both configurations performed flawlessly. We monitored error-free data packets continually flying back and forth over the bus. The bus' DC power supply was also used to drive a relay at one of the nodes and functioned without problem. A colleague, Raymond Vermeulen was on-hand with an oscilloscope to check the rising and falling edges of the bus signal: all within spec, perfect!

For the next test we used a 100-ft length of Cat 5E Ethernet cable (4 x twisted pairs, shielded). One twisted pair was used for the databus connections A and B, another twisted pair (with the two wires twisted together at either end) for the 12 V connection and another pair for the GND wire. Again with this cable length there were no surprises and the system performed without fault. A short video of the tests (including a brief ground-level tour of our workplace) can be viewed on the ElektorIM YouTube channel [4].

(120198)



- [1] www.elektor.com/110258
- [2] www.elektor.com/110405
- [3] www.kroschu-cable.de/documents/downloads/Schaltflex%202008-07-08%20%5Be%5D.pdf
- [4] www.youtube.com/watch?v=rbDSTXNARmw

Subscribe now to the leading US-based computer applications magazine specializing in embedded systems and design!



Select your personal subscription at www.elektor.com/cc-subs

12 editions per year for just

Digital: \$38
Print: \$63
Digital + Print: \$90

**CIRCUIT
CELLAR®**
THE MAGAZINE FOR COMPUTER APPLICATIONS

Platino in Arduino Land

A guide to integrating custom hardware



Platino, the ultra versatile AVR board presented in the October 2011 edition is compatible with the Arduino programming environment. Yet there are wheels within wheels because Arduino does not natively support all microcontrollers embraced by Platino, which means that a bit of code porting is in order. Luckily, most of the bits and bobs you require to do this without too much effort can be found on the Internet for free. For your convenience, they got collected and bundled.

By Clemens Valens (Elektor editorial team)

As you probably know the Arduino Uno and its predecessors are based on the 28-pin ATmegaXX8 family of AVR microcontrollers from Atmel. The larger Arduino Mega uses an ATmega2560 in a 100-pin package. Platino supports not just AVR controllers in 28-pin DIP packages but also the 40-pin ones Arduino cannot work with. If you design your own board you might be tempted to move to AVRs in SMA packages with yet other pin counts but not supported natively by Arduino. In this article I will attempt to show how to modify the Arduino environment in such a way it can be used easily with your own custom hardware. Allow Platino to be your guide.

Integrating custom hardware in Arduino 1.0

Actually it's fairly easy to integrate your own compatible hardware in the Arduino programming environment. All you have to do is to add a folder to the hardware folder of the Arduino distribution (i.e. `arduino-1.0\hardware\`). The new folder, which you should name properly — `platino` in this case — will contain the files specific to your hardware as well as, in some cases, links (sort of) to the standard Arduino core files.

To prevent problems with future versions of the Arduino IDE it is probably easier to simply copy the contents of the default `arduino` hardware folder (`arduino-1.0\hardware\arduino`) to your new folder (like `arduino-1.0\hardware\platino`) and modify your copies only. The disadvantage of the method is that you will not automatically integrate any future improvements of the core files, but on the other hand you do not have to modify anything either when a core file changes.

You will now have a folder with four subfolders, and two files:

- `bootloaders` (folder)
- `cores` (folder)
- `firmwares` (folder)
- `variants` (folder)
- `boards.txt` (file)
- `programmers.txt` (file)

The `bootloaders` folder contains the bootloader(s) for your hardware. If your board has a micro that's not natively supported by the included bootloaders you will have to modify (one of) them. This may seem a daunting task, but in reality it is not so complicated as long as you stay with the AVR family. More on this further on.

The `cores` folder contains the Arduino core files. These files define the Arduino functions and some of them should be adapted to suit your hardware. In a sense it is the Board Support Package (BSP), just to use a highly fashionable embedded electronics expression.

The folder `firmwares` contains the executables for the boards that use an ATmega8U2 for communication with the IDE. You may delete this folder if your hardware doesn't. Platino doesn't.

The `variants` folder is new in Arduino 1.0. It allows an easy way to define variants of boards that share a large common base. In the case of Platino, we can define two variants here, one for Platino with a 28-pin micro and one for Platino with a 40-pin device. Each variant has its own subfolder containing the file `pins_arduino.h` that used to sit in the `cores` folder in previous versions of Arduino. If needed you can add other variant-specific files here too.

Delete the file `programmers.txt`; you only need it if you have to define your own programmer. If you keep it without modi-

fyng it you will be faced with double entries in the IDE menu Tools->Programmer.

Finally, the file `boards.txt` contains information needed by the IDE to use the right folders and protocols etc. with the selected hardware. The boards listed in this file end up in the IDE menu Tools->Board where they can be selected. We will first edit this file before making changes to the other core files.

boards.txt

This file defines the boards the IDE is aware of. It has to be edited so that your custom hardware is recognised by the IDE. To do this, open it in a simple text editor like Notepad. You will see a list of blocks like the one in **listing 1** (without the line numbers).

Remove all the blocks except for one that will be adapted for your hardware. If you do not delete the unchanged blocks you will end up with duplicate entries in the Tools->Boards menu.

All lines in the block in listing 1 start with `uno`. This is a unique identifier for the board in all the `boards.txt` files available in the Arduino installation. The IDE will only recognize one of the boards that have an identical identifier.

The order of the lines is irrelevant.

Line 1 containing the `name` label defines the name of the board as shown in the IDE. You can use a lot of text here; I don not know how many characters are acceptable, but there is probably a limit so don't overdo it.

Line 2 & 4 (`upload.protocol` and `upload.speed`) are parameters passed to `avrdude`, the standard programmer utility used by the IDE. Arduino 1.0 includes the latest version of `avrdude` that's aware of the protocol `arduino`. Versions of `avrdude` included with previous Arduino distributions do not recognise this protocol and you should use `stk500` instead.

Line 3 defines the maximum size of memory available for a user application. It is calculated by subtracting the size of the bootloader from the maximum memory size. Here we have a 32 KB device with a 512-byte bootloader giving a maximum of 32,256 bytes.

Lines 5 to 11 define how to load the bootloader, and which one, from within the IDE (Tools->Burn Bootloader). These lines all specify parameters for `avrdude`. The Fuse and Lock settings of course depend on the MCU being programmed. The bootloader file from line 9 should sit in the `bootloaders` folder's subfolder specified in line 8. So long as you don't try to burn the bootloader from within the IDE all these parameters may be fictional. These lines also allow you to specify a bootloader that's not compatible with the Arduino communication protocol (as long as `avrdude` knows how to handle it) and you can use a bootloader from, for instance, Wiring if you prefer so.

Line 12 lets you specify the MCU on your board. Note that there are different types available for P-type AVR's. This is important because their ID byte is not the same as the non P-types. The optional 'A' suffix does not have any effect.

Line 13 specifies the clock oscillator frequency of the micro in Hz; this value corresponds to the quartz crystal frequency and is needed by the timer and UART functions of Arduino. If you want to know where, just search the core files for the parameter `F_CPU`.

Listing 1. A structure defining a board for the Arduino 1.0 IDE from the file `boards.txt`. The line numbers were added by me for (my) convenience.

```
1  uno.name=Arduino Uno
2  uno.upload.protocol=arduino
3  uno.upload.maximum_size=32256
4  uno.upload.speed=115200
5  uno.bootloader.low_fuses=0xff
6  uno.bootloader.high_fuses=0xdc
7  uno.bootloader.extended_fuses=0x05
8  uno.bootloader.path=optiboot
9  uno.bootloader.file=optiboot_atmega328.hex
10 uno.bootloader.unlock_bits=0x3F
11 uno.bootloader.lock_bits=0x0F
12 uno.build.mcu=atmega328p
13 uno.build.f_cpu=16000000L
14 uno.build.core=arduino
15 uno.build.variant=standard
```

Talking about the core files, they should reside in the hardware folder's subfolder specified in line 14 (here: `arduino-1.0\hardware\arduino`). Line 15 specifies the `variants` folder's subfolder that holds the file `pins_arduino.h` as needed for the particular flavour of your board.

The last two lines may point anywhere, which allows you to use core files from another board with the board you are working on. These are the 'links' I mentioned before.

Listing 2 shows how I adapted the structure from listing 1 to suit my Platino board with an ATmega164p. I did the same for all other flavours I could think of yielding a total of 16 new custom boards in the IDE. The variant is `ATmegaXX4` (40-pin devices), I also created a variant `ATmegaXX8` for the 28-pin devices. The other core files are identical for both variants since I used `#ifdefs` there to guide the compiler in the right direction.

Listing 2. A board structure for Platino with an ATmega164p processor. The final two lines ensure the right core files are used for this board.

```
platino164p.name=Platino 164p(a) @ 16 MHz
platino164p.upload.protocol=arduino
platino164p.upload.maximum_size=15872
platino164p.upload.speed=115200
platino164p.bootloader.low_fuses=0xff
platino164p.bootloader.high_fuses=0xdc
platino164p.bootloader.extended_fuses=0xfd
platino164p.bootloader.path=optiboot
platino164p.bootloader.file=optiboot_
    platino164p.hex
platino164p.bootloader.unlock_bits=0x3f
platino164p.bootloader.lock_bits=0x0f
platino164p.build.mcu=atmega164p
platino164p.build.f_cpu=16000000L
platino164p.build.core=platino
platino164p.build.variant=ATmegaXX4
```



```
int analogRead(uint8_t pin)
{
    uint8_t low, high;

    #if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
    if (pin >= 54) pin -= 54; // allow for channel or pin numbers
    #elif defined(__AVR_ATmega3204__)
    if (pin >= 54) pin -= 54; // allow for channel or pin numbers
    #elif defined(ATMEGA_X4) || defined(__AVR_ATmega16__) || defined(__AVR_ATmega32__)
    if (pin >= 21) pin -= 21; // allow for channel or pin numbers
    #else
    if (pin >= 14) pin -= 14; // allow for channel or pin numbers
    #endif

    #if defined(__AVR_ATmega3204__)
    pin = analogPinToChannel(pin);
    ADCSRB = (ADCSRB & ~(1 << MUX5)) | (((pin >> 3) & 0x01) << MUX5);
    #elif defined(ADCSRB) && defined(MUX5)
    // the MUX5 bit of ADCSRB selects whether we're reading from channels
    // 0 to 7 (MUX5 low) or 8 to 15 (MUX5 high).
    ADCSRB = (ADCSRB & ~(1 << MUX5)) | (((pin >> 3) & 0x01) << MUX5);
    #endif

    // set the analog reference (high two bits of ADMUX) and select the
    // channel (low 4 bits). this also sets ADLAR (left-adjust result)
    // to 8 (the default)
    #if defined(ADMUX)
    ADMUX = (analog_reference << 6) | (pin & 0x07);
    #endif
}
```

Figure 1. A fragment of the file wiring_analog.c, showing a code modification for the Platino port.

pins_arduino.h

Once the custom board(s) are properly defined we can start editing the pin definition file for the board pins_arduino.h in the variants folder. If we stick to our Platino-with-ATmega164p example, this file can be found in arduino-1.0\hardware\platino\variants\ATmegaXX4.

In this file you have to define the number of pins capable of functioning as a GPIO, the number of analogue inputs, the pins having PWM output capability, the ones that can be connected to a timer, the SPI port, etc. All these definitions are macros and data structures and you will need the MCU's datasheet to do this properly.

The order of the data structures determines the numbers of the digital pins in Arduino. Digital pin 0 is the first entry of every data

code defining Platino functions in C++ so I added an include statement #include "Platino.h" to the list at the end of the file, just below where it says #ifdef __cplusplus.

I also had to add a definition for the analogue voltage reference INTERNAL1V1 and INTERNAL2V56 for the 40-pin devices. I did this by adding the statement #elif defined(ATMEGA_X4). The macro ATMEGA_X4 is defined in the header file devices.h I created and for which I added an include statement at the top of the file.

Tone.cpp

Modifying this file is not really necessary because it is a bit weird; it even includes a comment confirming this. That said, for future compatibility it's probably better to adapt it anyway. I did this by adding an #elif defined(ATMEGA_X4) statement and copying the same code as for the other devices to it.

Wiring_analog.c

The function analogRead has to be modified so that it can map digital pin numbers to analogue inputs. This is only one line of code, again introduced by an #elif defined(ATMEGA_X4)... statement (figure 1).

Wiring_private.h

This file contains just interrupt definitions and since 40-pin AVR's

One Object Fits All Platino Peripherals

structure. Be careful to use the same order in all structures, otherwise pins and functions will get mixed up. Analogue input pins are treated differently and not in this file.

Before modifying anything, make sure you understand what you are doing. Although it is not rocket science it's not always clear either. If you look at my pins file you may notice #ifdefs concerning ATmega16 and ATmega32 devices. I could have created another variant for these 40-pin chips but the differences are too small to justify this. BTW, the MCU definition (__AVR_ATmega16__ and friends) result from the MCU definition in line 12 of listing 1.

Modifying core files

The cores folder (arduino-1.0\hardware\arduino\cores\) contains a folder arduino that in turn contains the Arduino core files; there are 36 at the time of writing this article. That may seem a lot to port for your own board, but it turns out that only a small fraction needs some work. For my Platino port I had to modify just four files (of my local cores copy), since I decided to create new files for the real stuff. Here are my core file mods:

Arduino.h

This file is included by many other core and library files, which makes it a good place to add links to your own code. I wrote my

have three external interrupt inputs, a definition has to be added here. Again look for the #elif defined(ATMEGA_X4)... statement to see how I did this.

You may be wondering why I added compiler directives for the 40-pin devices only. It's because the 28-pin devices are already supported by the Arduino core and Platino is compatible with the classic Arduino, like the Uno, so there was really nothing to change.

Adding core files

If your hardware only features the peripherals that are also on a standard Arduino board, then you might skip this paragraph. Platino on the other hand integrates an LCD, pushbuttons and/or rotary encoders, a buzzer, an RGB LED and configuration jumpers. These extra peripherals can of course be handled by libraries (that live in arduino-1.0\libraries)—the Arduino LiquidCrystal LCD library for instance works just fine with Platino—but that implies a more complicated port with more locations to keep an eye on. Consequently I made the Platino functions an integral part of the Arduino environment and add my custom code to my copy of the core.

Adding files to the core is straightforward since the compiler will compile anything it finds in the core folder. Here is, in alphabetical order, what I added and why:

PicoScope[®] Range

PC OSCILLOSCOPES

devices.h

Some definitions are needed to allow conditional compilation of core files for Platino. Since Platino supports several AVR devices and these devices can come in different flavours, `#ifdef` statements tend to become long. I therefore decided to create two groups, one group for 40-pin devices (`ATMEGA_X4`) and one for 28-pin devices (`ATMEGA_X8`). Add new devices here.

LiquidCrystal.cpp & h

As mentioned before the default Arduino LCD library works fine with Platino, but a modification is needed in order to use it in my Platino object (see below). Also for ease of compilation and maintenance I copied this library to my `cores` folder.

This library does not have a default constructor so I added the line `LiquidCrystal(void) {}` to the class definition in the header file. You don't know C++? Well, if you really must know, I needed it because my Platino class defines a LiquidCrystal object without initialising it and the function to do this was missing from the library.

Platino.cpp & h

To provide easy access to Platino peripherals I created a Platino object in the form of a handy class. Invoking the class will give you direct access to the Platino peripherals. It is actually more of a wrapper for the individual peripheral classes, but one that's capable of handling the configuration solder jumpers of Platino. More about this class later.

PushButton.cpp & h

These two files implement a class for a pushbutton object. It features debounced and direct access to pushbuttons, and it's used by the rotary encoder class (see below).

RotaryEncoder.cpp & h

Rotary encoders are essentially no more than two (or three if they have a push function) mechanically coupled pushbuttons. Consequently the class that handles them treats them as such. This class does not use (input-capture) interrupts and is not controlled from a timer, the user has to 'tick' it periodically. Call it from the sketch's `loop` function but beware of (blocking) functions from other libraries or wait loops that may slow down or block the main loop.

The optional pushbutton of a rotary encoder is not handled by this class and should be handled as a normal pushbutton using the PushButton class.

Using it

With the porting all done you can start using it. The first step is to select your board from the IDE's Tools->Boards menu. If you did the adaptation of the `boards.txt` file properly and you do have access to one of the programmers known to the IDE, then you should be able to burn the bootloader to your device directly from within the IDE (Tools->Burn Bootloader). With the bootloader programmed you can continue with the sketch.

The hard part here will probably be finding a usable bootloader. I

PicoScope 2100 Series	PicoScope 2200 Series	PicoScope 2205 MSO
		
1 Channel 10 to 25 MHz Bandwidth 50 to 100 MS/s Sampling 8 bits Resolution (12 bits enhanced) 8 or 24 kS Buffer memory Price from £125 \$206 €151	2 Channel + AWG 10 to 200 MHz Bandwidth 100 MS/s to 1 GS/s Sampling 8 bits Resolution (12 bits enhanced) 8 to 40 kS Buffer memory Price from £159 \$262 €192	2 Analogue, 16 Digital + AWG Channels 25 MHz Bandwidth 200 MS/s Sampling 8 bits Resolution 48 kS Buffer memory Price from £349 \$658 €483

PicoScope 3200 Series	PicoScope 4000 Series	PicoScope 4262 Series
		
2 + External trigger and AWG 60 to 200 MHz Bandwidth 500 MS/s Sampling 8 bits Resolution (12 bits enhanced) 4 or 128 MS Buffer memory Price from £399 \$658 €483	2 or 4 Channels 20 to 100 MHz Bandwidth 80 to 250 MS/s Sampling 12 bits Resolution (16 bits enhanced) 32 MS Buffer memory Price from £499 \$823 €604	2 External trigger + Low distortion AWG Channels 5 MHz Bandwidth 10 MS/s Sampling 16 bits Resolution 16 MS Buffer memory Price from £749 \$1235 €906

High end features as standard:
Advanced digital triggers, Persistence display modes,
Mask limit testing, Serial decoding



FOR THE FULL PRODUCT RANGE VISIT
www.picotech.com/PS132

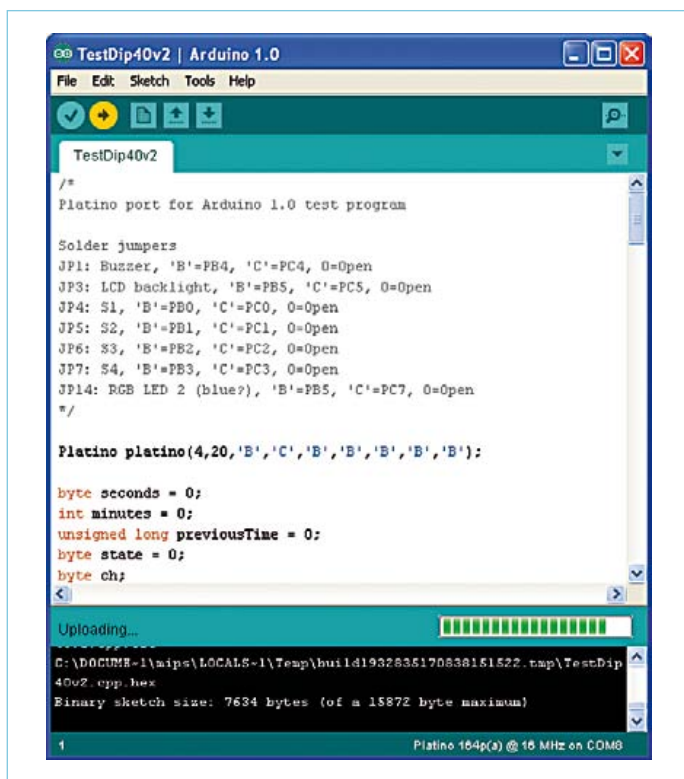


Figure 2. Platino within Arduino. The right-hand bottom corner shows which board is being targeted. The Platino object is invoked at the beginning of the sketch (in the centre of the screenshot).

have experimented with a few and came close to an almost ‘universal’ solution, but while testing I found that some of them still had issues requiring more attention than I could give at that time. In the free download that accompanies this article you will find a bootloader for (almost) every AVR device Platino can handle – use



Figure 3. The test sketch from Figure 2 running on Platino. The value in the lower l/h corner (302) is controlled by the rotary encoder at the right; the timer in the upper r/h corner can be reset with the pushbutton on the left.

them at your own risk and peril. They won’t break anything, but they might not program anything either. Currently I am using an ATmega164p with the included bootloader and that combination works great.

