



3D Pad

Touchless Gesture Control Interface

point to make the point



- **3D Pad** | Wireless Gateways | Intelligent Cuelight System for Theaters
- Microcontroller BootCamp | **RGB LED Lamp** | Current Probe with Transimpedance Amp
- Grid Frequency Logger** | Elektor Radiation Meter using PC | Zero-Electrolytics 555 Timer
- PCB Transformer | **Mystery Cap** | Post & Win
- Speedy 1200+ Acoustic Modem

US \$9.00 - Canada \$10.00



WIZnet

CONNECT THE MAGIC 2014 DESIGN CHALLENGE

LAUNCH: MARCH 3 / DEADLINE: AUGUST 3

Your mission?

Join the IoT revolution by designing an innovative 'Net-enabled electronics system using WIZnet's WIZ550io Ethernet module or W5500 chip.

Your reward? **A share of \$15,000 in cash prizes** and worldwide recognition from WIZnet, *Circuit Cellar*, and *Elektor*!

Anyone with a WIZ550io Ethernet module or W5500 smart Ethernet chip is encouraged to participate. The competition starts on March 3, 2014 at 12:00 PM EST and ends on August 3, 2014 at 12:00 PM EST. Get started today!

www.circuitcellar.com/wiznet2014

WIZnet's WIZ550io module is a complete Ethernet solution that includes the W5500 smart Ethernet chip (10/100 MAC/PHY, hardware TCP/IP, 32KB RAM buffer) and network interface (transformer and RJ-45) along with built-in MAC and default IP addresses.

For technical support: wizwiki.net/forum
To purchase: shopwiznet.com or shop.wiznet.eu

POWERED BY



Sign up for CC.Post to receive announcements and notifications!



sensors

expo & conference

www.sensorsexpo.com

June 24-26, 2014

Donald E. Stephens Convention Center • Rosemont, IL

Sensing Technologies Driving Tomorrow's Solutions

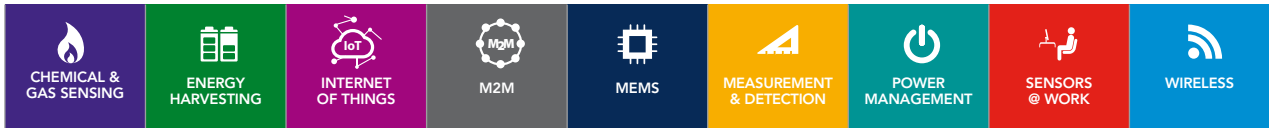


**SPECIAL
Subscriber
Discount!**

Register with code **A303C**
for \$50 off Gold and Main
Conference Passes.*


What's Happening in 2014:

Tracks



Plus+

- Full-day Pre-Conference Symposia
- Technology Pavilions on the Expo Floor
- **Internet of Things**
- **Energy Harvesting**
- **MEMS**
- **Wireless**
- **High Performance Computing**

- New** Co-location with High Performance Computing Conference
- Best of Sensors Expo 2014 Awards Ceremony 
 - Networking Breakfasts
 - Welcome Reception
 - Sensors Magazine Live Theater
 - And More!

Featuring Visionary Keynotes:



Reimagining Building Sensing and Control

Luigi Gentile Polese
Senior Engineer

Department of Energy, National Renewable Energy Lab



Sensors, The Heart of Informatics

Henry M. Bzeih
Head of Infotainment & Telematics

Kia Motors America



Innovative Applications. Expert Instructors. Authoritative Content. Tomorrow's Solutions.

Register today to attend one of the world's largest and most important gatherings of engineers and scientists involved in the development and deployment of sensor systems.

Registration is open for Sensors 2014!

Sign up today for the best rates at www.sensorsexpo.com or call 800-496-9877.



#sensors14

OFFICIAL PUBLICATION:

sensors

INDUSTRY SPONSOR

elektor

CO-LOCATED WITH:

hpc
High-Performance Computing Conference

*Discount is off currently published rates. Cannot be combined with other offers or applied to previous registrations.



● Projects

8 3D Pad: Touchless Gesture Control Interface

An electrode plane PCB, an Arduino Uno, and a shield are the ingredients for DIY-ing a circuit that supplies reliable 3-dimensional co-ordinates data. From now on it's wave, don't touch!

18 Wireless Gateways

If you do not fancy writing SPI drivers for LPR radio modules, consider using an ATmega micro, it's sure to make life much easier as far as programming is concerned.

26 Intelligent Cuelight System for Theaters

As opposed to most cuelight systems, this one's got a microcontroller built in to give the Stage Manager very accurate control of timings in a play or performance. Shakespeare would have loved it.

34 Microcontroller BootCamp (2)

This month we cut our teeth on digital inputs, not necessarily limited to digital signals though as we also venture out to sawtooth waveforms and latching effects.