The first sketch you write (does it actually upload?) should help you test if the Arduino pin assignments are correct. Is digital input or output X really where you thought it would be? The same for the analogue I/O, plus you should also test Arduino’s SoftwareSerial library to see if the interrupts work as you intended. What about the function `attachInterrupt`? If all of this is okay you can be confident you did a decent porting job. At this point you’re good to go.

Using the Platino class

True, just invoking the Platino class will set you back some 2.5 KB of program memory, but it greatly lessens your programming effort. To invoke it simply write at the top of your sketch (figure 2):

```
Platino platino(4,20,'B','C','B','B','B','B','B');
```

meaning you are using an LCD with 4 lines of 20 characters, and that the solder jumpers 1, 3-7 & 14 are in the positions port B or C (indicated by a ‘B’ or a ‘C’; an open jumper can be indicated by a zero ‘0’). You can then access the Platino peripherals like so:

```
platino.led(state); // state can be 1 or 0.

if (platino.pushButton(1)==0) ...
    // check pushbutton 1 (active low)

if (platino.encoder2.tick(counter) !=0) ...
    // call often, 1 kHz if possible.

platino.beep(frequency,duration_in_ms);

platino.lcd.setCursor(0,1);
    // goto column 0 of row 1.

platino.lcd.print("the quick brown fox");
```

Pretty cool uh?

Line 3 may need some clarification. It calls the tick function of rotary encoder number 2, updating the value of `counter` at the same time. When this function returns non-zero it flags a change to the value of `counter`. As the comment indicates you have to call this function often or you may miss clicks.

Of course all the public functions of the LiquidLcd and other classes are available in your sketch as long as you prefix them with ‘platino dot’ (i.e. `platino.`).

The free download file specially compiled for this article (www.elektor.com/120094) also contains some test sketches that may help you familiarise yourself with the Platino class.

Have fun!

(120094)

“Elektor is substance for professionals and those wanting to reach the level. The perfect entry to my studies!”

– Christian, 19, freshman –



**20%
Discount!**

Electronics at all the right levels

Secure a head start in electronics with a Student Subscription!*

* Only available in UK. Ask for the conditions.

Advantages to subscribers

- Students qualify for a discount of no less than **20%** compared to the normal price of an annual subscription
- Subscribers get up to **40% discount** on special Elektor products
- No queues, traveling, parking fees or ‘sold out’ Elektor is supplied to your doorstep every month
- Always up to date – read your copy before everyone else



www.elektor.com/subs • Tel. +44 20 8261 4509

elektor

Electronics for Starters (3)

Transistor measurements

By Burkhard Kainka (Germany)

Electronic devices are becoming more and more complex, with simple circuits practically a thing of the past. This makes it increasingly difficult for beginners to get up to speed. In this series we therefore aim to get back to basics. In this third instalment we focus on various types of measurements on transistors. A simple analogue multimeter is all you need to measure quite a few transistor parameters.

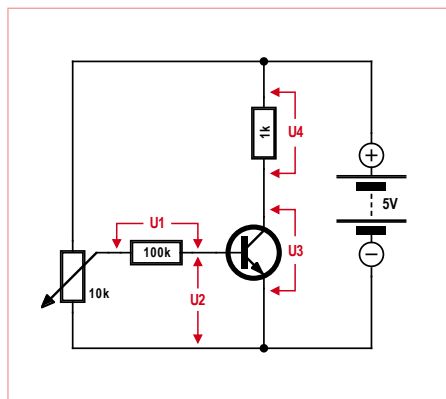


Figure 1. Measurement setup.

Table 1. Measured values for a BC547B transistor.							
	U_1	I_B	$U_2 (U_{BE})$	$U_3 (U_{CE})$	U_4	I_C	$A = I_C/I_B$
1	0 V	0 μ A	0 mV	5 V	0 V	0 mA	0
2	0 V	0 μ A	400 mV	5 V	0 V	0 mA	0
3	0.07 V	0.7 μ A	573 mV	4.9 V	0.1 V	0.1 mA	143
4	0.15 V	1.5 μ A	595 mV	4.8 V	0.2 V	0.2 mA	133
5	0.26 V	2.6 μ A	612 mV	4.6 V	0.4 V	0.4 mA	153
6	0.47 V	4.7 μ A	629 mV	4.2 V	0.8 V	0.8 mA	170
7	0.90 V	9.0 μ A	646 mV	3.4 V	1.6 V	1.6 mA	177
8	1.77 V	17.7 μ A	665 mV	1.8 V	3.2 V	3.2 mA	181
9	2.63 V	26.3 μ A	679 mV	0.3 V	4.7 V	4.7 mA	179
10	3.54 V	35.4 μ A	681 mV	0.15 V	4.85 V	4.85 mA	137
11	4.32 V	43.2 μ A	683 mV	0.13 V	4.87 V	4.87 mA	113

Manufacturers' data sheets for transistors typically include sets of characteristic curves that show how the transistors behave in different situations. However, it's even better to use your own instruments to measure as much as possible yourself. This gives you a feeling for the component and improves your understanding of how transistors work in practice — not just in theory.

You should measure the base current I_B , the collector current I_C , the base-emitter voltage U_{BE} and the collector-emitter voltage U_{CE} . If you wish to make all these measurements with a single multimeter, it's advisable to measure only voltages and to use the same measuring range as much as possible. The currents can be calculated relatively easily from the voltages and the resistances

in the circuit. **Figure 1** shows our measurement setup. The potentiometer connected to the circuit input allows the input voltage to be adjusted in small increments over the range from zero to 5 V. For each setting, you should measure and record the four voltages U_1 – U_4 . From this you can derive the currents and the amplification factor A .

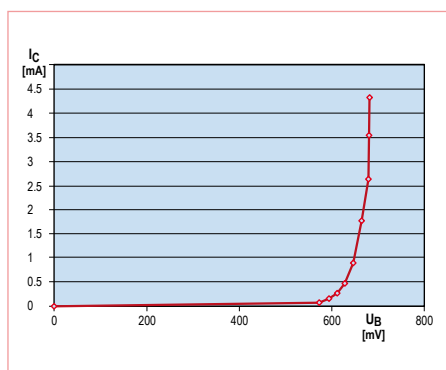


Figure 2. Base current versus base-emitter voltage.

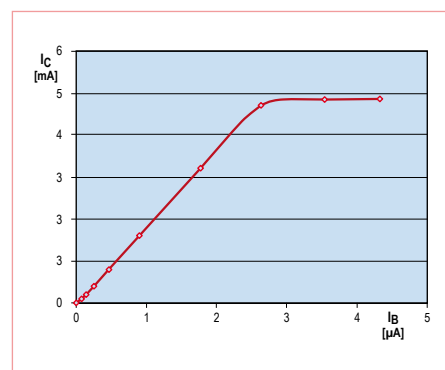


Figure 3. Collector current as a function of base current.

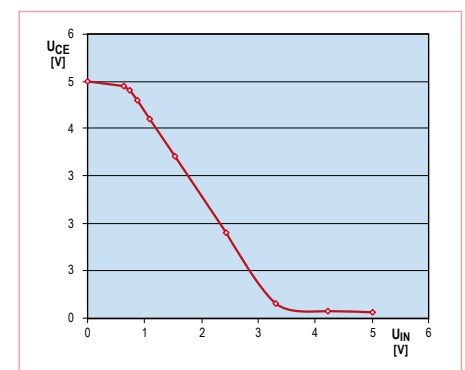
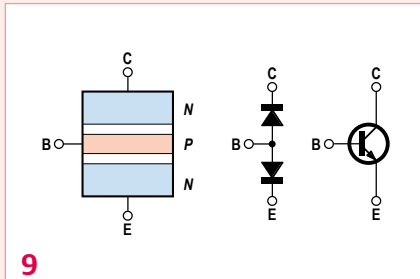


Figure 4. Output voltage as a function of input voltage.

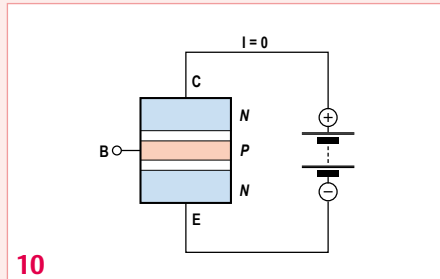
Basic transistor operation

Transistors are semiconductor components with three leads and are predominantly used as current amplifiers. Like diodes, transistors are made from p-doped and n-doped semiconductor materials, but they have three regions separated by junctions. These regions may be arranged as N-P-N or as P-N-P.

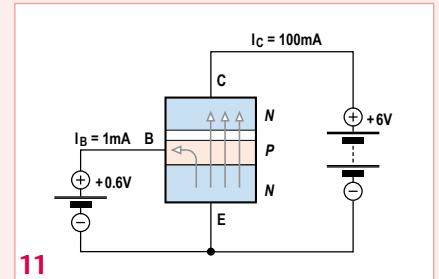
First let's consider the NPN transistor, whose structure and equivalent circuit diagram are shown in **Figure 9**. The individual regions of the



9



10



11

transistor are called the emitter (E), the base (B) and the collector (C). For proper operation the base region must be very thin. First consider the situation with the transistor connected to a voltage source, with the emitter connected to the negative terminal and the base lead unconnected (**Figure 10**). No current flows in this situation because the base–collector junction is reverse biased.

Now suppose a second voltage source is connected between the base and the emitter, with the positive terminal connected to the base and the voltage so low (approximately 0.6 V) that only a small current flows through the base–emitter junction. Here you will see a much larger current flowing between the emitter and the collector. This results from the fact that the base region is very thin. Negative charge carriers that enter the base region are exposed to a strong electric field across the collector–base junction, and most of them are drawn into the collector region. Only around 1 per cent of the charge carriers originating from the emitter reach the base lead (**Figure 11**). Put differently, this means that the collector current is around 100 times greater than the base current. The collector current is controlled by the base–emitter voltage or the base current.

Although the electrons travel from the emitter to the collector, due to the convention that current flows from the positive terminal to the negative terminal, here we say that the current flows from the collector to the emitter.

Table 1 shows an example of values measured with a BC547B transistor using separate, fixed multimeters for U_1 , U_2 and U_3 . U_4 was calculated from U_3 , and the currents and amplification factor were derived from the measured voltages.

Practical tips

What procedure should you follow for making the measurements? A good approach is to first adjust the collector current to 0.1 mA ($U_4 = 0.1$ V) and then double it repeatedly for each new measurement until it stops rising (set the value of U_4 successively to 0.1 V, 0.2 V, 0.4 V, 0.8 V, and so on). This reveals an interesting aspect of typical transistor behaviour: each time you double the collector current, the base current is approximately doubled as well, but the base–emitter voltage increases each time by a constant amount of around 20 mV.

You can calculate the current gain by dividing the collector current I_C by the base current I_B . As you can see from the table, the maximum amplification factor (current gain) was around 180 with our test transistor. This is a bit on the low side, since this

device should have a gain of at least 200. However, there are several potential error sources, such as the finite internal resistance of the meter (10 M Ω). Among other things, this means that a small portion of the intended base current flows through the meter instead of the transistor when U_2 is measured. Measurement errors are normal. If you take all the error sources and tolerances into account (including resistor tolerances), the gain of this transistor could actually lie right at the lower limit of 200. Try this for yourself – maybe your transistor can do better.

With such a large amount of measurement data, it's a good idea to analyse it in graphic form. You can do this with pencil and paper, or with a worksheet on your PC. This yields the following results:

Figure 2 (I_C versus U_B) shows the typical exponential curve of a silicon diode. On the linear scale you see a breakpoint at approximately 0.6 V. Above this level the slope of the current curve keeps rising (exponential curve). The measurement data shows that no measurable base current flows at a base voltage of (for example) 400 V, so there is

also no collector current. Accordingly, you can note that the base voltage is usually somewhere between 0.6 V and 0.7 V.

Figure 3 (I_C versus I_B) shows that the collector current increases linearly with the base current (in the first approximation) until it stops rising at just under 5 mA, which is called the saturation point of the transistor. In any case it isn't possible to measure more than 5 mA with this circuit because the collector resistor limits the current to 5 mA ($5 \text{ V} \div 1 \text{ k}\Omega = 5 \text{ mA}$). You can also clearly see that even this 5 mA limit cannot be reached. The transistor is driven fully into conduction ('hard on'), with a residual collector–emitter voltage of slightly less than 0.1 V.

This curve also has a lower slope (lower current gain) at very low current levels. This is actually true, because the gain decreases slightly at very low and very high collector current levels, but it also reflects a measurement error that amplifies this effect. A small current flows through the meter that measures U_2 , and at low current levels this makes the base current appear to be greater than it actually is.

Finally, **Figure 4** shows the output voltage (U_{CE}) as a function of the input voltage ($U_1 +$

U_2) on the wiper of the potentiometer. Here you can see right away that increasing the input voltage causes the output voltage to drop. The reason for this is simple: when the collector current rises, the voltage drop over the collector resistor increases.

Negative feedback

If you are designing a transistor circuit and you do not know the transistor gain precisely, you must adapt to this situation.

Things are easy with a switching stage: you only need to dimension the base current such that the circuit will work properly even with the lowest possible current gain. This means that in case of doubt you should use a bit more base current, to ensure that there's enough for every transistor of the chosen type.

Things are different when you need to amplify an analogue signal. Too much base current may be exactly the wrong thing in this situation, due to the risk of saturating the transistor. As much as possible you should aim for a mid-range collector current that can be adjusted upward or downward. One way to achieve this with a variety of transistors is to use negative feedback. This can be done by connecting the base resistor to the collector instead of the supply voltage (see **Figure 5**). A transistor with especially high gain tends to produce more voltage drop across the collector resistor, which reduces the collector voltage and with it the base current. At the other end of the scale, a transistor with low current gain automatically operates with more base current. The end result is that the circuit is suitable for a wide variety of transistors.

Making measurements with an ohmmeter

Pointer-type analogue meters still have certain advantages for component testing and troubleshooting. They allow results to be read faster than with digital multime-

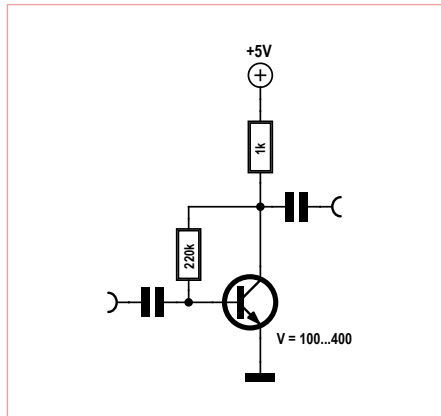


Figure 5. Using negative feedback to set the operating point.

ters, at least roughly. However, digital multimeters are indispensable when you need high precision.

Most simple analogue multimeters have one or more resistance ranges, and with a bit of practice you can use a simple ohmmeter to check not only resistors, but also transistors, diodes, capacitors and many other types of components. A multimeter needs a battery to measure resistance, although this battery is often not used for

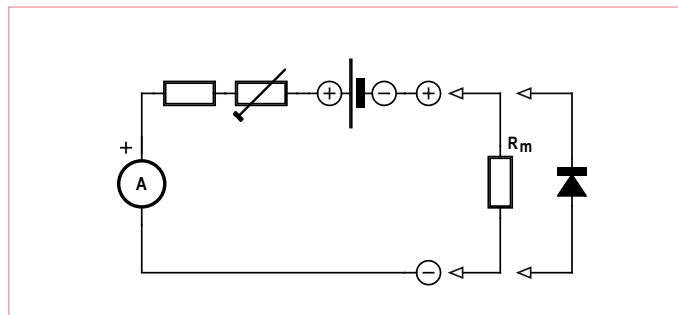


Figure 6. Basic circuit of an analogue ohmmeter.

any other purpose. Resistance is determined by measuring the amount of current that flows with a constant supply voltage. As a result, the resistance scale is not linear. The full-scale indication at zero ohms must be adjusted using a potentiometer in order to compensate for variations in the battery voltage (see **Figure 6**). The other end of the scale corresponds to infinite resistance, regardless of the range.

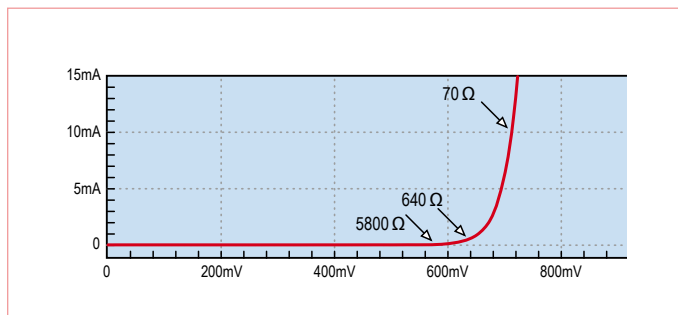


Figure 7. DC resistance of a silicon diode at various measurement currents.

A consequence of the usual circuit configuration used in simple analogue multimeters is that the polarity of the voltage on the terminals for the resistance ranges is opposite what you would expect from the markings for the current and voltage ranges. Consequently when a resistance range is selected, the negative terminal of the multimeter is positive and the positive terminal is negative. This must be taken into account if you want to use a simple ohmmeter to check diodes or transistors.

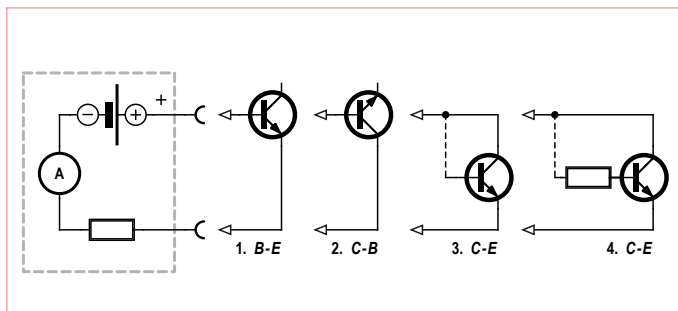


Figure 8. Transistor measurements.

When measuring diode junctions, you should bear in mind that it's impossible to specify a fixed resistance value for a semiconductor junction. The indicated value is strongly dependent on the measurement current, and therefore on the selected measuring range.

Transistor tester

A microcontroller with an internal A/D (analogue-to-digital) converter can be put to good use as a measuring instrument, such as (how did you guess?) a transistor tester. Here all we're interested in is measuring the current gain. An ATtiny13 microcontroller is sufficient for this task if the resulting measurement data is sent in serial form to a PC and displayed on the PC monitor using a terminal emulator program.

The circuit for this is very simple (see **Figure 12**). Only the collector voltage is measured. The transistor is operated with negative feedback, which allows a wide range of gain values to be measured. This requires a bit more computation from the program, but that's what microcontrollers are for.

```

`Transistor tester
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4

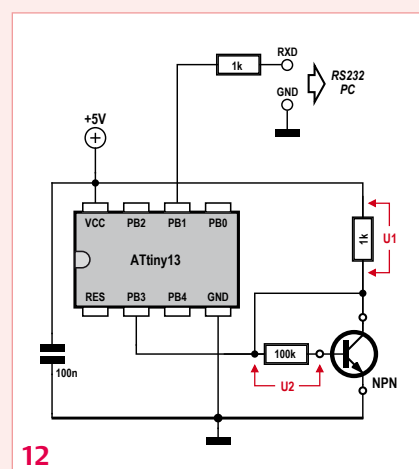
Dim UC As Word
Dim U1 As Word
Dim U2 As Word
Dim I1 As Word
Dim I2 As Word
Dim V As Word

Config Adc = Single , Prescaler = Auto
Start Adc

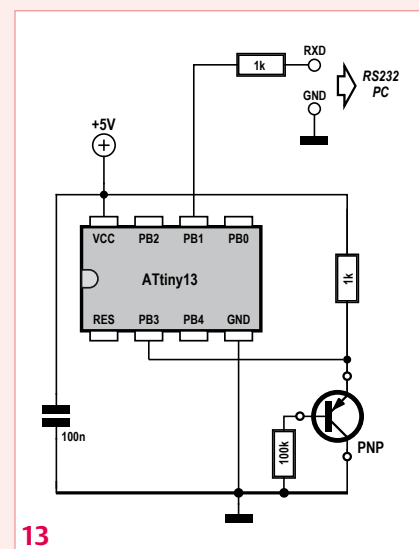
Open "comb.1:9600,8,n,1,INVERTED" For Output As #1

Do
  UC = Getadc(3)
  UC = UC * 50
  U2 = UC - 6000
  U1 = 51150 - UC
  I1 = U1
  I2 = U2 / 100
  V = I1 / I2
  Print #1 , V
  Waitms 1000
Loop

End
  
```



12



13

The program calculates the voltage drop U_1 over the collector resistor and the voltage drop U_2 over the base resistor. From these values it derives the collector current I_1 and the base current I_2 . The current gain A is then $I_1 \div I_2$.

To make all of this possible with a small microcontroller, the program uses only integer values of type word. We also took precautions to avoid numerical overflow and to avoid reduced accuracy due to small intermediate results.

The microcontroller can also make measurements on PNP transistors with the same program. This only requires a slightly different connection scheme, as shown in **Figure 13**.

Quiz

Can you determine the current gain with only one measurement, without the aid of a microcontroller? A transistor circuit with a 2.2-k Ω collector resistor and a 470-k Ω base resistor connected to the collector has an output voltage U_{CE} .

1) If you measure a value of 2.8 V for U_{CE} , what is the current gain?

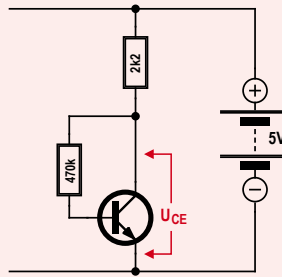
- A) Approximately 152
- B) Approximately 214
- C) Approximately 472

2) If you measure a value of 0 V for U_{CE} , what should you suspect?

- D) The transistor is shorted (second breakdown failure)
- E) The base resistor is not connected properly
- F) The resistance of the base resistor is 470 Ω instead of 470 k Ω

3) If you measure a value of 4.9 V for U_{CE} , what could be the problem?

- G) A broken collector lead
- H) A broken base lead
- I) The emitter and collector leads reversed



If you send us the correct answers, you have a chance of winning a **Minty Geek Electronics 101 Kit.**

Send your answer code (composed of a series of three letters corresponding to your selected answers) by e-mail to basics@elektor.com.

Please enter only the answer code in the Subject line of your email.

The deadline for sending answers is March 31, 2012.

The solution to the Quiz in the February 2012 edition will be published next month.

All decisions are final. Employees of the publishing companies forming part of the Elektor International Media group of companies and their family member are not eligible to participate.

Nevertheless, it is possible to draw meaningful conclusions. If the pointer deflection is around half of full scale with an internal voltage of 1.5 V, the voltage drop over the object being measured must be approximately 0.75 V. Due to the exponential shape of the diode characteristic curve, the pointer deflection changes only slightly when you switch to a different range. Although a different resistance is indicated in each measuring range, the pointer deflection is nearly

the same because the voltage drop is always approximately 0.6 V. The forward voltage of the diode can also be used to judge the diode type. For example, the diode in **Figure 7** is most likely a silicon diode.

Transistor testing

Using just a simple ohmmeter for transistor testing, you can draw several conclusions about the transistor type and its general condition. Even with a fully unknown

transistor type, you can at least determine which lead is which.

A transistor can be fully tested with the aid of three typical measurements. First you measure the base-emitter junction and the base-collector junction (see the 'Basic transistor operation' inset), which allows you to distinguish between silicon and germanium transistors and identify any short-circuits (**Figure 8**, parts 1 & 2). After this you measure the resistance between the emitter and the collector with and without base current (3). A transistor in good working order will show infinite resistance (no current) with the base open. With the base connected to the collector, the indicated current should be somewhat greater than the current through the base-emitter junction alone.

The final test (4) should be made with the a small base current flowing through a resistor connected between the collector and the base. The base current can be provided by touching the collector and base leads with a wet finger. The resulting deflection of the ohmmeter pointer gives a rough indication of the current gain of the transistor. A small current gain can be seen even with the emitter and collector leads reversed, so in case of doubt you should try reversing the transistor leads if the lead assignments are not known reliably.

Most digital voltmeters use an entirely different internal circuit configuration for resistance ranges. In this case the resistance measurement is based on measuring the voltage drop with a constant current. This results in a linear scale and a clearly defined measuring range, and it eliminates the need for adjusting the zero point. Another difference with respect to pointer instruments is that the voltage, current and resistance ranges have the same terminal polarity.

In theory, the resistance range of a digital multimeter can be used to make the same component tests as with an analogue meter. Many such meters also have a separate measuring range for diode junctions. Although this is based on the same measuring principle as the resistance range, it displays the voltage drop in millivolts or a reading that is proportional to the forward voltage.

(120003-1)

Portable and Powerful

Discover the World's Most Trusted USB Data Acquisition Devices

*USB M Series
Starting at £499*

*NI CompactDAQ System
Starting at £749*



*NI USB-9000 Series
Starting at £199*

*Low-Cost USB DAQ
Starting at £69*

*NI USB-TC01
Starting at £89*

SEVERAL SOFTWARE CHOICES

NI LabVIEW

NI LabVIEW SignalExpress

ANSI C/C++

C#

Visual Basic .NET

Visual Basic 6.0

National Instruments offers more than 40 USB devices for high-quality measurements, and each device is shipped with free NI LabVIEW SignalExpress LE software. These industry-leading devices use the latest hardware and software innovations to deliver true high-performance measurements while offering the simplicity and portability of USB. With up to 1.25 MS/s sampling rates, 24-bit resolution and 256 channels, NI USB devices fit a full range of application needs from low-cost to high-performance modular systems, all with world-class technical support.

>> Find the right NI USB device for your application at ni.com/daq/usb

01635 517300
uk.ni.com
info.uk@ni.com

Optimise your RF Measurements: 7-part Webcast Series
uk.ni.com/webcasts/rfmeasurements

JOIN THE CONVERSATION: Follow us on



Search **niukie**

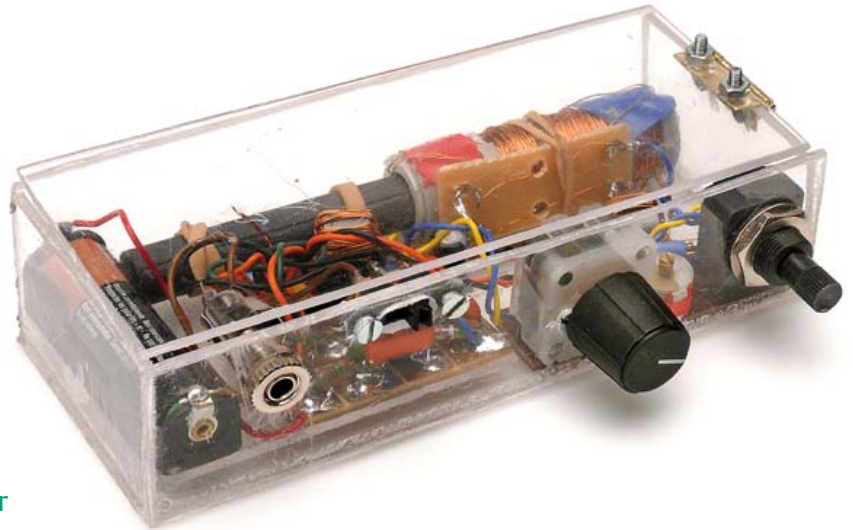


Economic Longwave Receiver

By Jean-Pierre Redouté (France)

Even today in 2012, radio is still one of the most dramatic ways to enter the world of electronics. Long Wave (LW) still offers an experimental ground with enough stations to satisfy all tastes: BBC Radio 4, France Inter, Europe 1, RTL and other even more exotic ones, depending on where you are in the world. And the magic of a receiver you've built yourself is all the greater when the

technical side matches up to the rest: use of modern earpieces, powering from a single 1.5 V cell, easy to build, no integrated circuits, no ferric chloride, and very few holes to drill. Now here's the perfect circuit to learn about and reinvent the pleasure of electronics.,



Often, radio receivers intended for home building are powered from a 9 V battery, most of which voltage is dissipated in heat by the collector load resistors. The earpieces are high-impedance types that are virtually impossible to get hold of. Worst of all, they only receive shortwave, where generally there's not much left to listen to except if you know Chinese or Eastern European tongues. The circuit described in this article, however, is powered from a single AAA (LR3) cell and draws approximately 1.5 mA, which allows a lot of listening time on the long wave band without an external aerial on a headset.

Easy to build, it uses only around twenty discrete components mounted on two copper-clad boards simply 'engraved' with a few saw-cuts (no PCB layout, no ferric chloride), and most of the components are fitted on the copper side.

Operating principle

The basis of the receiver described here is a standard: two cascaded transistors in reflex mode, first described almost 50 years ago by Sir Douglas Hall in *The Radio Constructor*

magazine in April 1964 using germanium transistors, then again in November 1968 using silicon transistors. A simplified version presented in January 1968 by G. Short was the starting point for countless variants that have appeared in various magazines all over the world as well as on the web. One of these, by R. Haig in *Everyday Practical Electronics* in July 2003, forms the basis of the band-stop filter and level adjustment used here.

The inherent drawback with these cascaded junctions is that their threshold voltages add together, leading to a base voltage on T2 of over 1.2 V, meaning it's impossible to power it from a single 1.5 V cell unless it is brand new.

The particular feature of our version here is the addition of a third transistor (PNP) to offset the bias to T2, while maintaining the automatic stabilisation of the original circuit.

Block diagram

The circuit diagram of the receiver is shown in **Figure 1**. The circuit combines three operating modes:

1. RF

The radio signal is picked up by a ferrite rod aerial, with the frequency tuned by L1 and C6. The coupling winding L2 matches the tuning circuit to the T1's low input impedance; its cold end is grounded via C1 (whose impedance is negligible at RF) and its hot end may or may not go through an optional band-stop filter. A proportion of the signal, defined by P1, is applied to the base of T1 for amplification (emitter grounded, collector loaded by R1), then by T2 (emitter grounded, collector loaded by L3, collector grounded, as C3 and C2 have negligible impedance at RF). The amplified signal is passed by T3 (collector grounded, emitter loaded by R3) then C5 to the demodulation network formed by D1 and C1.

The band-stop filter tuned by L5 and C8 is optional, its purpose is to attenuate or block a nearby and/or powerful transmitter that might hamper reception due to the poor receiver selectivity of the single tuned circuit on the ferrite rod.

One part of the RF signal amplified by T1 is re-injected in phase into the input tuning circuit via C7; the level of the positive

feedback ('reaction') thus obtained is set by potentiometer P1.

2. Baseband

The audio signal available at C1 passes through L2 and L4 (the tuned circuit and band-stop network have negligible impedance at baseband frequencies) and undergoes adjustable attenuation via P1. It is then amplified by T1 (emitter grounded, collector loaded by R1) then T2 (collector loaded by the earpieces, emitter grounded, as L3 and C2 have very low impedance at baseband).

3. DC (biasing)

Since T1's emitter is grounded, its base is at a voltage of around 0.6 V. Hence P2 wiper is at this voltage of 0.6 V plus the voltage drop across diode D1. T3 emitter is at a voltage at or above that of the wiper, as defined by the setting of P2. T3's base voltage is around 0.6 V lower than its emitter voltage, and hence can be adjusted by P2, and can go down as low as the value of the diode voltage drop. This voltage, divided by R2, defines the quiescent current in T2. Moreover, by adding around 0.6 V we get the base voltage for T2, which in this way can be set to a low enough value to allow T1 to operate even with a battery that's quite old.

Construction

Let's start with the hardest bit: the tuning circuit. Variable capacitors have become almost impossible to find in electronic component shops, at least at reasonable prices, which leaves us with two solutions: salvage one from an old receiver, or buy one by mail order. The ferrite rod is hardly easier to find. You can order one from Conrad Electronics (8 mm diameter, 50 mm length, part number 53-55-75).

To improve sensitivity, our circuit uses a longer rod (100 mm) from Rapid Electronic Components (part no. 88-3098), along with the variable capacitor (part no. 12-0255) and while we're at it, the 34 SWG enamelled copper wire.

One tip for anyone wanting to cut down a salvaged rod and achieve a clean cut at the right length: this material breaks like china, but it's impossible to saw or file. How-

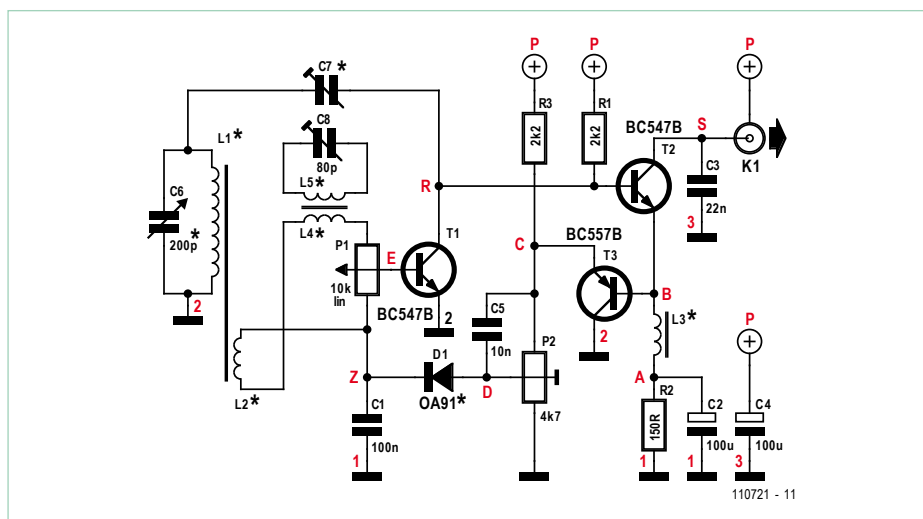


Figure 1. Block diagram. The red characters correspond to the engraving plan in Figure 2.

ever, with a little care, it is possible to cut a slight groove around the rod using a sawblade, then grasping the rod firmly either side between thumb and index finger, you can snap it, like breaking off a square of chocolate.

Don't use a new blade, as this operation will blunt it and make it unusable. Be patient and proceed carefully: often, the part of the ferrite rod clamped in the vice will break into several pieces while you're making the groove, while the section outside remains intact.

Windings L1 and L2 are pile-wound on a 14 × 1.6 mm piece of CPVC tube. To reduce its stray capacitance, L1 is divided into three windings of 90 turns each (the inductance obtained is around 5.5 mH). L2 is a single winding of 50 turns. The windings are separated by small rings cut from 16 mm diameter PVC tube, split so that they can be fitted onto the CPVC former (Figure 3).

The 34 SWG (i.e. around 0.24 mm) wire used is in no way obligatory, but just a practical compromise. Do note that the number of turns depends on the capacitance of C6 and the A_L coefficient of the ferrite rod; hence the values given here will need to be reviewed if you use components different from the 100 mm rod and 200 pF variable capacitor shown in the circuit.

The coils are connected to the rest of the circuit via a small piece of tagboard, but a small piece of copper-clad board divided into four squares would do just as well. The whole thing is held in place just using elastic bands.

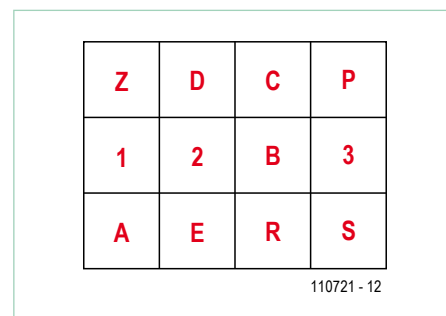


Figure 2. The engraving plan.



Figure 3. Inductors L1 and L2 are wound on a ferrite rod with a diameter of 10 mm and a length of 100 mm.



Figure 4. This close-up shows how to solder the components onto the board.

Table 1. Voltages measured with respect to 0 V on the squares of the prototype board.		
Square	Voltage	Notes
P	1.315 V	a battery that's see extensive use
A=B	0.15 V	
C	0.745 V	
D=	0.64 V	
Z=E	0.6 V	
R	0.79 V	
S	1.23 V	two 42 Ω (measured value) earpieces in series



Figure 5. C7 is formed by two wires twisted together – a 'pigtail' capacitor.

The RF blocking inductor L3 consists of 85 turns of 34 SWG enamelled copper wire on a 9 mm ferrite bead (part no. 50-79-62 from Conrad); this gives us around 3.2 mH. The band-stop filter coil L5 consists of 40 turns of the same wire on the same type of tube (around 750 μH); the coupling winding L4 has 7 turns. In conjunction with a 6–80 pF trimmer capacitor (C8), and allowing for the stray capacitance, this covers the LW range.

C6 (both gangs in parallel) and P1 (along with the band-stop filter, if fitted) are assembled onto a 50 × 15 mm piece of single-sided copper-clad board. To avoid the need to lay out a PCB, the holes for the pins and wires are drilled, and the component positions checked. Saw-cuts are then used to divide the copper into parallel strips (pre-drilled). It is then possible to assemble and solder up this little module.

The other components are soldered onto the copper side of a piece of single-sided copper-clad board, divided by five saw-cuts into 12 squares 10 × 10 mm (a method borrowed from Brazilian radio ham PY2OHH) – see **Figures 2** and **4**. The component leads must be as short as possible, and if there is any risk of a short, you should feed them through a scrap of insulation stripped from a piece of wire.

Trimmer capacitor C7 in actual fact consist of two pieces of enamelled copper wire twisted together, adjusted by trimming them to length (1 cm corresponds to approximately 2 pF). It is can be seen between the board and the ferrite rod (**Figure 5**).

The preset potentiometer P2 can be fitted

temporarily, then replaced after setting by two fixed resistors.

This module is fixed using a bolt through square 3.

Just one small regret: it would have been more elegant to use double-sided copper-

clad board (with the underside as a ground plane). In point of fact, the 0 V rail here uses three squares (nos. 1, 2, & 3) that have to be joined by wires, instead of three through-hole links soldered on each side if double-sided board had been used. But as the receiver works OK as it is, this modification has not been carried out.

In order to present an impedance of 64 Ω, the headset's two earpieces must be con-

nected in series. If you want to keep the original cable, the stereo jack socket must be wired just to the two rings, i.e. without using the common.

Figure 6 shows the author's finished, working prototype.

Setting up

Start by checking the wiring, polarity, and that there are no (sources of) shorts. Then, without connecting the headset, connect up the battery with a milliammeter in series; the current drawn should be less than 1 mA.

Then, with the battery connected normally, connect the milliammeter in place of the headset (or in series with it) and adjust P2 to obtain 1 mA in T2 collector.

If you wish, it is then possible to replace P2 with a fixed potential divider with the same ratio and similar total resistance (this setting is not very critical).

As a guide, **Table 1** shows the voltages measured with respect to 0 V on the squares of the prototype board.