42 An RGB LED Lamp

Traditional incandescent light sources found in night clubs, stage lighting, car instrument panels and interior room lighting have all benefited from a semi-conductor makeover. Here we add the convenience of a ready made hand-held remote controller.

48 Current Probe with Transimpedance Amp

To make precise measurements of alternating currents we use a wide variety of ferrite cores in conjunction with a simple circuit incorporating a transimpedance amplifier.

52 Grid Frequency Logger

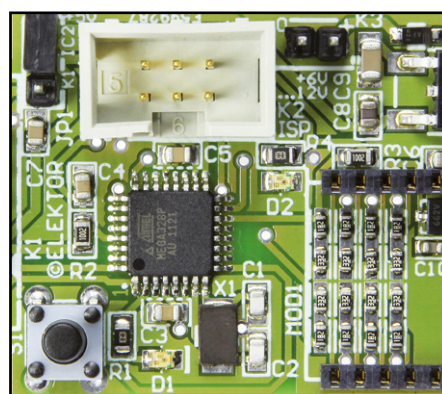
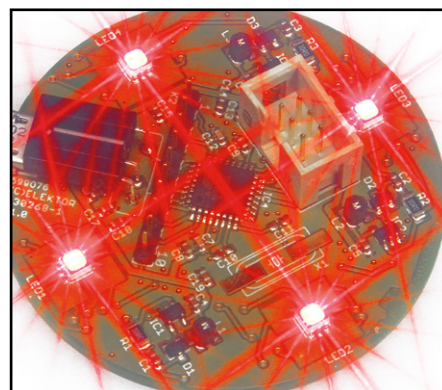
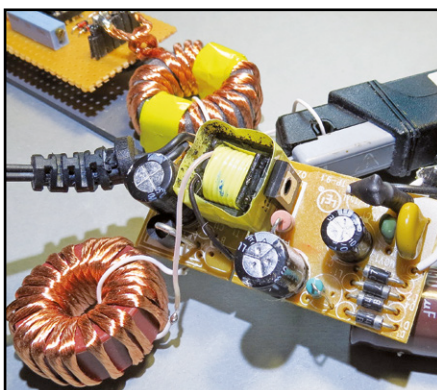
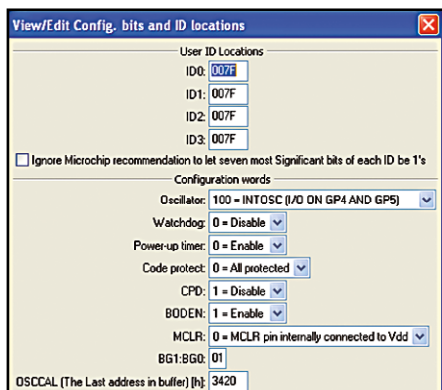
Our popular Grid Frequency Monitor now gets extended and adapted to supply a stream of data enabling a log file to be compiled.

58 Elektor Radiation Meter using PC

Here Reinier Ott explains his methods of migrating the hugely popular Elektor Radiation Meter from ATmega to PIC, with some pretty impressive results.

74 Zero-Electrolytics 555 Timer

There's nothing scary or odd about using giga-ohm resistors, in fact for our application they are much preferred over leaky wide-tolerance electrolytics.



● Labs

- 64 **PCB real estate ‘transformed’**
Does it take copper to build a power transformer? Yes, but not necessarily one with wire.
- 66 **What’s up with this cap? (2)**
So what was that ‘105’ print on last month’s mystery capacitor all about in relation to ‘104’ on the flip side?
- 68 **Post and Win**
This month: Join our live Q & A sessions on popular projects from the labs; Why copy/paste may not work on the labs website.

● DesignSpark

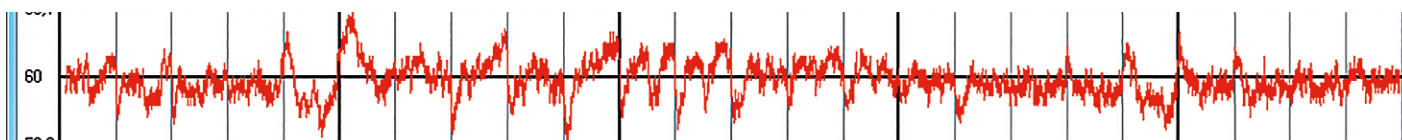
- 70 **DesignSpark Tips & Tricks**
Day #10: Custom Outline Models
It’s quite instructive to see what happens if you import a PCB back into a 3D model.
- 72 **Unijunction Transistors**
Weird Components—the series

● Industry

- 76 **News & New Products**
A selection of news items received from the electronics industry, labs and organizations.