For the rest, it's best to position all the elements in their final positions, as the stray capacitances and inductances will vary

according to the respective positions. If we start deliberately with too high a value for C7 (very long twisted wires, e.g. 10–15 cm), it should be possible to cause T1 to oscillate by increasing the setting of P1; its frequency is defined by C6 and L1, and can be detected using a LW receiver placed close alongside. To check our circuit's frequency setting, all we need do is set this check receiver towards the bottom end of the LW band and very slowly adjust C6. As you tune

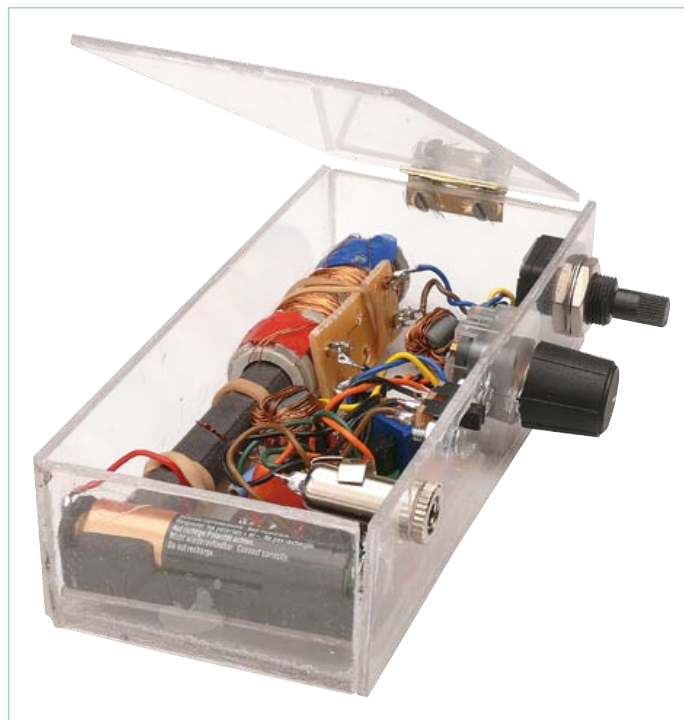


Figure 6. The receiver fitted into a home-made Perspex® case.

COMPONENT LIST

Resistors

R1, R3 = 2.2k Ω
 R2 = 150 Ω
 P1 = 10k Ω linear potentiometer
 P2 = 4.7k Ω trimpot

Capacitors

C1, C5 = 10nF
 C2, C4 = 100 μ F electrolytic
 C3 = 22nF
 C6 = 200pF tuning capacitor (see text)
 C7 = twisted wire (pigtail) capacitor (see text)
 C8 = =80pF trimmer

Inductors

L1–L5 = see text

Semiconductors

D1 = germanium diode like OA91, AA112,
 AA119, etc.
 T1, T2 = BC547B
 T3 = BC557B

Miscellaneous

K1 = 3.5 mm stereo jack socket

through the frequency the receiver is tuned to, you'll hear a characteristic 'chirp'. Repeat the operation near the top of the band. If this result is obtained around 160 kHz, but not 280 kHz, you'll need to remove a few turns from L1 or even disconnect the "oscillator" gang of C6. In the reverse situation, you'll need to add a few turns or fit a small fixed capacitor in parallel with C6.

If the circuit is fitted with the optional band-stop filter: plug in the headset, turn the power on, set P1 below the oscillation point (if necessary, shorten C7 a little to increase the adjustment range of P1 so as to obtain an adequate level) and listen to the interfering transmission while adjusting C6 and the orientation of the ferrite rod for maximum sound level. Then slowly adjust C8 for

maximum attenuation (use a plastic or brass screwdriver or trimmer tool if possible, otherwise the presence of the metal blade will influence the adjustment). Lastly, adjust C7 (using cutters, a millimetre at a time, and **taking care not to short across the two wires!**) in such a way as to allow the maximum setting of P1 over the whole of the LW band without going into oscillation. Don't forget that a receiver with a ferrite rod aerial is very sensitive to its orientation: the

ferrite rod must be horizontal and at right angles to the direction of the transmitter. Although designed for use with a walkman-ish headset (connected in series, 64 Ω), this receiver also works with the headsets provided on long-haul flights (connected in series, 600 Ω).

Happy listening!

(110721)

See your project in print!

Elektor magazine is looking for Technical Authors/Design Engineers

If you have

- ✓ an innovative or original project you'd like to share with Elektor's 140 k+ readership and the electronics community
- ✓ above average skills in designing electronic circuits
- ✓ experience in writing electronics-related software
- ✓ basic skills in complementing your hardware or software with explanatory text
- ✓ a PC, email and Internet access for efficient communications with Elektor's centrally located team of editors and technicians

then don't hesitate to contact us for exciting opportunities to get your project or feature article published.

Our Author Guidelines are at: www.elektor.com/authors.

Elektor

Jan Buiting MA, Editor

Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, United Kingdom

Email: editor@elektor.com

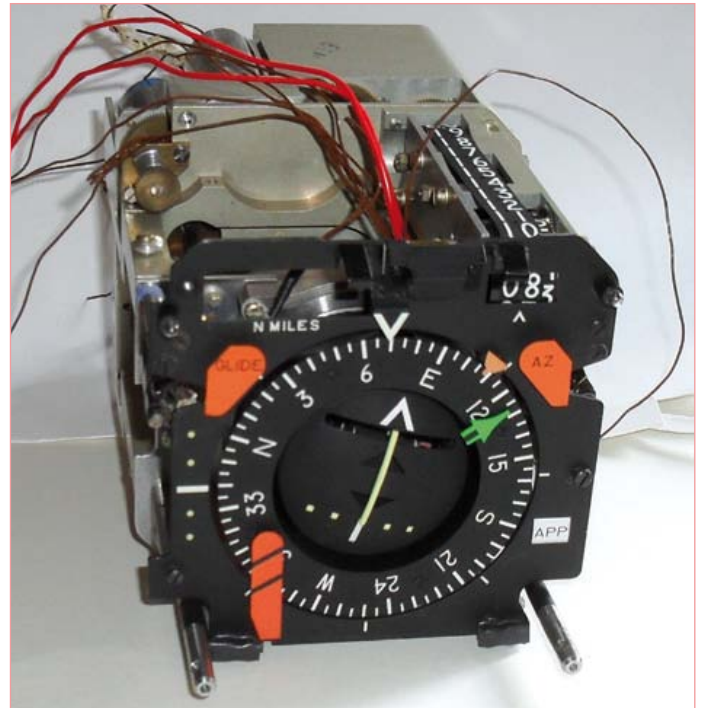


Ground Control for Aircraft HSI

Repurposing avionic instruments

By Martin Ossmann (Germany)

Although the author is an electronics specialist he is also fascinated by electromechanical contraptions of every description. When he saw this old aircraft instrument on offer from a local surplus trader he couldn't resist. Armed with a set of screwdrivers and some novel circuit solutions he set about coaxing this mechatronic masterpiece back to life.



A local supplier of surplus electronic instruments [1] had already caught the attention of the Elektor team and was the subject of a Retronics instalment in Elektor [2]. The surplus piece of kit in question here (Figure 1) is a 'Horizontal Situation Indicator' (HSI) [3] which in some previous life graced an aircraft's instrument panel. This combination instrument includes indicators for the VHF Omnidirectional Range (VOR) radio navigation system, the glide path (ILS) and the Heading Indicator (HI).

With such an impressive bit of kit lying on the workbench thoughts turn to how easy it would be to design a controller to exercise all the different dials and indicators on the HSI. An obvious use for such an instrument together with an external controller would be in a flight simulator cockpit but with appropriate software it could also be turned into a novelty 'HSI clock' for the office wall or club house. The author has posted short video clips on YouTube showing his HSI controller at work [4].

Hack1: Synchros and Motors

A view around the back of the instrument (Figure 1) reveals the impressive collection of electro-magnetic devices used in the instrument. Some parts have a blue end cover and some have a plain metal cap. The blue components come from the aerospace and avionics component manufacturer Muirhead [5]; these devices have three wires for the primary side and two for the secondary side. It turns out that the units function as synchros [5] or resolvers. A good description of how these components work can be found at

[6] and [7]. The main job of a synchro is to receive angular rotation information and convert it to an indicator needle position on a display device. This angular information can be supplied by a directional gyro for example. The gyro is fitted with a Synchro-Control-Transmitter (CT) to measure the rotation of a gimbal shaft. The CX has three secondary stator windings offset by 120°. The rotor car-

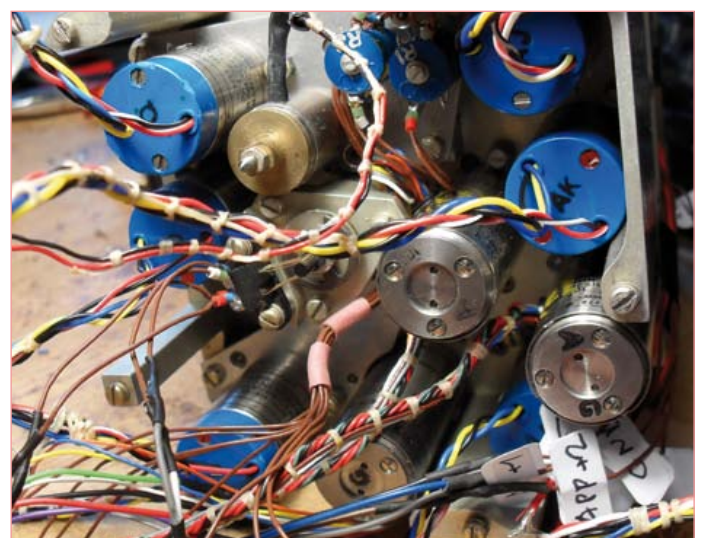


Figure 1. A peek around the back showing the motors and synchros

ries the primary excitation winding and the signal it induces in each of the three secondary windings is dependant on the rotor angle θ (Figure 2 left). The primary excitation signal is a 400 Hz sine wave $U_R(t) = \hat{U} \sin(2\pi t 400 \text{ Hz})$.

The signals induced in the three secondary windings are given by:

$$U_{S1}(t) = \hat{U} \sin(\theta) \sin(2\pi t 400 \text{ Hz})$$

$$U_{S2}(t) = \hat{U} \sin(\theta + 120^\circ) \sin(2\pi t 400 \text{ Hz})$$

$$U_{S3}(t) = \hat{U} \sin(\theta - 120^\circ) \sin(2\pi t 400 \text{ Hz})$$

These three signals give the rotor's angle information; with three wires we can pass this positional information to move a needle on the display. The circuit symbol for a CX is shown on the right in Figure 2. To convert this electrical angular information into a mechanical angle a control loop is formed (Figure 3).

The electrical angle information is connected to the so-called Synchro-Control-Transformer (CT) in the HSI. The primary side (stator) of the CT consists (like the secondary side of the CX) of three windings set 120° apart. The rotor is the secondary winding.

The amplitude of the 400 Hz signal produced at the secondary side of the CT is equal to $\sin(A - B)$ where A is the angle of the CX shaft and B is the angle of the CT shaft. This voltage is zero when both angles are the same. The secondary voltage is used as an error signal to drive the motor and adjust angle B until $A = B$. This principle is used to control all the rotating display needles of the HSI.

The alternative

To implement the original control methodology requires a generator to produce the three electrical signals to mimic the CX signals and a feedback control system for each of the display pointers. For our purposes there is however a simpler method using less hardware.

The CT windings can be powered as shown in Figure 4 by two 400 Hz phase-shifted sine waves. The degree of phase shift corresponds exactly to the angular rotation of the output shaft. An A/D converter can then be employed to measure the secondary signal from the CT to determine the phase and rotation angle. All of the CTs can use the same fundamental sine wave signal so it is only necessary to generate it once.

PWM signal generation

A 400 Hz sine wave can be generated from a PWM output of a microcontroller. It needs a processor clock of 20 MHz and a fast PWM with 8-bit resolution. The PWM frequency works out at $20 \text{ MHz} / 256 = 78.125 \text{ kHz}$, which is relatively fast when compared with the required 400 Hz sine wave. An interrupt frequency of $36 \times 400 \text{ Hz}$ has been selected by dividing 20 MHz by 1389. A new PWM value for the sine or cosine wave is sent at each interrupt. The

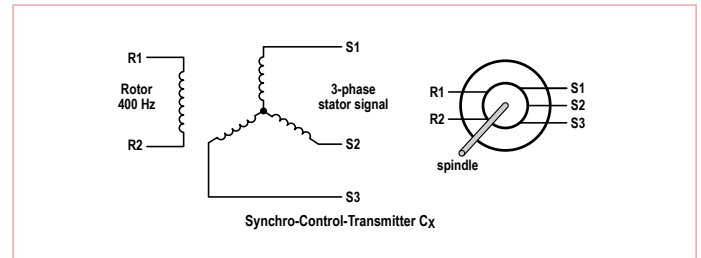


Figure 2. Diagram of a Synchro-Transmitters (CX).

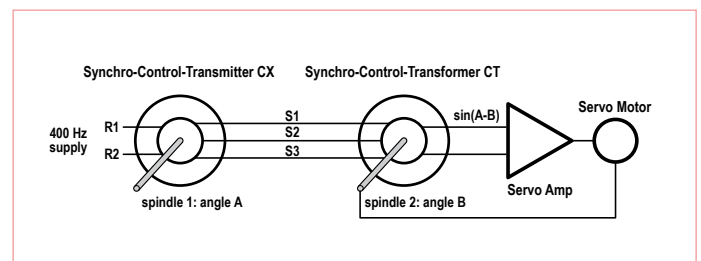


Figure 3. An electrical 'shaft extender'.

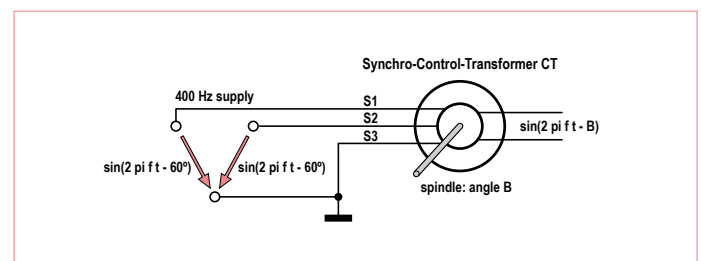


Figure 4. A simpler CT motor controller.

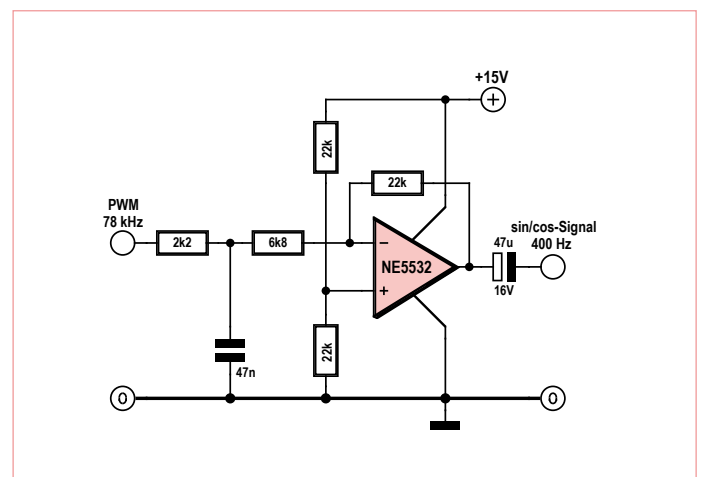


Figure 5. Low-pass filter and amplifier for the 400 Hz sine wave.

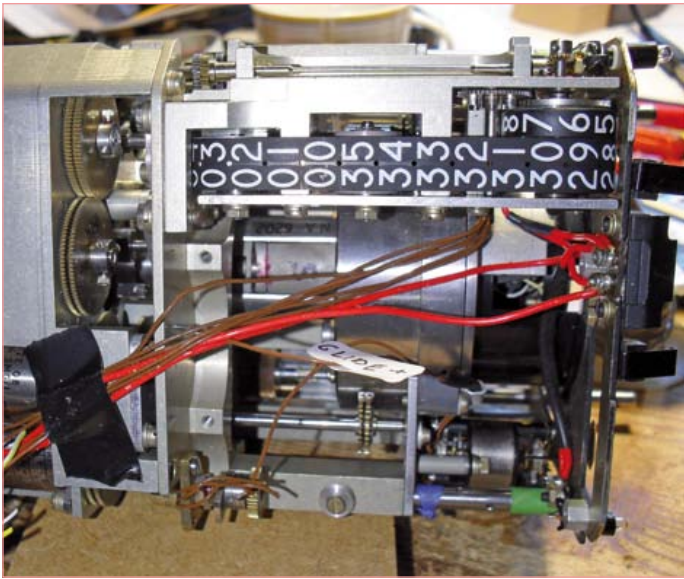


Figure 6. The unit from above.

resulting PWM signal is then low-pass filtered and amplified before it drives the CT windings. The circuit for one of the two channels is shown in Figure 5.

Phase measurements

In order to determine the mechanical angle of the CT shaft it is necessary to find the phase of the rotor signals. The signal is sampled during the same interrupt routine which services the PWM generation so this equates to $400 \times 36 \text{ Hz} = 14400 \text{ Samples/s}$. One period of the fundamental will consist of 36 measurements. The phase can then be determined from these 36 values in the same way that a Software Defined Radio (SDR) achieves phase demodulation. The signal is decomposed into its sine and cosine parts and from the quadrature components it is possible to determine angle information.

Using this principle we can determine the angle of the various HSI display pointers and use this information in a control loop to achieve the desired position.

Get your motor running

Now with the control solved the next task is to design the interface to the motors. One of the motors has 10 wires and as far as

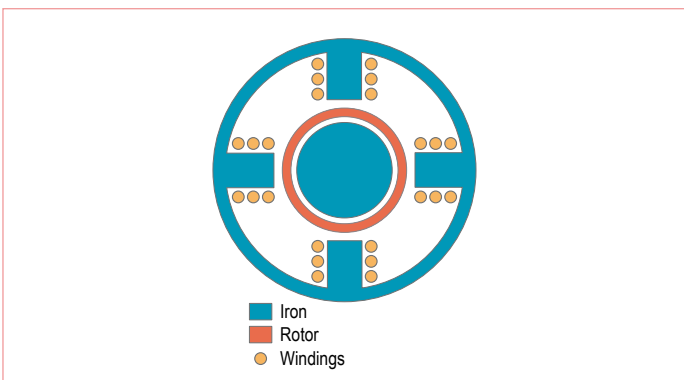


Figure 7. Cross-section of a drag-cup motor.

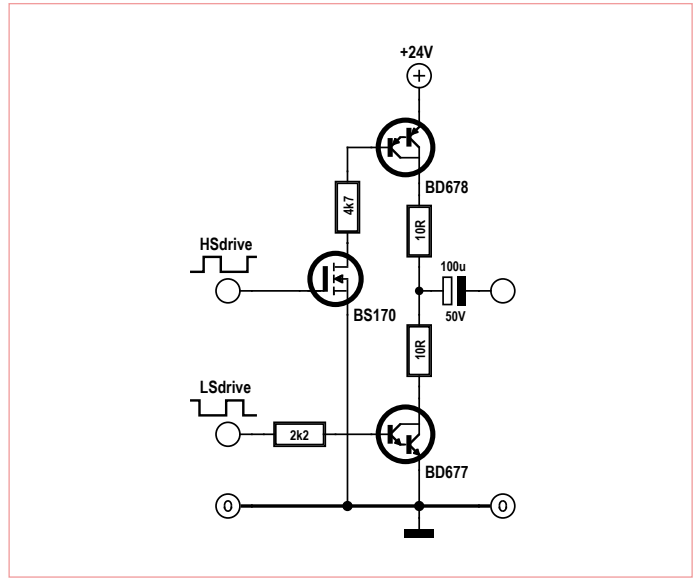


Figure 9. 24 V half bridge.

can be made out from the labelling four of these are connected to the motor’s tachogenerator. In addition there are three windings, one of which is labelled ‘REF 26V 400c/s’. This is connected to the 26 V/400 Hz signal. After much fruitless searching we eventually managed to track down a relevant documentation for the motor [8]. The motor is apparently a drag-cup design [9] with a tachogenerator. The rotor is built like that of an induction motor but has a stationary magnetic core. The stator consists of a so-called reference winding, permanently energised by a 400 Hz signal. A second winding is positioned at right angles to the first. This winding is powered by a 400 Hz signal phase-shifted from the reference winding by 90° which generates rotor rotation. A change in polarity changes the direction of rotation. The winding is made up of two sections to simplify the control loop connection.