● Regulars

- 80 **Retronics**
Speedy 1200+ Acoustic Modem (1984). A well-researched, critical recollection of digital communications over the telephone in the pre-internet age. Series Editor: Jan Buiting.
- 84 **Hexadoku**
The Original Elektorized Sudoku.
- 85 **Gerard’s Columns: Appearing Strong**
A column or two from our columnist Gerard Fonte.
- 90 **Next Month in Elektor**
A sneak preview of articles on the Elektor publication schedule.



Volume 40, No. 449
May 2014

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

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

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

and

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

Head Office:

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

USA / Canada Memberships:

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

UK / ROW Memberships:

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

USA / Canada Advertising:

Peter Wostrel
Phone: 1.978.281.7708
E-mail: peter@smmarketing.us

UK / ROW Advertising:

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

www.elektor.com/advertising

Advertising rates and terms available on request.

Copyright Notice

The circuits described in this magazine are for domestic and educational use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, DVDs, software carriers, and article texts published in our books and magazines (other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning and recording, in whole or in part without prior written permission from the Publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The Publisher disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from schematics, descriptions or information published in or in relation with Elektor magazine.

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



New and old are relative

Recently I bought a Bluetooth audio receiver device for the price of two Elektor magazines. It pairs to my smartphone or PC and outputs a stereo audio signal through a pair of RCA sockets. I am using it to replace the messed up RF and IF sections of a 1960s tube radio I was given. Although I should have been able to repair the 2- and 3-tube sections, the cost and time involved is no match against dropping the Bluetooth device in the huge wooden case, and wiring it to the volume pot. I effectively disconnected the faulty sections in the radio. Now, it plays my favorite songs, and I can even tune it remotely using the FM radio app on my smartphone. Just as state of the art electronics can provide functionality equivalent to, or surpassing, old technology, the reverse process also applies occasionally. I am referring to the 3D Touch Pad project in this issue—where a good old 4046 phase locked loop IC from the early 1980s is the key component in a system that might be part of the next generation of smartphones or tablets. After typing, speaking, touching and swiping keyboards, screens and pads, you can now look forward to gesturing to get your smart device to understand your commands and selections. With some practicing of course. A version of the project, OutsideBox, is now about to apply for crowdfunding.

I am also happy to publish afterburner articles on two of our most exciting and popular projects, the Improved Radiation Meter and the Grid Frequency Analyzer. Both started out as modest, all-experimental setups with their respective authors. After post engineering by labs and publication in the magazine they got a warm reception from the Elektor crowd and beyond, witness the number of kits sold and the flood of responses and discussions on our forums. The Radiation Meter now appears with a PIC in control, while the Grid Frequency Analyzer is extended with a logger function.

Speaking of the Elektor forums, they have been restyled, cleaned and relocated to forum.elektor.com. Write access is granted through your Elektor Labs user name and password. Looking forward to seeing your contributions there. Happy Reading,