A diagram of the motor cross section is shown in Figure 7. The bell shaped rotor rotates around an inner iron core. Around the outside of the rotor there are two windings which, like a two-phase induction motor, generate a rotating field [10].

During some experimental bench testing we found that the motor was not fussy about the purity of the sine wave drive, in fact is also



Figure 8. One of the motor/tachogenerators units.

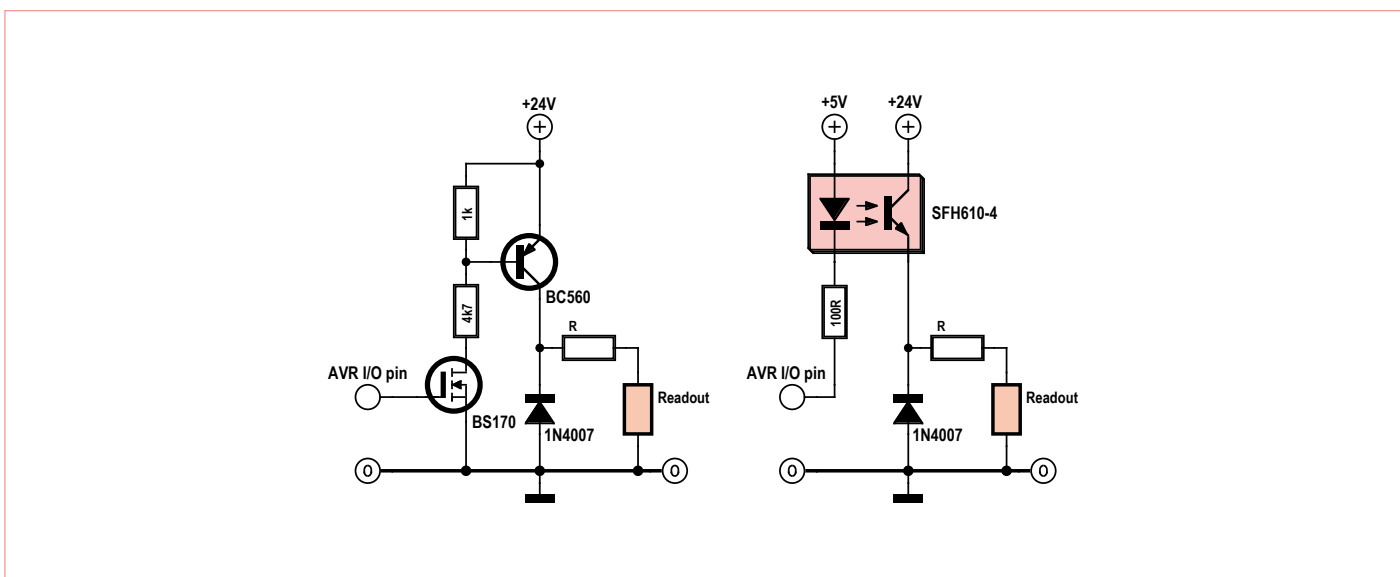


Figure 12. 24 V driver.

functions when driven by a square wave. With this in mind the half-bridge circuit shown in **Figure 8** was built. The drive signals are produced by the controller, the low-side and high-side drive signals are separated by a dead band to ensure that the two output transistors are never conducting simultaneously.

Two of these half-bridge circuits are used to control one motor as shown in **Figure 10**. The rotational speed is controlled by their relative phase shift.

Tachogenerator

Another interesting feature of the instrument is the construction of the tachogenerator. It consists of an excitation winding driven by a 400 Hz sine wave. Placed at 90° to this winding is the sensor winding which because of its orientation, picks up no signal from the excitation winding while the rotor is not turning. The rotor armature consists of short-circuited windings (i.e. squirrel cage) just like the motor part shown in **Figure 7**. When the rotor starts to turn it distorts the field so that the sense winding now starts to pick up the excitation field signal. The voltage induced in the sense winding is proportional to the rotor speed. This type of brushless design is both reliable and robust.

Course indicator CDI

The Course Deviation Indicator (CDI) is a pointer which moves left or right, indicating course deviation. Information from a VHF Omni-directional Range (VOR) navigation system is used to calculate the displayed deviation value. The CDI mechanism is based on a centre-zero moving-coil analogue movement.

The circuit given in **Figure 11** will generate the necessary positive and negative current to move the needle left or right for our purposes. Using a PWM signal with a 50:50 mark-space ratio puts the bridge into balance so that the indicator needle remains in the middle. By altering the duty cycle to greater or less than 50:50 moves the CDI to the left or right. The preset allows for sensitivity adjustment.

TO/FROM indicator

The TO/FROM arrowheads of the VORs are driven by a moving-coil type analogue mechanism but can be driven by a digital signal. The circuit from **Figure 11** is used but a tri-state output can be used instead of a PWM signal. With the output in high-impedance mode no current will be flowing so both the TO arrow and FROM arrow are hidden. Drive the output to a '0' or '1' and the TO arrow or FROM arrow respectively will be displayed.

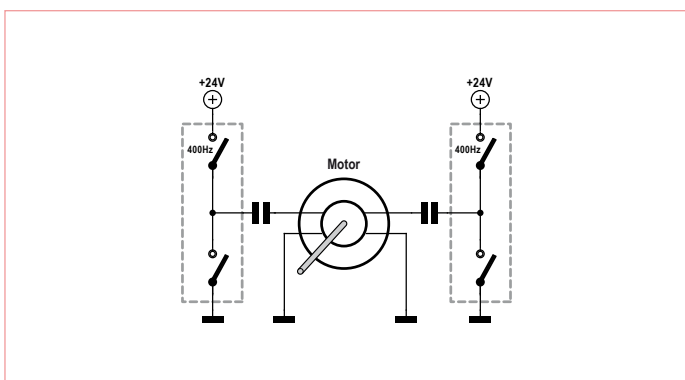


Figure 10. Two-phase motor control.

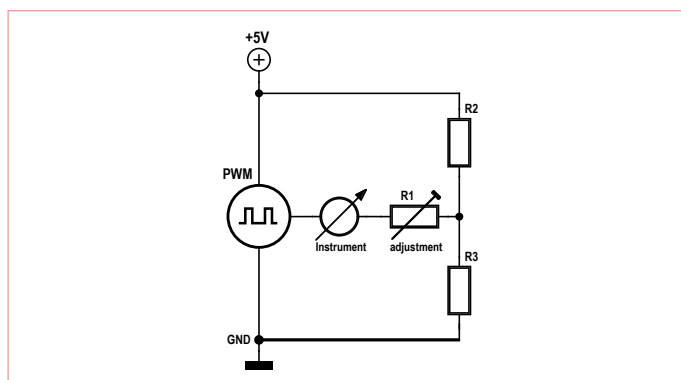


Figure 11. Bridge to control the CDI.

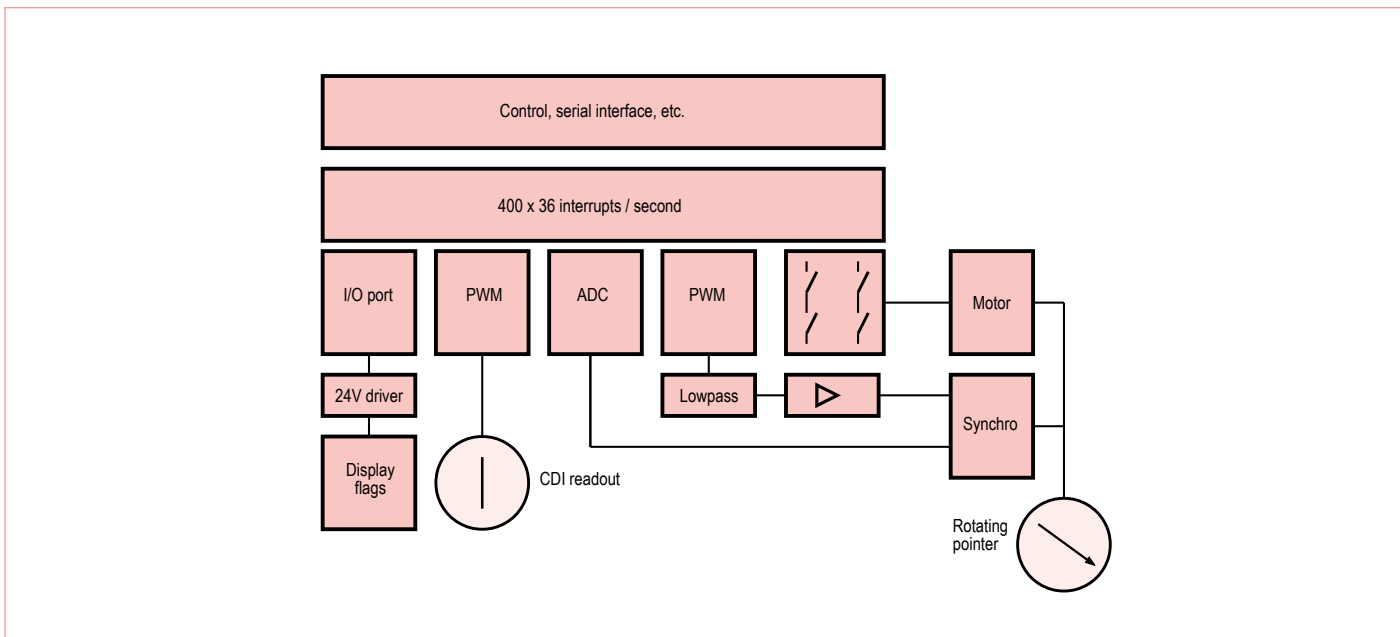


Figure 13. System diagram.

Wave those flags

The HSI also has a few orange indicator flags that are normally retracted behind the instrument’s mounting bezel. These can be flipped into the dial viewing area by applying a voltage to their actuating electro magnets. Several of these have a common earth connection and require a voltage greater than 12 V for operation. The driver circuit shown in **Figure 12** has been developed for this purpose.

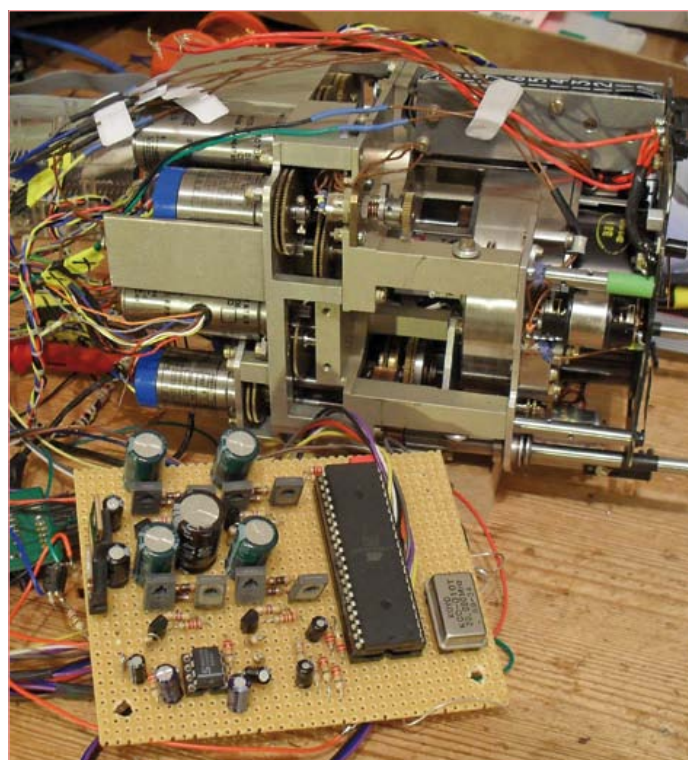


Figure 14. Prototype PCB.

For low current actuators the high gain optocoupler circuit should be sufficient but for higher current applications the circuit using a BS170 to level shift together with a BC560 to switch the current will be more appropriate. The value of resistor R in series with the load can be selected to limit the output current.

Chocks away

With the help of all the circuits described above it is possible to remotely control all of the HSI functions. The complete control concept using an ATmega644 microcontroller is shown in **Figure 13**. This controller can run at 20 MHz and has all the necessary timers and A/D converters on board.

A prototype controller board (**Figure 15**) has been built on a small square of perfboard. It can be seen driving the HSI in the You Tube video mentioned above. Now what’s stopping you kitting out a flight simulator cockpit with real instruments and taking your flying skills up to the next level?

(110756)

Internet Links

- [1] www.helmut-singer.de
- [2] www.elektor.com/090287
- [3] http://en.wikipedia.org/wiki/Horizontal_situation_indicator
- [4] www.youtube.com/user/ossimodding
- [5] www.muirheadaerospace.com/motion-technology/synchros.html
- [6] <http://en.wikipedia.org/wiki/Synchro>
- [7] www.ddc-web.com/documents/synhdbk.pdf
- [8] www.hnsa.org/doc/neets/mod15.pdf
- [9] www.google.com/patents (US-Patent 3641376)
- [10] A Textbook of Electrical Machines by R. K. Rajput, page A-5

Low Cost DMX Mixer Desk

Colour control for LED spotlights



Figure 1. The PAR56 RGB LED spotlight used by the author is fitted with 153 LEDs. The bigger PAR64 RGB LED spotlight is also very popular and uses 183 LEDs.

By Ingo Sundermann (Germany)

Disappointed that he could not find a simple cheap DMX remote controller for a multi-coloured LED spotlight the author set about designing his own. Armed with an ATmega8 and a few slide pots this basic, low-cost mixer was born.

LED spotlights for stage lighting or house parties are now quite reasonably priced. Most of them are described as PAR (Parabolic Aluminized Reflector) 'spots' conforming to the standard recognisable cylindrical shape (a so-called 'can'). The number following PAR refers to the diameter of the can in eighths of an inch so a PAR56 has a

can diameter of 7 inches or approximately 18 cm. If the spotlight is not required to provide the highest levels of illumination then LED spots have a number of advantages. They are more efficient so they generate less heat and you can expect a very long service life compared with the older spots using filament lamps. Fitted with red, green and blue LEDs the spot can also produce a wide range of coloured light under DMX control without the need for additional coloured filters (gels). DMX (Digital MultipleX) is the recognised standard communication network for the control of stage and entertainment lighting. It uses a twisted pair cable in accordance with RS485 standards. A master controller (usually a DMX controller or lighting mixer desk) sends out byte-sized control information over a network of up to 512 channels (functions). The starting point for this Reader's Project was a low-cost PAR56 LED spotlight with 153 LEDs (Figure 1) which requires just five channels. This particular lamp comes with a built-in sound to light mode but the author felt it was not particularly effective. A simple (low cost) RGB DMX mixer desk allowing remote control of the lamp (and any additional lamps) would be much more useful. After much fruitless searching the author set out to build his own from scratch.

Hardware

A professional mixer desk would of course do a nice job of mixing the red green and blue spotlight colours but for the author's

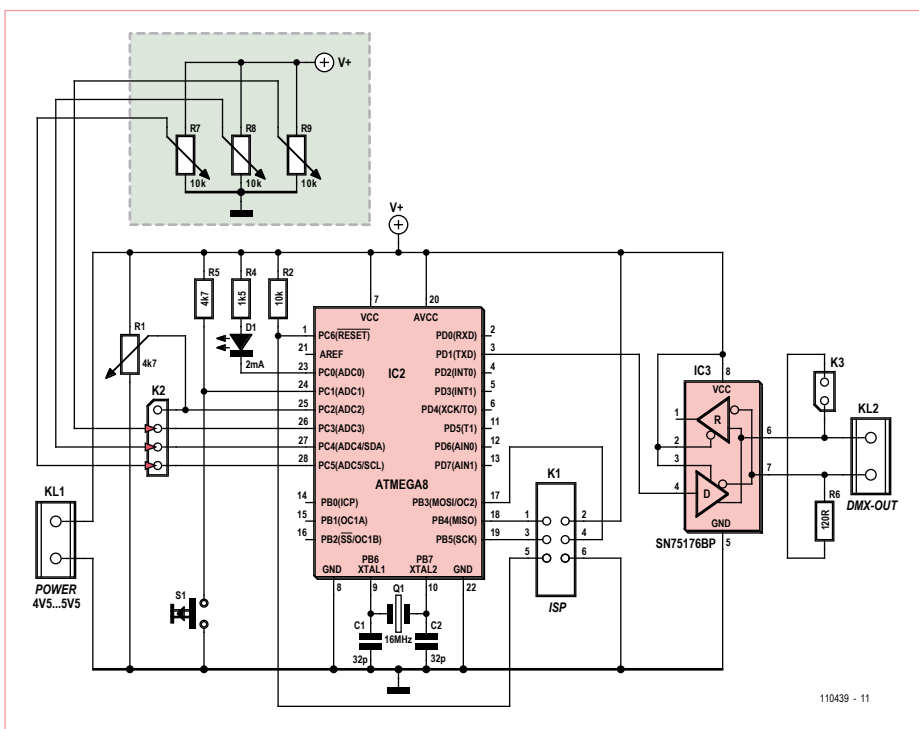


Figure 2. The circuit consists of little more than an ATmega8 microcontroller and an RS485 transceiver.

Note. Readers' Projects are reproduced based on information supplied by the author(s) only. The use of Elektor style schematics and other illustrations in this article does not imply the project having passed Elektor Labs for replication to verify claimed operation.

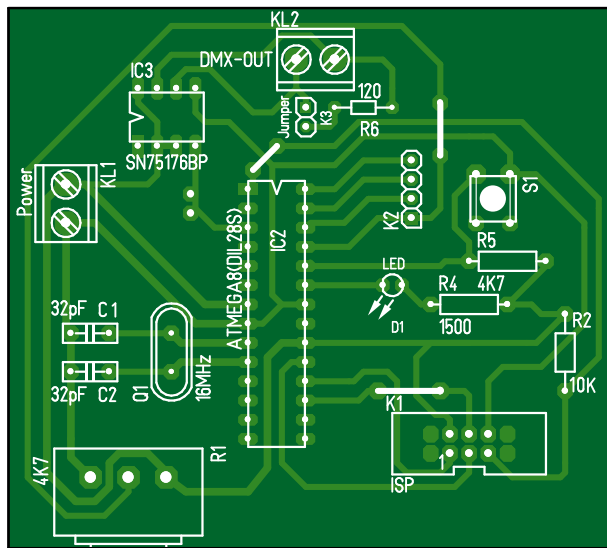


Figure 3. The simple PCB layout makes it easy to produce.

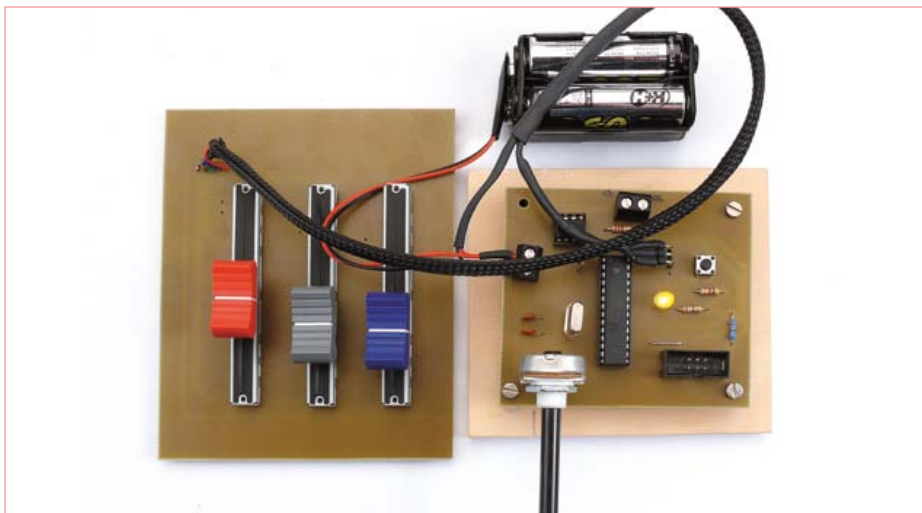


Figure 4. In the prototype version the three slider pots to control red, green and blue levels are mounted on a separate board.

needs this option was just too expensive. With an ATmega microcontroller and a few standard pots the cost of this design comes in under 10 pounds. In addition the firmware can be easily modified or expanded at will to implement any additional features that you wish for.