Jan Buiting, Editor-in-Chief

The Team

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



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



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



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



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



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



Spain
Jaime González-Arintero
+34 6 16 99 74 86
j.glez.arintero@elektor.es



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



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



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



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



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



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



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



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

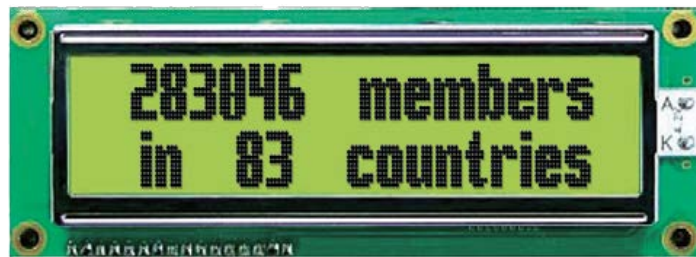


China
Cees Baay
+86 21 6445 2811
CeesBaay@gmail.com

Our Network



Connects You To



Supporting Companies



Batronix
www.batronix.com 23



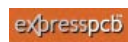
Cleverscope
www.cleverscope.com 57



DLP Design
www.dlpdesign.com 25



Eurocircuits
www.elektorpcbservice.com 67



Express PCB
www.expresspcb.com 77



Front Panel Express
www.frontpanelexpress.com 57



Pico
www.picotech.com/ps242 92



Pololu
www.pololu.com 63



Saelig
www.saelig.com 25



Sensors Expo 2014
www.sensorsexpo.com 3



SID Symposium
www.sid.org 47

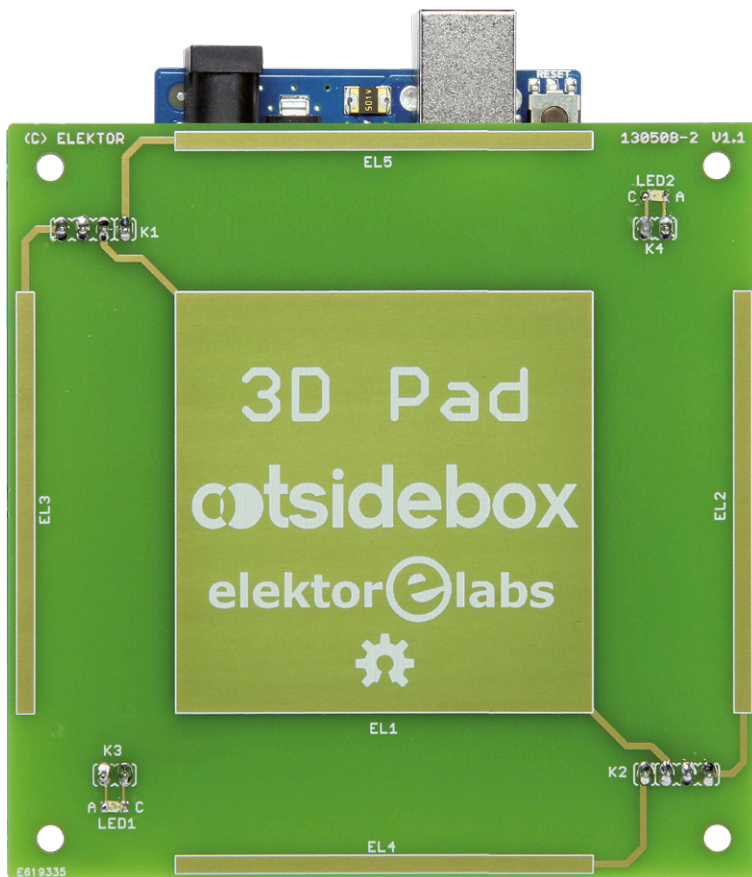


WIZnet
www.circuitcellar.com/wiznet2014 23

Not a supporting company yet?

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

3D Pad: Touchless Gesture Control interface point to make the point



By Jean-Noël Lefebvre (France)

You can easily produce a Touchless Gesture Control interface that's capable of reliably providing 3-dimensional co-ordinates (X, Y, and Z axes). This simple, experimental device, designed in collaboration with the Elektor Labs team, consists of a sandwich made up of a PCB called an electrode plane, an Arduino Uno, and a shield, with control software.



Be a part of the new revolution in person/machine interfaces, which began a long time ago with... punched cards, followed by keyboards, screens, joysticks, mice... Today, touch screens are everywhere, with so-called 'gesture' control, but using finger contact (to zoom, unlock, etc.) The new step that Elektor is inviting you to take now is *touchless* interaction, using gestures in the air in three dimensions. This isn't science-fiction—this article is here to prove to you that it's even ... at your very fingertips.

Detection principle

How about 'projecting' capacitive detection?

To detect the proximity of the human body (e.g. the hand or finger), we are going to be making use of a technique known as projected capacitive [3]. This is the principle of virtually all the touch screens on modern phones and tablets—except that here we are going to be detecting gestures in the air, relatively far from the detector surface (10 cm max.) The principle is simple: normally, in our circuits, we expect a capacitor to present

the least possible leakage of the electrostatic field outside its plates. Well, for our proximity detector, it's just the opposite. We need a capacitor that's as open as possible, built using copper tracks (which we'll be calling *electrodes*) on an epoxy panel, so as to obtain maximum "hand effect" (**Figure 1a**).

An oscillator under the influence?

The capacitor formed by our electrodes is part of a logic-gate oscillator (**Figure 1b**) whose frequency is influenced by the proximity of a hand or finger when it enters the electrostatic field. This intruder in fact forms a third electrode which is going to cut the field lines and divert the electrical charges. One of the electrodes, connected to an inverter output, at low impedance, is called *emitting*. The other, connected to the junction between the resistor R and the inverter input is called *receiving*; this is at high impedance (depending on the value of R). Result: the closer the hand approaches the electrodes, the greater the extent to which the capacitance between the electrode diminishes, and the more the oscillator frequency increases.

When the hand is 10 cm from the electrodes, the oscillator frequency only varies very slightly—a few hundred ppm at the very most (100 ppm = 0.01%!) Now with this sort of oscillator, the frequency is largely determined by other factors too – in particular, the temperature and supply voltage. So the main challenge in our circuit is to distinguish the very slight variations in the oscillation frequency caused by the hand from those resulting from the other influencing factors. One part of this task is entrusted to the software, so it will also read from two reference electrodes that are not (or only slightly) influenced by the hand, but which are subject to all the other factors.

"Switching" oscillator

As we're not dealing here with a simple proximity detector, but with a system capable of providing co-ordinates in three dimensions (X, Y, and Z axes), our oscillator will be connected to one of six receiving electrodes in turn. This device has four spatialization electrodes: top, bottom, left, right, and two reference electrodes. In the centre of the electrode plane is the single emitting electrode (Figure 1a). To ensure a good response time for the person/machine interface, the six electrodes will be scanned in turn around 200 times a second, in an operation vaguely akin to

Specifications

- Touchless 3D gesture detection
- Projected capacitance via a plan of four electrodes, Z = 4 inches/ 10 cm
- Arduino shield and sketch
- *Creative Commons* license

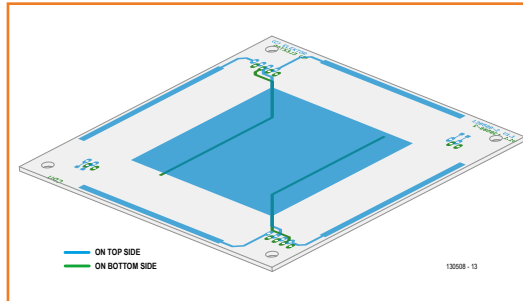


Figure 1a. The first step in understanding this circuit is to imagine the four electrodes as capacitors: in the center, the common electrode, and around it the 4 spatialization electrodes (top, bottom, left, right)

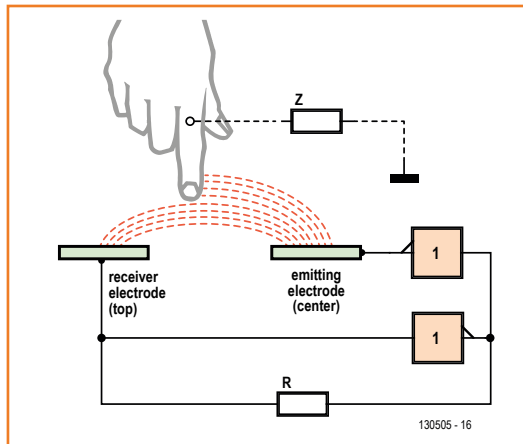


Figure 1b. By diverting part of the capacitive field projected by the electrodes, the hand upsets the oscillator frequency.

What's it used for?

There's a wide range of applications—in the kitchen, for example, to adjust the oven or induction ring controls without touching them with your greasy fingers! In the medical field, when care staff no longer have to touch devices, the risk of nosocomial infections will be reduced. In the huge field of artistic expression, with instruments like the *Theremin* which are played by using gestures in space. In the gaming field, like *Fruit Ninja* [1] or *Despicable Me—Minion* [1]. Instead of touching the tablet screen, you'll soon be able to play with movements of your hand in the air. I'm counting on you and your imagination and impatient to see what you can create around the 3D-Pad!

Who is it useful to?

Anyone who wants it, as it's a **very open design**. To make it easier to build and share, and above all to encourage you to express your own creativity, it is presented in the form of a shield for Arduino. All the design documentation, diagrams, and software are *Open Source* under a *Creative Commons CC BY-NC-SA 4.0* License [2].

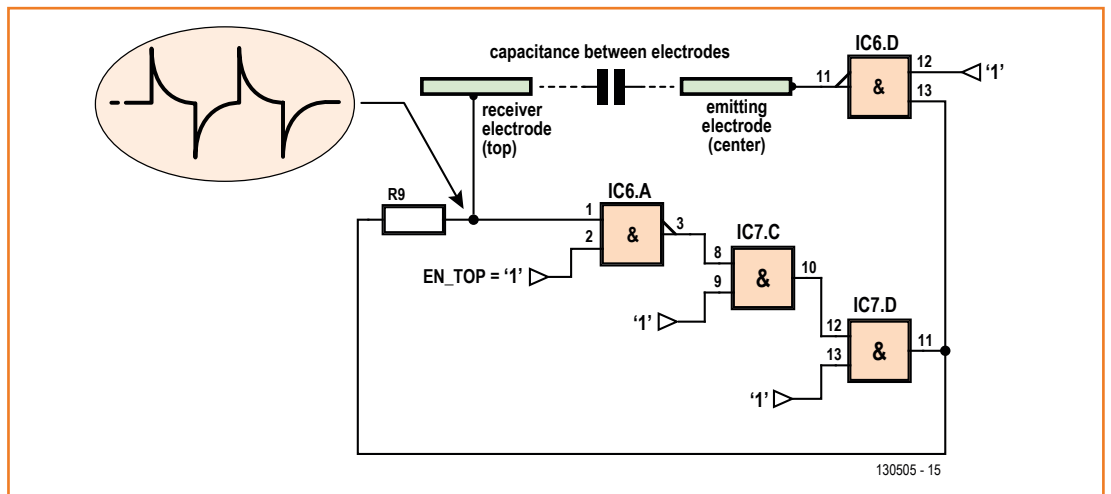


Figure 1c. Detail of the switched electrode oscillator circuit from Figure 3. Here, the logic circuit is shown configured with the top electrode (EN_TOP = 1) connected to the oscillator.

multiplexing. The software’s job, at each step in the scanning, is to analyze the oscillator frequency with each of the electrodes in turn.