An Atmel ATmega8 microcontroller was chosen for the design to perform the task of reading the three analogue voltage levels at the red, green and blue slide pots, converting them to data values and sending them to the spotlight over the DMX bus. The circuit diagram shown in (Figure 2) consists of the microcontroller (IC1), an RS485 transceiver (IC2) and the pots. The

DMX protocol specifies a baud rate of 250 kBd which requires the ATmega microcontroller to be clocked with a 16 MHz crystal (X1). The mixer only needs to transmit data onto the DMX bus so the RS-485 transceiver chip is driven by the TX output from the ATmega's built-in serial interface. The RS485 bus requires one 120 Ω terminating resistor at each end. The circuit includes a 120 Ω resistor and jumper which can be used to terminate the driver end of the bus. The voltage levels produced at the sliders of the three linear dimmer pots are connected to A/D inputs on port C of the microcontroller (pins PC3 to PC5). The pots should have a logarithmic characteristic to

compensate for the non-linear response of the eye to light. A fixed resistor can be fitted in series with one end of the pot connections to optimise the control range of the slider movement. The fourth pot (connected to PC2 input) will be used to control the strobe flash frequency and should have a linear characteristic. The pushbutton S1 is used to select additional features. In the current software version it switches the circuit to strobe mode.

Power for the circuit is provided by four AA size NiMH rechargeable batteries giving a typical supply voltage of 4.8 V (5.3 V with freshly charged batteries). The circuit draws around 60 mA so 2500 mAh cells will allow more than 40 hours continuous use. Power can alternatively be supplied by a stabilised 5 V 100 mA mains adapter.

A PCB has been designed for the circuit (Figure 3). Both the PDF and T3001 (for TARGET 3001!) format files can be downloaded for free from the project website [1]. On the prototype version (see Figure 4) the RGB slide pots are mounted on a separate PCB.

Software

The firmware for this project is written in C and implements two state machines. State machine 1 controls DMX communication while state machine 2 handles the analogue to digital conversion process.

The current version of the firmware uses just five DMX communication channels to talk to the one spotlight but more channels can easily be added as necessary. The DMX communication protocol is shown in Figure 5. To generate the reset pulse (BREAK) state the serial interface is momentarily deactivated and pin PD1 pulled low. The spotlight start address is set to channel one. Brightness information for the red LEDs is sent in channel 1.

The channel assignments for the three colours are:

Red	DMX channel 1
Green	DMX channel 2
Blue	DMX channel 3

The four A/D channels are read one after the other: the voltage levels at PC3 to PC5 represent the dimmer values for RGB and PC2 (not yet implemented) for the flash

frequency (strobe value). The DMX protocol uses 8-bit data so the 10-bit values produced by the A/D converters are reduced to eight bits by right-shifting to give values in the range of 0 to 255.

An ISP connector (2x5 pin header) is provided on the PCB to allow in-circuit programming. The firmware source and hex-code files are both available for free download from the Elektor website [1].

Daisy chaining

DMX [2] uses the RS485 bus as the physical layer to transfer data allowing up to 32 receivers to be daisy chained. The LED spotlight used here loops DMX IN through to DMX OUT via a standard panel-mounted 3-pin XLR plug and socket arrangement (Figure 6). As mentioned earlier the bus requires a 120 Ω terminating resistor at each end to suppress signal reflections. The DMX mixer PCB is fitted with a 120 Ω resistor which can be used (by fitting a jumper) in to take care of the generator end of the cable. At the other end an XLR plug with a 120 Ω resistor soldered between the data pins (Figure 7) is plugged into the DMX OUT connector of the last spotlight in the chain.

Room for expansion

In the current version of the software push-button S1 switches the lamp into strobe mode but control of the strobe flash rate over DMX has not yet been implemented.



Figure 6. The input and output DMX connectors on the rear of the RGB LED spotlight. The DIP switches are used to set the DMX communication channel for the lamp.

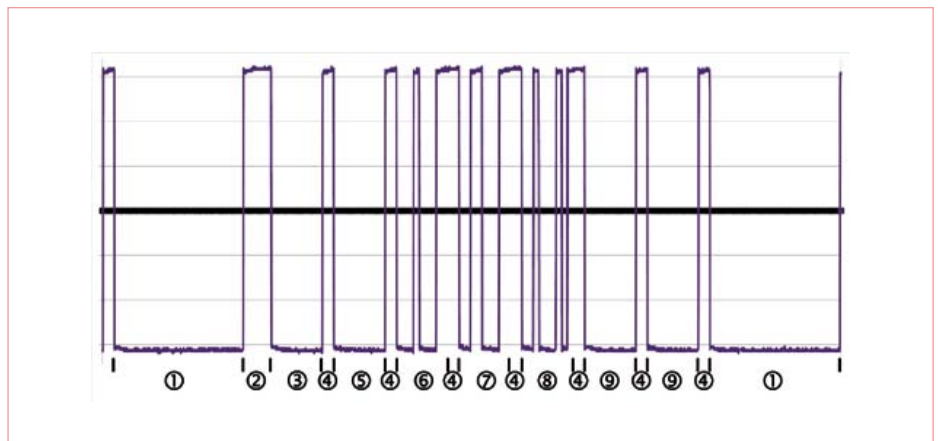


Figure 5. This DMX timing diagram shows a data block composed of the following parts:

- 1 - BREAK (91 μs)
 - 2 - Mark after BREAK (20 μs)
 - 3 - START (value must always be 0x00) START bit (4 μs) + 8 data bits (32 μs)
 - 4 - 2 STOP bits (8 μs)
 - 5 - DMX-Channel 0 START bit (4 μs) + 8 data bits (32 μs)
 - 6 - DMX-Channel 1 (red dimmer level) START bit (4 μs) + 8 data bits (32 μs)
 - 7 - DMX-Channel 2 (green dimmer level) START bit (4 μs) + 8 data bits (32 μs)
 - 8 - DMX-Channel 3 (blue dimmer level) START bit (4 μs) + 8 data bits (32 μs)
 - 9 - DMX-Channel 4 und 5 (contents = 0x00) START bit (4 μs) + 8 data bits (32 μs)
- The data bit sequence is LSB to MSB.

For this it would be necessary to read the analogue voltage level on the wiper of the pot using pin PC2 input on the microcontroller. In addition it would be necessary to use timer interrupts so that the strobe frequency would not be affected by software workload.

An interesting add-on to the controller's features would be an automatic run mode which could be triggered to vary the lamp's brightness and colour in accordance with a pre-programmed table of changes. Another possibility would be to implement a radio time receiver using DCF-77 transmissions,

making colour changes according to date and time. As soon as any newer version of the firmware has been tested it will be made available to download for free from the project website [1].

(110439)

Internet Links and Literature

- [1] www.elektor.com/110439 (The web page for this project)
- [2] www.elektor.com/010035 ('DMX512 Revealed' article in Elektor May 2001)

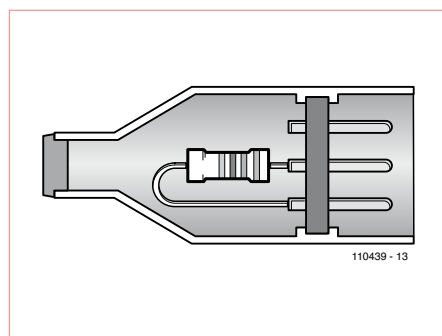


Figure 7. An XLR plug with a 120 Ω resistor between the two data pins is plugged into the 'DMX OUT' of the last spotlight in the chain to act as a bus terminator.

About the author

After qualifying as a sound engineer Ingo Sundermann went on to study industrial electronics at technical college in Cologne.

Since graduating he has enjoyed working as a development engineer specialising in vehicle diagnostics for mobile construction equipment. He has always had an interest in LED systems and their control technologies.

Componenten Tips

By Raymond Vermeulen (Elektor Labs)

USB protection

The theme for this month's instalment is USB protection. With regards to the electrical aspects of USB, there are certain details that require particular attention. The power supply current of the +5 V line has to be limited. The quick solution is often just to grab a fuse. Irrespective of whether the fuse is resettable or not, such fuses are not very accurate. The difference between the amount of current that's allowed to pass through the fuse so that it does not activate and the minimum guaranteed current to ensure that the fuse operates is often 100%(!). In addition, the data lines should be protected against ESD. Although this is frequently done, it often requires a lot of board area and there is significant capacitive loading of the lines. This can have a negative impact on the signal wave shape. To solve these problems I'd like to introduce you to two suitable ICs.

(120095)

TPD2EUSB30ADRTR

This IC is very appropriate to clamp the data lines. At first glance this doesn't look particularly special, even its price is reasonable. There are nevertheless a few remarkable features: This is a very small IC, measuring only 1x1 mm(!). So it fits nicely between the two data lines on a small circuit board. The manufacturer also claims that you can use it with differential lines that operate at higher frequencies, such as USB3.0, SATA, PCIe, etc. This will probably also work with other chips, but the waveforms will then not be as good.

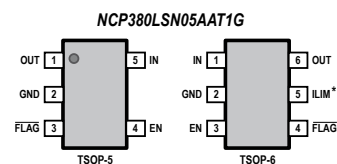
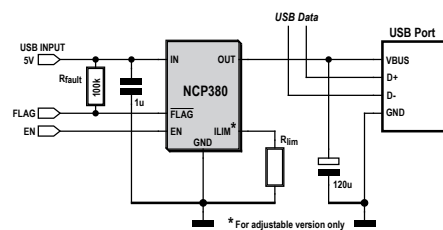
What is also remarkable is that this design does not require V_{bus} . This is very handy, because in some applications V_{bus} is not available. This IC has quite a small breakdown-voltage of 4.5 V, 3.3-V devices are therefore closely protected. A consequence of this is that the IC cannot be used on lines that carry a higher voltage than 3.3 V, but you don't see many of those any more (and none at higher speeds). It is however quite strange that the manufacturer shows d+ and d- on the component, while, in my opinion, it makes no difference in the operation if you connect the IC the other way around (they even do that in the datasheet). In any case, your data lines are protected against ESD up to ± 8 kV and the IC has a negligible affect on the signal. This can prevent a lot of trouble.

PARAMETER	CONDITION	VALUE
Reverse stand-off voltage	D-, D+	3.6 V
Breakdown voltage	$I_{io} = 1$ mA	4.5 V
ESD protection	D-, D+	8 kV
C_{io-io}	$V_{io} = 2.5$ V	0.05 pF
C_{io-gnd}	$V_{io} = 2.5$ V	0.7 pF

Internet Link: www.ti.com/lit/ds/symlink/tpd2eusb30a.pdf

NCP380LSN05AAT1G

For limiting the current I came across this IC. What caught my eye was its price, in some cases even 50% lower than comparable products(!). The drawing shows the circuit for USB Host devices. But I found it more interesting to use it in a slave device, since there is always the possibility that a project draws too much current (according to Murphy's Law). If a circuit is powered from USB as well as from an external power supply, then this IC will shut down if the output voltage is 100 mV higher than the input voltage. With USB it is convenient to use the version with the built-in 500-mA limit. Another nice feature is that this IC also offers ESD protection. A few other features are soft start and soft shutdown. This can be very useful, particularly when turning off. The series resistance when switched in amounts to only 70 m Ω , a fuse can easily be 10 times higher. If the IC overheats it will automatically switch off, the same with an input voltage that is too low. All these types of errors are signalled via the 'flag' pin. An IC such as this can come in very handy.



PARAMETER	CONDITION	VALUE
V_{in}	Operational voltage	2.5 V – 5.5 V
$V_{rev} (V_{out} - V_{in})$	Trip point	100 mV
$R_{DS(on)}$	$V_{in} = 5$ V	70 m Ω
Current limit	$V_{in} = 5$ V	[Min 0.5 A] [Typ 0.58 A] [Max 0.65 A]

Internet Link: www.onsemi.com/pub_link/Collateral/NCP380-D.PDF

Hexadoku

Puzzle with an electronics touch

How many Hexadokus can be generated using 16 x 16 hex digits? To be honest we're unsure but the number should be staggering. Here's another variant you can't claim having seen before. Simply enter the right numbers in the puzzle below. Next, send the ones in the grey boxes to us and you automatically enter the prize draw for one of four Elektor Shop vouchers.

The instructions for this puzzle are straightforward. Fully geared to electronics fans and programmers, the Hexadoku puzzle employs the hexadecimal range 0 through F. In the diagram composed of 16 x 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once

in each column and in each of the 4x4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation. Correct entries received enter a draw for a main prize and three lesser prizes. All you need to do is send us the numbers in the grey boxes.

Solve Hexadoku and win!

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

Participate!

Before April 1, 2012, send your solution (the numbers in the grey boxes) by email, fax or post to
Elektor Hexadoku – 1000, Great West Road – Brentford TW8 9HH
United Kingdom.
Fax (+44) 208 2614447 Email: hexadoku@elektor.com

Prize winners

The solution of the January 2012 Hexadoku is: **43ADE**.
The Elektor £80.00 voucher has been awarded to Christian Klems (The Netherlands).
The Elektor £40.00 vouchers have been awarded to Antje Völksch (Germany),
Raúl Elguezabal Martínez (Spain) and Marc Herzog (Luxembourg).
Congratulations everyone!

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

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

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

Elektor ‘Simple Function Generator’ (1978)

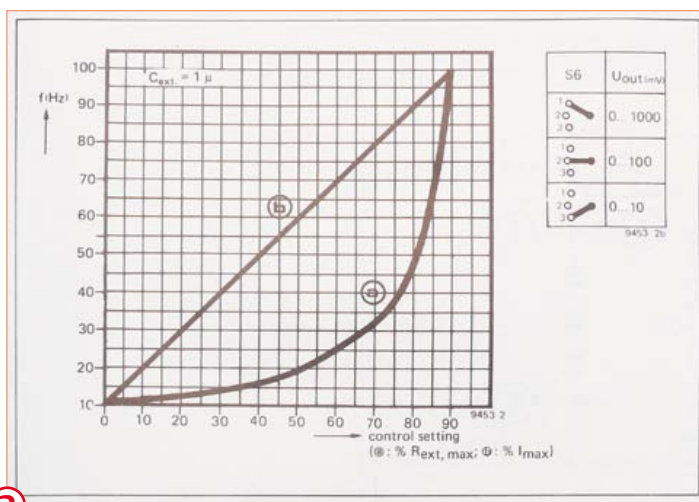
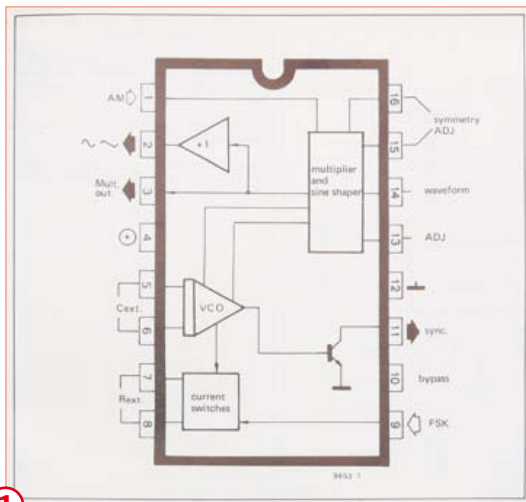
By Jan Buiting (Elektor UK/US Editorial)

Look at what’s powered up quasi permanently in any reasonably equipped, not overly microcontroller-biased electronics lab or workshop and it’s likely to be a solder iron, an adjustable power supply of some sort, an oscilloscope and a function generator, the latter often disguised as a PC. The generator is a problem solver *par excellence* and the quintessential link between equipment that’s: (tick what’s applicable)

- Faulty
- Suspected faulty
- Under active development
- Under passive development
- Incomprehensible
- Dunno
- Smoking, smelly, flaky, grotty
- Labelled ‘DOA’ but eBay savvy, surely

800 Hz or 1,000 Hz), the more advanced equipments with an adjustable output frequency covering the audio range. The more demanding service technicians and lab workers however required additional waveforms, a wider frequency range and adjustable signal levels, not forgetting stability and low distortion of the sinewave output. The ‘instruments to match’ were professional and consequently out of the financial reach of enthusiasts and those doing paper rounds. Although tube-based function generators were widely available from surplus stores in the mid 1970s, the complaint was: bulky and still too expensive.

I have been unable to get confirmation that the term ‘function generator’ refers to the mathematical functions used to derive a sinewave, triangular wave or a sawtooth from the basic rectangular wave. To this I should hasten to add that there are also function generators that derive their waveforms from the triangular wave. Some old hands told me “Jan them instruments just have loads of functions”.



and the input of your oscilloscope. Basically, the way your ‘Box X’ responds to the various signals and signal levels supplied by the function generator reveals its operation, shortfalls, troubles or inability to operate. In Bob-Pease vernacular: “One scope screen tells a thousand words” and this is where the true electronics fan gets intrigued — and Joe Bloggs says goodbye to watch TV.

Historically and functionally the function generator or ‘waveform generator’ is the “Extra-Plus-Deluxe” successor of the generator part of the old signal tracer, which typically contained a sinewave source only; the cheapest ones with a single frequency only (usually

It wasn’t until 1974/75 that an integrated circuit (IC) became widely available with everything (well almost) on board to make a powerful, feature-rich function generator at low cost. And very famous it became: the XR2206 from Exar (Figure 1). Thirty-eight years on, you can still buy it from Conrad Electronics and Jaycar! In 1977, at Elektor, designer/editor Ernst Krempelsauer (now retired, see inset) set out to work and effectively turned the rather dry XR2206 data-sheet into a hugely successful DIY project # EPS 9453 successfully replicated by many thousands of readers in the years following publication. What a wonderful achievement. On YouTube I saw someone using a shoddily built version of the instrument to demonstrate

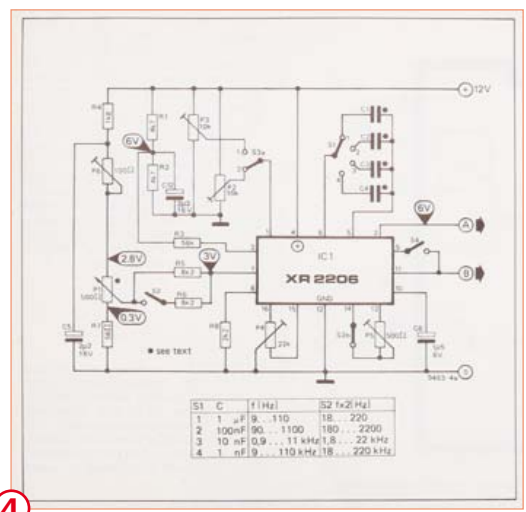
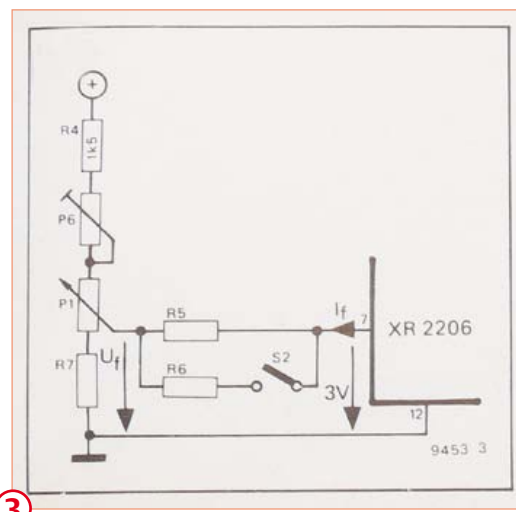
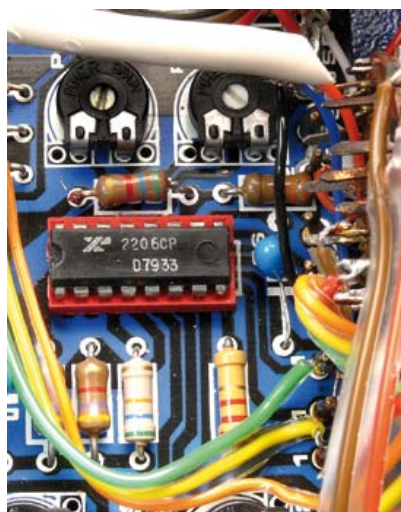
Retronics is a monthly column covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcomed; please send an email to editor@elektor.com



a small JYETech DSO 062 oscilloscope [1]. An Elektor schematic with the XR2206 is also on the momentous RadioMuseum.org website, which is a sign of recognition if not a feat [2].