Figure 1c shows the electrode oscillator in the state it is at the moment the top receiving electrode is in circuit. The control signal EN_TOP is high, all the other control signals EN_X are low. The three ports IC6A, IC7C, and IC7D now form just a simple inverter. IC6D is also configured as an inverter.

So this circuit is equivalent to that of the “easily influenced oscillator” in Figure 1b. With a bit of concentration, once you have familiarized yourself with the circuit, you’ll be able to find it in the full circuit diagram in **Figure 3**, by spotting

the logical AND and NAND operators (IC6, IC7, and IC8). You’ll also see that the switching and receiving electrode (EN_X) selection signals come from the Arduino Uno board (top left). The nominal frequency of our oscillator, in the absence of hand influence, is set by R9–R14 at around 1.6–1.8 MHz.

Now that we have an oscillator whose frequency varies by a few hundred ppm according to the proximity of a hand, all that remains for us to do is to derive – if possible using a simple solution—from that a signal that can be used for a final application. To do this, we need a frequency comparator comprising:

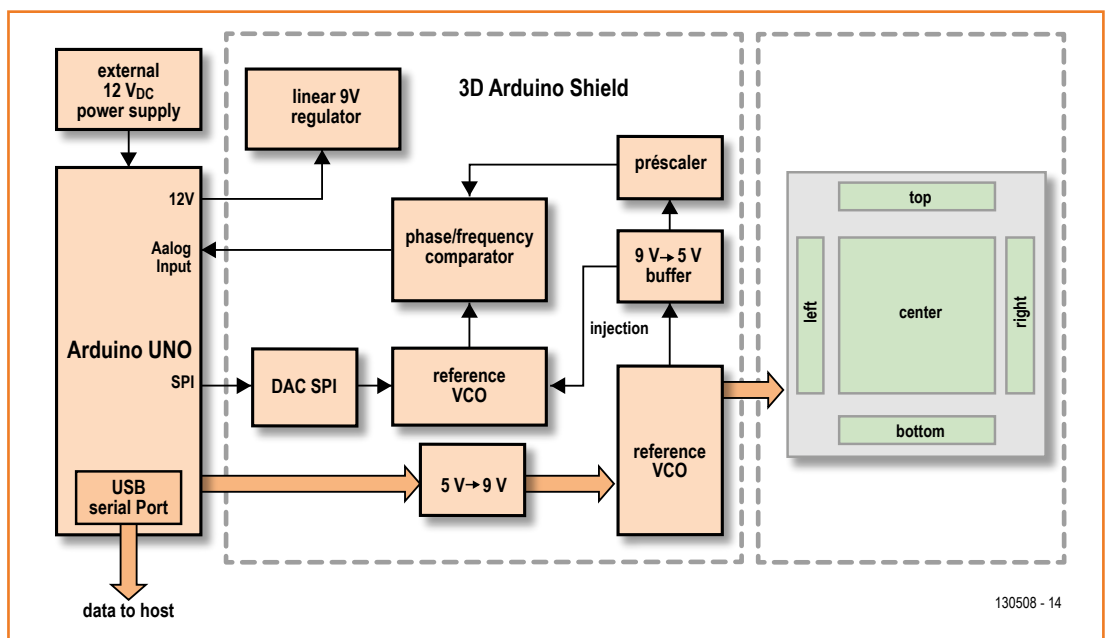


Figure 2. The functions of each of the three boards.

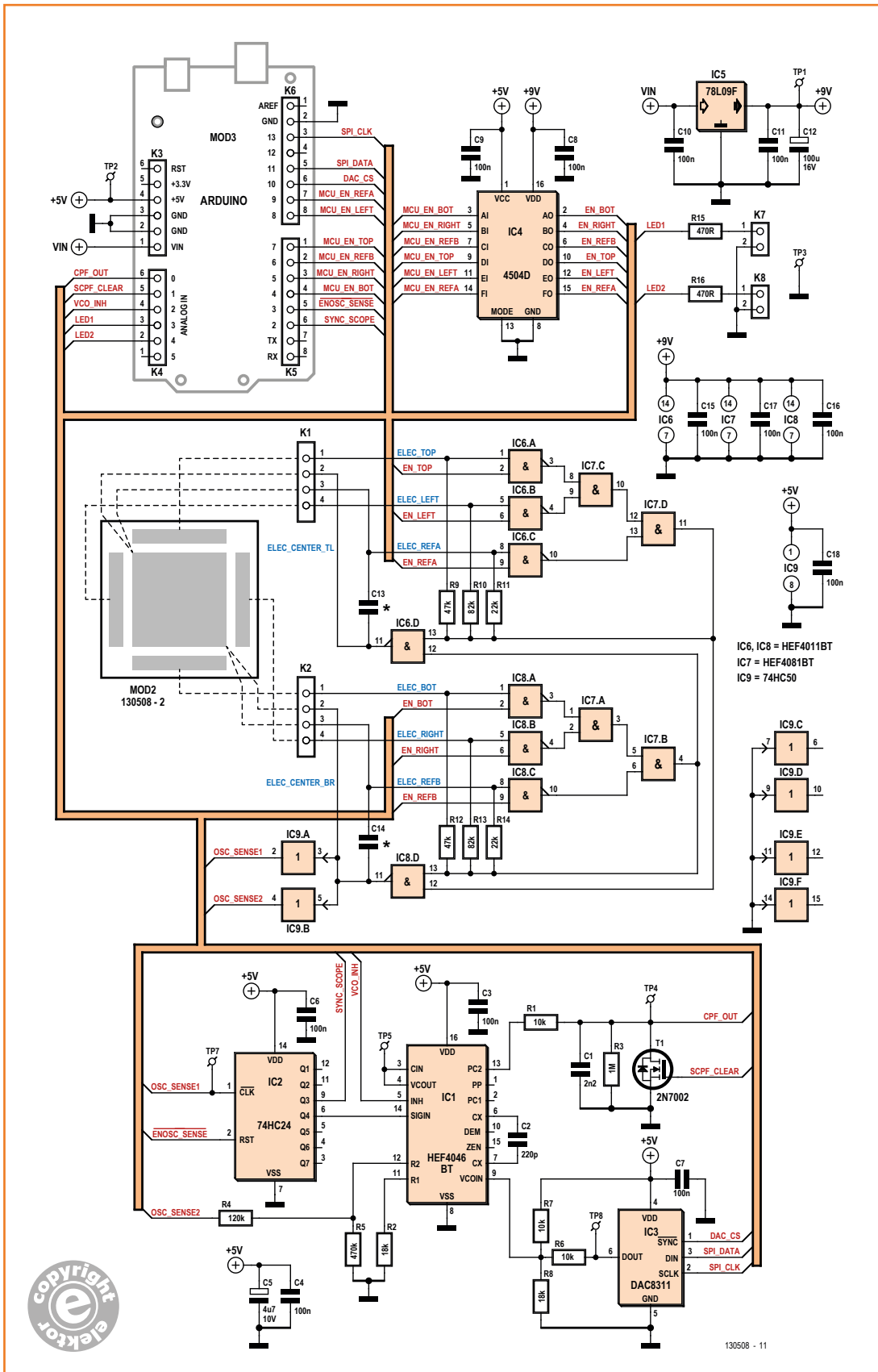
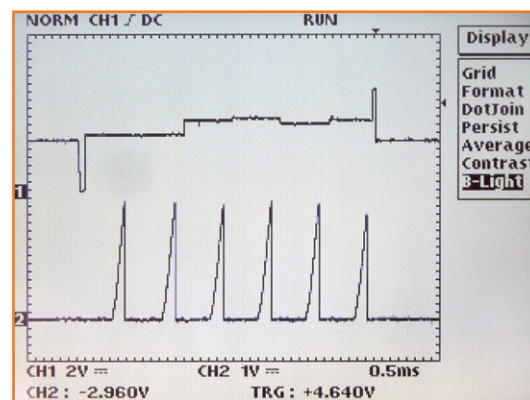
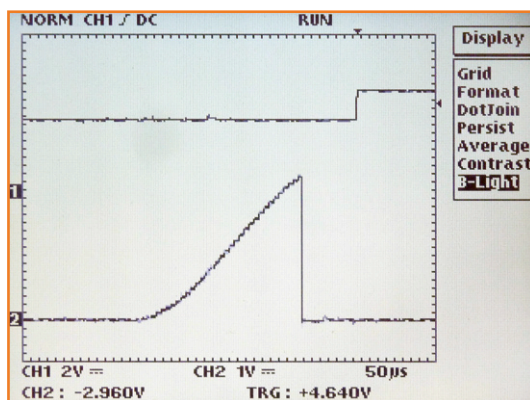


Figure 3. Full circuit diagram. Top left, the Arduino Uno board and below it, the electrode plane. Everything else is on the Arduino shield. The supply voltage comes from the Arduino Uno board.