The really clever bit of Ernst’s design around the XR2206 is the way an ordinary potentiometer is used to obtain a nearly linear frequency control, all thanks to a constellation of resistive elements

If you want the full details & maths, read the original article [3]. The upshot is that the adjustment range of *linear* pot P1 enables the frequency to be varied *virtually linearly* over slightly more than a decade, i.e. from 9 Hz to 110 Hz for example. As a bonus, the frequency can be doubled when switch S2 is closed (assuming R5=R6). Of the three partial diagrams that make up the instrument’s complete schematics, I’ve chosen the XR2206 section to reproduce here



at the frequency control input of the chip. I’ve reproduced the relevant illustrations in **Figures 2 and 3**. Without this circuitry, you’d end up with a frequency control response the shape of a hyperbole, or a quest for a negative logarithmic potentiometer — both options are mathematician’s delights but in practice, nothing to look forward to.

In blissful ignorance, Fred in The Shed having connected nothing but a pot between pins 7 and 8 of the XR2206 gets

$$f = 1 / (R_{ext} C_{ext})$$

whereas with the Elektor *I-V* converter thinkware at the I_f pin,

$$f = (3 - U_f) / (3 R_s C_{ext})$$

in **Figure 4**. Attention all Sticklers-4-Copyrights: I am allowed to do so. The output amplifier and power supply aren’t terribly exciting to be honest, being based on dead common parts like BC140, BC160 trannies and an L130 voltage regulator. Funny though, ‘AC’ is printed to indicate the generator’s output signal terminals. Being low impedance, the output supplies ‘alternating current’ for sure but ‘AC’ in my mind equates to power cords, power outlets and terminals you don’t want to touch. In 1977 too, Elektor staff were an international bunch, the editors speaking in different tongues and probably unable to agree on industry-standard English wordings like “output”, “out” or “signal out”.

Calibrating the frequency scale of the instrument was remarkably easy and based on feeding a few volts of 100-Hz hum (tapped from a bridge rectifier), and the generator output signal to a single small

Designer Ernst Krempelsauer remembers...

When I was able for the first time to get hold of a sample of Exar's function generator IC type XR2206 I had been with Elektor for just over a year, having been promoted, rather unexpectedly, to Elektor Germany Deputy Editor in June 1975.

The XR2206 was not the first IC of its kind, Intersil's ICL 8038 having been used in an Elektor circuit as early as 1973. However the 2206 was much better in terms of frequency range, waveform quality, versatility and ease of use. After studying the datasheet and intensive testing of the device an article with tried out example circuits was published in Elektor September 1975, which included the option of linear frequency setting by using a control voltage.

In spite of that all 2206 function generators on the market continued to use the Exar application circuit, i.e. adjusting the frequency with a potentiometer as a variable resistance, which causes an exponential frequency scale. My aim was to design a function generator with a linear frequency scale, enabling relatively accurate frequency adjustment without the use of a frequency counter.

However, the following year I was busy with the well known FORMANT synthesizer project (in addition to my job as an Editor), so that the function generator had to wait. My goal was a very practical design — simple to build and to adjust and providing optimum cost/performance ratio. This included a classic front panel design inspired by the front panels of the FORMANT modules. The optimization of the PCB by Peter Verhoosel took a little more time than expected, but finally the "Simple Function Generator" article was published in the October 1977 issue of Elektor Germany. Apparently at the right time, because thousands of circuit boards # 9453 were sold and kits stayed on the market for many years — even illegally reproduced versions "based on" the Elektor design...



loudspeaker and then adjusting a trimpot for zero beat. The specimen of the Simple Function Generator pictured here was rescued from the former Elektor offices and safeguarded in gross violation of corporate instructions at the time "to get rid of the old grot".

It's easy to tell from the Perspex cover that the lab prototype was built for displaying at shows and exhibitions, which flourished back then. The huge equipment ID label affixed to the underside is a clear indicator of the equipment having travelled extensively, passing Customs & Excise many times (European borders were very real in those days). A cable between the generator output and the loudspeaker inside the case is cut, telling a woeful tale of Elektor exhibition booth crew fed up with the sounds from the instrument when the next visitor felt an urgent need to turn controls or flip switches. Petrolheads always want to kick tyres.

The instrument had some loose wires inside and worryingly the XR2206 was missing. However Luc Lemmens came to my rescue by digging a 1980s Playtronic WOG 2206 sig gen out of his removal boxes and pulling the XR2206 chip from it. The WOG 2206 looks suspiciously like the Elektor project, as you can see from **Figure 5**. Half an hour later the Elektor generator was up and running again. A case of *Seven-Year Itch* or just plain coincidence, a reworked version of the function generator was published in Elektor in 1984. This design, housed in a grey & white Verobox, addressed and eradicated problems with small spikes in the sinewave. There were some other improvements too, but the instrument was still based on the XR2206. As far as I have been able to ascertain, it was a blast just like the 1978 version.

Today, laptop PCs and even smartphones running dedicated software for the inbuilt sound system seem to have taken over as function generators in the modern electronics lab. They do a good job in terms of stability, accuracy and distortion, but somehow I prefer to turn real controls on a dedicated instrument secure on my test

equipment shelf. Judging from the steady interest in Elektor's 1978 and 1984 function generator publications, mine is not an uncommon preference.

(120068)

To celebrate the resurfacing of the Simple Function Generator, a scanned copy of the original article from Elektor January 1978 may be downloaded free of charge [2]. Regrettably parts, circuit boards and tech support are no longer available for the project.

Internet References

- [1] www.youtube.com/watch?v=KPtsgFw5Fno
- [2] www.radiomuseum.org/tubes/tube_xr2206.html
- [3] www.elektor.com/120068



Consulting

By Gerard Fonte (USA)

When I was a kid in high school I liked to try to fix TV's and radios ('try' was the operative word). I wasn't very good. But I never charged unless I truly repaired something, so the client had nothing to lose. Often they would say that if I couldn't fix it, I could keep it (for parts). This illuminates two important concepts: 1) I was not being entrusted with anything valuable and 2) it was a clear conflict of interest.

Trust is a Must

The first rule of consulting is that clients must believe that you can do what you say you can do. Obviously, if they don't, you won't get any business. Therefore, you have to make yourself credible. As a high-school student, it was potentially possible that I could fix a TV or radio (and on at least several occasions, I did). Thus, if you're just starting out on your own to make a few extra bucks, be reasonable. People expect young entrepreneurs to be good at things like creating web-sites or fixing computers. But, if you say you can streamline their accounts receivable system or can design spread-spectrum receivers, they'll be skeptical. It's also important to specialize. No one can do everything. I've never seen an engineer 'handyman'. It's probably best to choose something you like. In that way, you'll be more likely to enjoy your work and also keep up with the latest developments (learning is forever).

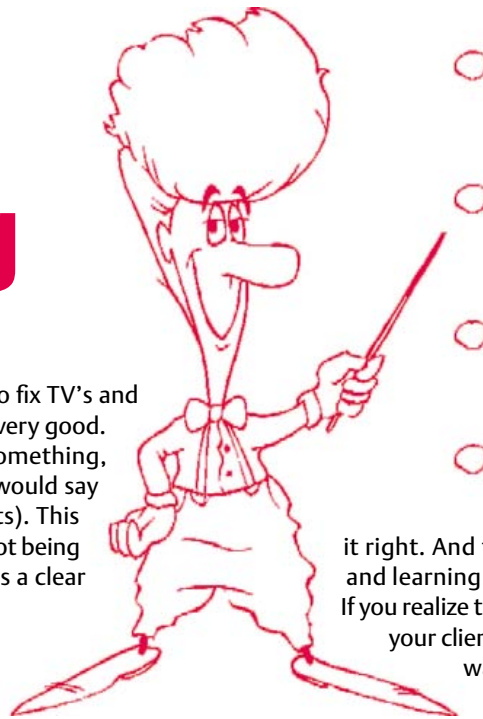
However, there is a down side to working your hobby. People generally use their pastime to relax. If that's your work too, you may not find it very soothing to come home and do more work. It can be very hard to separate business from pleasure if they are both the same thing. Unfortunately, there is no easy way to predict if this will happen to you.

Results Oriented

The first rule of consulting (yes, there are two first rules) is that you must be able to do what you say you can do. There is nothing worse than hiring someone and finding out weeks or months later that the person is floundering. Unfortunately, this is not uncommon. There are many people who say they can do something when they can't. Don't be one of them. Don't let the lure of a contract make you promise something you can't achieve. And don't fool yourself into thinking you can do anything.

When a client asks for something you haven't done before, tell him that. This is hugely important. It immediately gives you credibility. And being believed is crucial (see rule one). Then tell him that you will need a few days to examine the project. During those days, research and develop the key parts of the contract. If possible, build a breadboard and test it. That way you *know* you can complete the assignment.

Sometimes, you will need to learn a new skill or specialty or tool. Maybe it's a new computer language or how to build an FPGA (Field Programmable Gate Array). This is not something that you can obtain and master in a day or so. In this case, tell the client



you will need a couple of weeks or so to get up to speed (and make it clear that you are not charging for this learning time). If you can show your customer that this approach is superior to other methods, he will rarely balk at giving you the extra time to do it right. And then spend that time diligently studying and learning.

If you realize that you are in over your head, you must tell your client immediately. Don't make excuses. Don't wait until he asks why there are no results. Don't pretend you're making progress, when you're not. And, above all, don't lie

to yourself. It's one thing to encounter a sticky technical problem like an invisible software bug, or too much noise in an amplifier. It's quite something else if you can't determine how to design your product, or if you don't understand the test results. If you're having problems with fundamentals, you aren't swimming.

Simply have a face-to-face meeting with your client (no e-mails or telephone calls) and tell him that you are unable to complete the assignment, return any money and apologize. Yes, it's embarrassing and humiliating. But that's what happens when you over-reach. Quite honestly, most clients are understanding when this occurs. And while they aren't happy about the lost time, most realize that people make mistakes. Few will shout and throw things.

Money Matters

There is always the question of how much to charge. This is quite variable depending on the specialty, area of the country and the experience you have. Generally \$25 to \$50 per hour is a reasonable for starting out.

Some people charge by the hour. I charge by the job. It's been my experience that everyone benefits with 'Firm-Fixed Contract'. In this way, there are no surprises. And you learn very quickly just how long a particular task will take. I have the client pay actual costs for major purchases (like PC boards). (Some people add 10% or so for 'handling'). Separating out the big expenses makes my quote seem lower and more attractive.

It's vital that you have a contract for the first time you work with a client. This spells out exactly what you will do and what the client will do. Obviously you will perform some labor. But in order to succeed, you will need: information from the client, perhaps special parts, test procedures, access to proprietary instrumentation, etc. The contract defines success — exactly what the product is and its specifications. There is nothing worse than presenting a 'finished' product only to find that it doesn't do what the client wants (or said he wanted). After a few jobs, and after you understand each other well, a contract may not always be necessary. But neither party should ever balk at having one.

Contracting is a lot of fun and you get to work on a lot of different projects. Unlike a regular nine-to-five job, you have a lot of flexibility and every day is different.

(120100)

ELEKTOR SHOWCASE

NEW: ELEKTOR SHOWCASE ONLINE • coming soon on www.elektor.com
 365 days per year all Showcase ads online with up to date and relevant information.
 For more information: Johan Dijk, j.dijk@elektor.com

BRITISH AMATEUR ELECTRONICS CLUB ARCHIVE



<http://baec.tripod.com/>

The British Amateur Electronics Club Archive Website. Archives from 140+ Newsletters from 1966-2002. Currently have interesting and useful selected articles from 21 Newsletters. Also a section about built electronics projects with schematics and photos. Plus useful info., downloads and links. NO ADVERTS!

FIRST TECHNOLOGY TRANSFER LTD.

<http://www.ftt.co.uk>

- Training and Consulting for IT, Embedded and Real Time Systems
- Assembler, C, C++ (all levels)
- 8, 16 and 32 bit microcontrollers
- Microchip, ARM, Renesas, TI, Freescale
- CMX, uCOSII, FreeRTOS, Linux operating systems
- Ethernet, CAN, USB, TCP/IP, Zigbee, Bluetooth programming



SCOPES and more

HAMEG
Instruments

A Rohde & Schwarz Company

www.hameg.com

Great Value in TEST & MEASUREMENT

LCD Displays

2 line x 16 character LCD's with Backlight



£4.00 each + P&P

Wow!



4 line x 16 character LCD's with Backlight

£3.50 each + P&P

www.cstech.co.uk/elektor

FLEXIPANEL LTD

www.flexipanel.com

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

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



minty geek

satisfy your inner geek

a fresh approach

Cool portable electronic projects and kits



www.mintygeek.com

EASYSYNC LTD.

www.easysync-ltd.com/

Supplier of communications and instrumentation products with specialist expertise in serial connectivity solutions based on USB, CAN and RS232/RS422/ RS485 interfaces.

- USB to Serial RS232/RS422/RS485 converter cables.
- CANbus solutions
- Ethernet to Serial Adapters or to USB hubs.
- USB based Logic Analysers, Oscilloscopes & Data Loggers.
- OEM & ODM design services.



FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD.

www.ftdichip.com <<http://www.ftdichip.com>>

FTDI specialise in USB silicon, hardware and software solutions.

- USB WHQL complaint drivers.
- USB host and slave solutions.
- Free firmware development tools.
- USB IC's, modules, cables and turnkey custom solutions.
- World renowned FOC application support.

USB MADE EASY



MaxSonar Ultrasonic Range Finder

XL-MaxSonar-EZ

- Beam pattern choice
- High acoustic power
- Real-time calibration
- 39.95USD / unit

MaxSonar-WRC IP67

- Compact packaging
- Quality narrow beam
- 99.95USD / unit

www.active-robot.co.uk
www.coolcomponents.co.uk
www.oceancontrols.com.au
www.maxbotix.com



ELNEC

www.elnec.com

Europe's leading device programmers manufacturer:

- reliable HW:
- 3 years warranty for most programmers
- support over 65.000 devices
- free SW updates
- SW release: few times a week
- excellent technical support: Algorithms On Request, On Demand SW
- all products at stock / fast delivery



HEXWAX LTD

www.hexwax.com

World leaders in Driver-Free USB ICs:

- USB-UART/SPI/I2C bridges
- TEAleaf-USB authentication dongles
- expandIO-USB I/O USB expander
- USB-FileSys flash drive with SPI interface
- USB-DAQ data logging flash drive

www.elektor.com

products and services directory

ROBOT ELECTRONICS

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

Advanced Sensors and Electronics for Robotics

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



ROBOTIQ

<http://www.robotiq.co.uk>

Build your own Robot!

Fun for the whole family!

Now, available in time for X-mas

- Arduino Starter Kits *NEW!!*
 - Lego NXT Mindstorms
 - Affordable Embedded Linux Boards
 - Vex Robotics (kits and components)
 - POB Robots (kits and components)
- email: sales@robotiq.co.uk Tel: 020 8669 0769



www.elektor.com

TYDER

<http://www.tyder.com>

- ONEoverT Digital Filter Design Software (Full version for only £30)
- Design FIRs, IIRs, NCOs, FFTs for DSPs and FPGAs
- VHDL Code Generators
- Makes DSP design simple
- Download demos from website

TYDER IP for DSP



NEW: ELEKTOR SHOWCASE ONLINE

coming soon on
www.elektor.com

365 days per year all
Showcase ads online
with up to date and
relevant information.

For more information:
Johan Dijk
j.dijk@elektor.com

Create complex electronic systems in minutes using Flowcode 5

FLOWCODE5

Design → Simulate → Download



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



NEW!
*Flowcode 5
for PIC*

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

Going Strong

A world of electronics from a single shop!



A whole year of Elektor magazine onto a single disk

DVD Elektor 2011

The year volume DVD/CD-ROMs are among the most popular items in Elektor's product range. This DVD-ROM contains all editorial articles published in Volume 2011 of the English, American, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy of PCB layouts at printer resolution, adapt PCB layouts using your favourite graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-276-1 • £23.50 • US \$37.90



Bestseller!

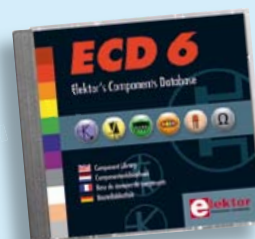
Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Circuits issues 2001-2010 of Elektor, supplemented with various other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & modelling, home & garden, high frequency, power supply, robotics, test & measurement and of course a section miscellaneous for everything that didn't fit in one of the other sections.

ISBN 978-1-907920-06-6

£34.50 • US \$55.70



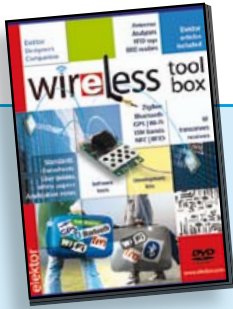
More than 70,000 components

CD Elektor's Components Database 6

This CD-ROM gives you easy access to design data for over 7,800 ICs, more than 35,600 transistors, FETs, thyristors and triacs, just under 25,000 diodes and 1,800 optocouplers. The program package consists of eight databanks covering ICs, transistors, diodes and optocouplers. A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data.

ISBN 978-90-5381-258-7

£24.90 • US \$40.20



RFID, NFC, Zigbee, GPS and more

DVD Wireless Toolbox

On this DVD-ROM you'll find a number of technical documents and tools that will enable you to add wireless data exchange to your electronics systems. The choice of equipment depends on the transmission distance: a few centimetres using Near Field Communication (NFC) or Radio Frequency Identification (RFID), tens of metres with the Bluetooth, Wi-Fi or ZigBee systems, or indeed thousands of kilometres using a module for receiving GPS data. The DVD contains technical documentation (spec. sheets, application notes, user guides, etc.) on various devices according to the frequency and/or protocol used. All of the documents are PDF files (in English).

ISBN 978-90-5381-268-6
£28.50 • US \$46.00



110 issues, more than 2,100 articles

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.

ISBN 978-0-905705-76-7
£69.00 • US \$111.30



Bestseller!

Processor design in the real world

Microprocessor Design using Verilog HDL

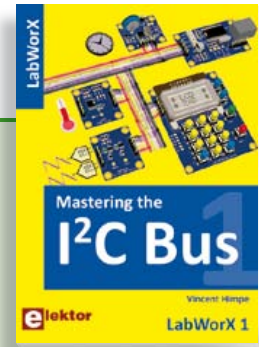
If you have the right tools, designing a microprocessor shouldn't be complicated. The Verilog hardware description language (HDL) is one such tool. It can enable you to depict, simulate, and synthesize an electronic design, and thus increase your productivity by reducing the overall workload associated with a given project. This book is a practical guide to processor design in the real world. It presents the Verilog HDL in an easily digestible fashion and serves as a thorough introduction about reducing a computer architecture and instruction set to practice. You're led through the microprocessor design process from the start to finish, and essential topics ranging from writing in Verilog to debugging and testing are laid bare.

337 pages • 978-0-9630133-5-4
£27.90 • US \$45.00

More information on the
Elektor Website:
www.elektor.com

Elektor
Regus Brentford
1000 Great West Road
Brentford
TW8 9HH
United Kingdom
Tel.: +44 20 8261 4509
Fax: +44 20 8261 4447
Email: order@elektor.com

elektor

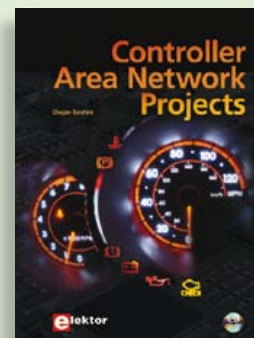


LabWorX: Straight from the Lab to your Brain

Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C Bus and its applications. Besides the Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I²C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9
£29.50 • US \$47.60

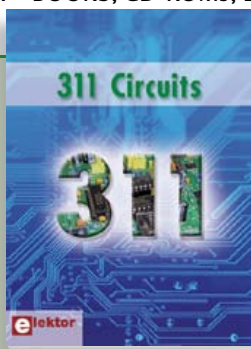


Free mikroC compiler CD-ROM included

Controller Area Network Projects

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters.

260 pages • ISBN 978-1-907920-04-2
£29.50 • US \$47.60

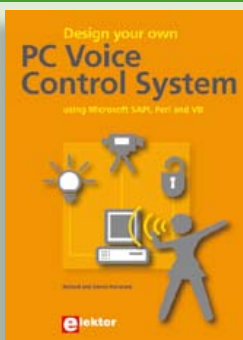


Creative solutions for all areas of electronics

311 Circuits

311 Circuits is the twelfth volume in Elektor's renowned 30x series. This book contains circuits, design ideas, tips and tricks from all areas of electronics: audio & video, computers & microcontrollers, radio, hobby & modelling, home & garden, power supplies & batteries, test & measurement, software, not forgetting a section 'miscellaneous' for everything that doesn't fit in one of the other categories. 311 Circuits offers many complete solutions as well as useful starting points for your own projects.

420 pages • ISBN 978-1-907920-08-0
£29.50 • US \$47.60



Talk with your computer

Design your own PC Voice Control System

This book guides you through practical speech recognition, speech annunciation and control of really useful peripherals. It details a project which will enable you to instruct your computer using your voice and get it to control electrical devices, tell you the time, check your share values, get the weather forecast, etc. and speak it all back to you in a natural human voice. If you are interested in the practical technology of interfacing with machines using voice, then this book is your guide!

216 pages • ISBN 978-1-907920-07-3
£29.50 • US \$47.60



AndroPod

(February 2012)

With their high-resolution touchscreens, ample computing power, WLAN support and telephone functions, Android smartphones and tablets are ideal for use as control centres in your own projects. However, up to now it has been rather difficult to connect them to external circuitry. Our AndroPod interface board, which adds a serial TTL port and an RS485 port to the picture, changes this situation.

Andropod module with RS485 Extension

Art.# 110405-91



Improved Radiation Meter

(November 2011)

This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive samples. The photodiode has a smaller sensitive area than a Geiger-Müller tube and so has a lower background count rate, which in turn means that the radiation from a small sample is easier to detect against the background. A further advantage of a semiconductor sensor is that it offers the possibility of measuring the energy of each particle.

Kit of parts incl. display and programmed controller

Art.# 110538-71 • £35.50 • \$57.30



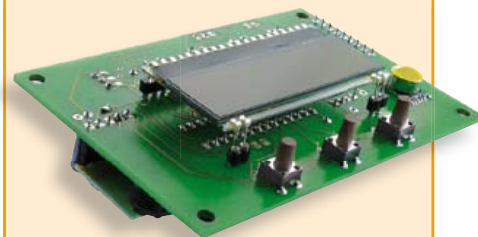
FT232R USB/Serial Bridge/BOB

(September 2011)

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

PCB, assembled and tested

Art.# 110553-91 • £12.90 • US \$20.90



USB Long-Term Weather Logger

(September 2011)

This stand-alone data logger displays pressure, temperature and humidity readings generated by I²C bus sensors on an LCD panel, and can run for six to eight weeks on three AA batteries. The stored readings can be read out over USB and plotted on a PC using gnuplot. Digital sensor modules keep the hardware simple and no calibration is required.

Kit of parts incl. PCB, controller, humidity sensor and air pressure sensor modules

Art.# 100888-73 • £31.10 • US \$50.20

March 2012 (No. 423)

£ US \$

+ + + Product Shortlist March: See www.elektor.com + + +

February 2012 (No. 422)

AndroPod (1)

110258-91 USB/RS485 converter, ready-made module	22.2035.70
110405-91 Andropod module with RS485 Extension.....	www.elektor.com	
110553-91 USB-FT232R breakout board, assembled and tested..	12.9020.90
120103-92 5-way cable USB 2.0 A male to USB micro-B, black.....	3.505.70
120103-94 5V / 1A (5W) PSU with micro-USB connector.....	8.0012.90

Pico C-Plus and Pico C-Super

110687-41 Pico C-Plus controller, programmed (ATTINY2313-20PU).....	4.40 7.10
110687-42 Pico C-Super controller, programmed (ATTINY2313-20PU).....	4.407.10

Electronics for Starters (2)

ELEX-1 Prototyping board	4.907.90
ELEX-2 Prototyping board (double size)	8.8514.30

January 2012 (No. 421)

Wideband Lambda Probe Interface

110363-41 Programmed controller ATMEGA8-16AU.....	8.8514.30
-----------	----------------------------------------------	------	------------

Audio DSP Course (7)

110002-71 Printed circuit board partly populated with SMD's	44.5071.80
-----------	-------------------------------------------------------------	-------	------------

Grid Frequency Monitor

110461-41 Programmed controller AT89C2051-24PU, for 50 HZ areas (Europe)	8.8514.30
110461-42 Programmed controller AT89C2051-24PU, for 60 Hz areas (USA)	8.8514.30

Here comes the Bus! (11)

110258-1 Experimental Node Board.....	5.308.60
110258-1C3	... 3 pcs Experimental Node Board	11.5018.60
110258-91 USB/RS485-Converter, ready made module.....	22.2035.90

Time / Interval Meter with ATtiny

080876-41 Programmed ATtiny2313	7.7012.60
-----------	----------------------------------	------	------------

December 2011 (No. 420)

Here comes the Bus! (10)

110258-1 Experimental Node board.....	5.308.60
110258-1C3	... 3 pcs Experimental Node board	11.5018.60
110258-91 USB/RS485 Converter, ready made module	22.2035.90

USB Data Logger

110409-1 Printed circuit board.....	9.7515.70
110409-41 Programmed controller PIC24FJ64GB002-I/sp dil-28s	13.3021.40

November 2011 (No. 419)

Improved Radiation Meter

110538-41 Programmed controller ATmega88PA-PU.....	9.3515.10
110538-71 Kit of parts incl. display and programmed controller	35.5057.30

Simple Bat Detector

110550-1 PCB, bare	8.8514.30
----------	----------------------	------	------------

OnCE/JTAG Interface

110534-91 Programmer board, assembled and tested	35.6057.30
-----------	---------------------------------------------------	-------	------------

Here comes the Bus! (9)

110258-1 Experimental Node board.....	5.308.60
110258-1C3	... Printed circuit board 3x print Experimental Node	11.5018.60
110258-91 USB/RS485 Converter, ready made module	22.2035.90

Dual Linear PSU for Model Aircraft

081064-1 Printed circuit board.....	14.5023.80
----------	----------------------------------	-------	------------

October 2011 (No. 418)

Versatile Board for AVR Microcontroller Circuits

100892-1 Printed circuit board.....	11.5518.70
----------	----------------------------------	-------	------------

Audio DSP Course (4)

110001-91 PCB, populated and tested DSP board.....	115.70186.70
110001-92 Bundle DSP board (110001-92) with Programmer (110534-91)	133.50215.00

Bestsellers

Books	1	↑	Microprocessor Design using Verilog HDL ISBN 978-0-9630133-5 £27.90US \$45.00
	2	→	Controller Area Network Projects ISBN 978-1-907920-04-2 £29.50US \$47.60
	3	↻	311 Circuits ISBN 978-1-907920-08-0 £29.50US \$47.60
	4	↻	Mastering the I²C Bus ISBN 978-0-905705-98-9 £29.50US \$47.60
	5	↻	Design your own PC Voice Control System ISBN 978-1-907920-07-3 £29.50US \$47.60
CD/DVD-ROMS	1	→	CD 1001 Circuits ISBN 978-1-907920-06-6 £34.50US \$55.70
	2	↻	DVD Elektor 1990 through 1999 ISBN 978-0-905705-76-7 £69.00 ...US \$111.30
	3	↻	CD Elektor's Components Database 6 ISBN 978-90-5381-258-7 £24.90US \$40.20
	4	↑	DVD Wireless Toolbox ISBN 978-90-5381-268-6 £28.50US \$46.00
	5	↻	DVD Elektor 2010 ISBN 978-90-5381-267-9 £23.50US \$37.90
Kits & Modules	1	→	Improved Radiation Meter Art. # 110538-71 £35.50US \$57.30
	2	→	FT232R USB/Serial Bridge/BOB Art. # 110553-91 £12.90US \$20.90
	3	↻	Here comes the Bus! Art. # 110258-91 £22.20US \$35.90
	4	↻	Audio DSP Board + Programmer Art. # 110001-92 £133.50 ...US \$215.00
	5	↻	USB Long-Term Weather Logger Art. # 100888-73 £31.10US \$50.20

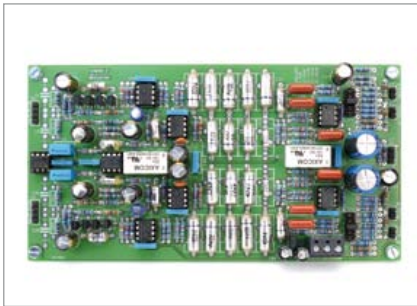
Order quickly and securely through
www.elektor.com/shop
or use the Order Form near the end
of the magazine!

elektor
Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH • United Kingdom
Tel. +44 20 8261 4509
Fax +44 20 8261 4447
Email: order@elektor.com



Electromechanical Thermometer

Electronic thermometers with an LCD can be found for little money even at supermarkets, but the true electronics fan of course wants a more original way to display temperatures. The one described in this article is very special indeed, using the mechanical score counter of an old pinball machine. The electromechanical rolling-digit counter is controlled by an ATtiny2313 receiving a TMP100 temperature sensor signal and supplying counter pulses to the excitation coils.



Preamp-2012

After the remarkable 5532 OpAmplifier audio power amp published in Elektor's October and November 2010 editions audio designer Douglas Self set to work on a matching pre-amplifier, also employing a good number of paralleled NE5532 opamps. Over the past months Elektor Labs designed circuit boards to match this comprehensive design with its multitude of adjustment and control capabilities. In the April 2012 edition we kick off by discussing the preamp's overall structure. To whet your appetite the high-end MD/MC preamplifier is pictured here.



RS485 Switch Board

The ElektorBus project is ideal not just for measurement, switching and control tasks, but also for home automation applications. In the next edition we present a compact switch module for home automation applications. The board comprises two relays, an ATmega88 micro and an RS485 driver, the lot enabling AC powered loads to be switched on and off remotely. In addition, two controller inputs are accessible via screw terminals so switches and other devices are easily connected too. As a matter of course the software developed for this module is fully ElektorBus compatible.

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

Elektor UK/European April 2012 edition: on sale March 22, 2012.

Elektor USA March 2012 edition: published March 19, 2012.

www.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com

Elektor on the web

All magazine articles back to volume 2000 are available individually in pdf format against e-credits. Article summaries and component lists (if applicable) can be instantly viewed to help you positively identify an article. Article related items and resources are also shown, including software downloads, hyperlinks, circuit boards, programmed ICs and corrections and updates if applicable.

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

Also on the Elektor website:

- Electronics news and Elektor announcements
- Readers Forum
- PCB, software and e-magazine downloads
- Time limited offers
- FAQ, Author Guidelines and Contact



ORDERING INSTRUCTIONS, P&P CHARGES

All orders, except for subscriptions (for which see below), must be sent BY POST or FAX to our Brentford address using the Order Form overleaf. Online ordering: www.elektor.com/shop

Readers in the USA and Canada should send orders, except for subscriptions (for which see below), to the USA address given on the order form. Please apply to Elektor US for applicable P&P charges. Please allow 4-6 weeks for delivery.

Orders placed on our Brentford office must include P&P charges (Priority or Standard) as follows: Europe: £6.00 (Standard) or £7.00 (Priority) Outside Europe: £9.00 (Standard) or £11.00 (Priority)

HOW TO PAY

All orders must be accompanied by the full payment, including postage and packing charges as stated above or advised by Customer Services staff.

Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London. IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP).

Please ensure your full name and address gets communicated to us.

Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.

GCredit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in this magazine. If difficulties in the supply of components are envisaged, a source will normally be advised in the article. Note, however, that the source(s) given is (are) not exclusive.

TERMS OF BUSINESS

Delivery Although every effort will be made to dispatch your order within 2-3 weeks from receipt of your instructions, we can not guarantee this time scale for all orders. **Returns** Faulty goods or goods sent in error may be returned for replacement or refund, but not before obtaining our consent. All goods returned should be packed securely in a padded bag or box, enclosing a covering letter stating the dispatch note number. If the goods are returned because of a mistake on our part, we will refund the return postage. **Damaged goods** Claims for damaged goods must be received at our Brentford office within 10-days (UK); 14-days (Europe) or 21-days (all other countries). **Cancelled orders** All cancelled orders will be subject to a 10% handling charge with a minimum charge of £5.00. **Patents** Patent protection may exist in respect of circuits, devices, components, and so on, described in our books and magazines. Elektor does not accept responsibility or liability for failing to identify such patent or other protection. **Copyright** All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, diskettes and software carriers published in our books and magazines (other than in third-party advertisements) are copyright and may not be reproduced or transmitted in any form or by any means, including photocopying and recording, in whole or in part, without the prior permission of Elektor in writing. Such written permission must also be obtained before any part of these publications is stored in a retrieval system of any nature. Notwithstanding the above, printed-circuit boards may be produced for private and personal use without prior permission. **Limitation of liability** Elektor shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connexion 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 in respect of the goods. **Law** Any question relating to the supply of goods and services by Elektor shall be determined in all respects by the laws of England.

January 2012

SUBSCRIPTION RATES FOR ANNUAL SUBSCRIPTION

	Standard	Plus
United Kingdom & Ireland	£54.00	£66.50
Surface Mail		
Rest of the World	£68.50	£81.00
Airmail		
Rest of the World	£86.00	£98.50
USA & Canada	See www.elektor.com/usa for special offers	

HOW TO PAY

Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London. IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP). Please ensure your full name and address gets communicated to us.

Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.

Credit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

SUBSCRIPTION CONDITIONS

The standard subscription order period is twelve months. If a permanent change of address during the subscription period means that copies have to be despatched by a more expensive service, no extra charge will be made. Conversely, no refund will be made, nor expiry date extended, if a change of address allows the use of a cheaper service.

Student applications, which qualify for a 20% (twenty per cent) reduction in current rates, must be supported by evidence of studentship signed by the head of the college, school or university faculty. A standard Student Subscription costs £43.20, a Student Subscription-Plus costs £55.70 (UK only).

Please note that new subscriptions take about four weeks from receipt of order to become effective. Cancelled subscriptions will be subject to a charge of 25% (twenty-five per cent) of the full subscription price or £7.50, whichever is the higher, plus the cost of any issues already dispatched. Subscriptions cannot be cancelled after they have run for six months or more.

January 2012

• **Subscribe** to *audioXpress* magazine!

Do your **electronics speak** to you? Are the words **"audio"**, **"vacuum tubes"**, and **"speaker technology"** music to your ears?

Then you should be **reading *audioXpress*!**

Recently acquired by The Elektor Group, *audioXpress* has been providing engineers with incredible audio insight, inspiration and design ideas for over a decade. If you're an audio enthusiast who enjoys speaker building and amp design, or if you're interested in learning about tubes, driver testing, and vintage audio, then *audioXpress* is the magazine for you!

What will you find in *audioXpress*?

- In-depth interviews with audio industry luminaries
- Recurring columns by top experts on speaker building, driver testing, and amp construction
- Accessible engineering articles presenting inventive, real-world audio electronics applications and projects
- Thorough and honest reviews about products that will bring your audio experiences to new levels

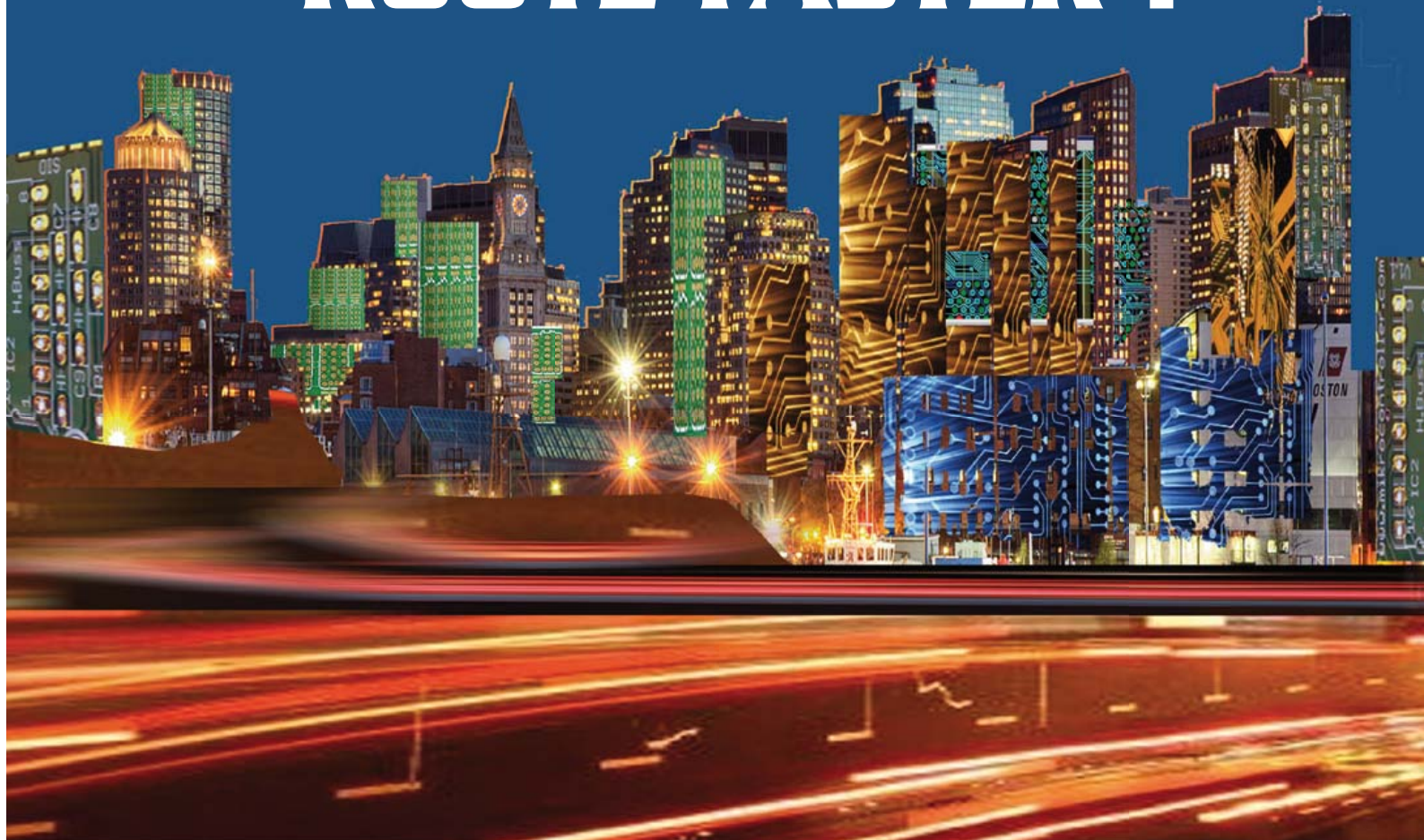
Choose from print delivery, digital, or a combination of both for maximum accessibility.

Subscribe to *audioXpress* at www.cc-webshop.com today!

audioXpress



ROUTE FASTER !



WITH PROTEUS PCB DESIGN

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

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

PROTEUS DESIGN SUITE Features:

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

Prices start from just £150 exc. VAT & delivery

labcenter  www.labcenter.com
Electronics

Visit our website or
phone 01756 753440
for more details

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