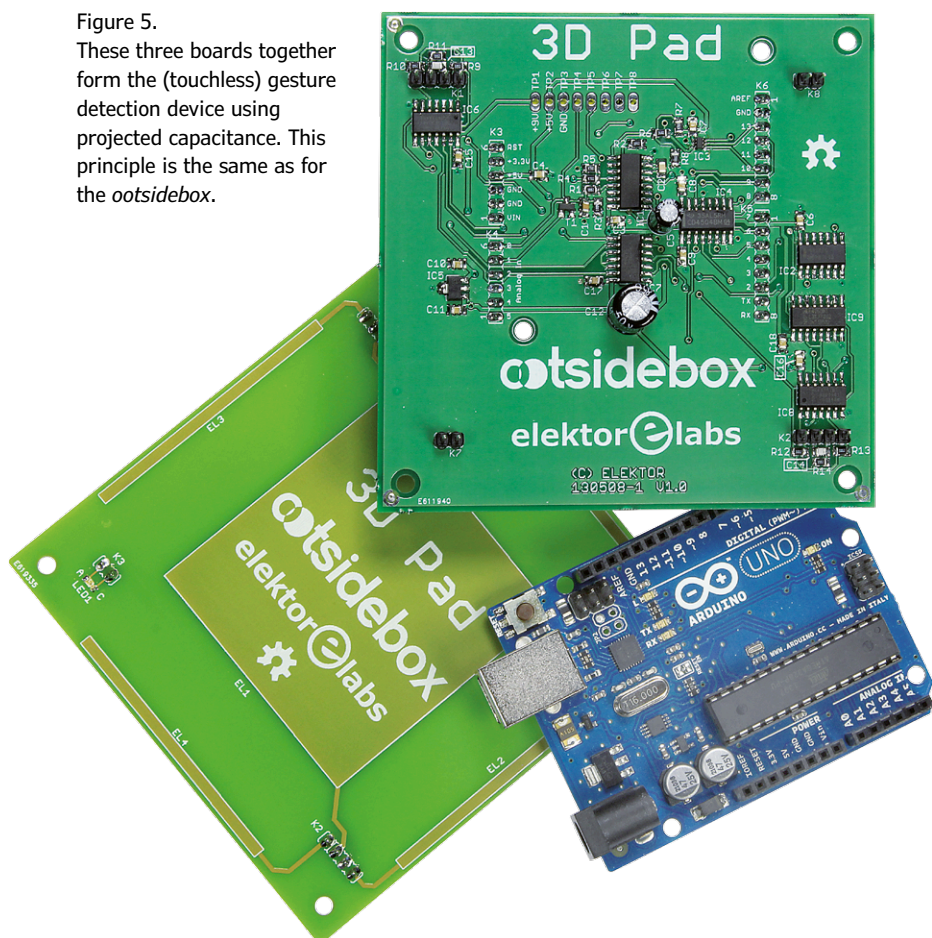
Figure 4. The scanning of each electrode generates a sawtooth whose amplitude is an indication of the capacitance detection.



- a reference oscillator
- a phase/frequency (P/F for short) comparator
- a control and locking program.

Have you already seen that somewhere before? Yes, these elements, often used for phase-locked loops (PLLs), are found all together in the 4046, a prodigious elderly integrated circuit, well known

Figure 5. These three boards together form the (touchless) gesture detection device using projected capacitance. This principle is the same as for the *outsidebox*.



and widely available. Here, it's not being used as a PLL, but let's see in the block diagram (**Figure 2**) how the 4046's voltage-controlled oscillator (VCO) and P/F comparator are used. The latter compares the signal from the VCO, which is going to be our reference oscillator, with the one from the electrode oscillator, whose frequency is brought down by the prescaler into a range compatible with that of the VCO, i.e. a few hundred kHz. Proven over decades, this simple system allows us to assess the frequency variations in the oscillator formed by the system of electrodes against the VCO frequency. The conversion of the (small) frequency shift into a variation in a voltage signal will be easy to use later. The VCO frequency is adjusted by way of a digital/analog converter (DAC) controlled by Arduino via the SPI port.

Now let's see how the scanning sequence runs for each of the electrodes:

- Initial situation: electrode oscillator at rest (no EN_X signal active), VCO inhibited, no oscillation. P/F comparator output is zero.
- A set-point value is applied to the VCO such that its nominal frequency is close to that of the electrode oscillator. This set-point stays the same for the rest of the sequence.
- The two oscillators are unblocked and we let the analog signal coming from the P/F comparator change for around 300 µs (i.e. around 30 cycles at 100 kHz) and the scan is stopped so as to capture the analog value by launching an acquisition on Arduino; then the oscillators are blocked again.

The result of this scanning sequence (on one electrode) at the output of the P/F comparator is

an analog signal, a sort of sawtooth (**Figure 4a**), whose exact shape and above all maximum amplitude are a direct function of the frequency difference between the two oscillators. A similar sequence will be produced for each of the receiving electrodes, starting the electrode oscillator with the corresponding `EN_X` signal and applying to the VCO each time the set-point appropriate for this electrode (**Figure 4b**). Obviously, it's the software that determines this set-point.

In the block diagram (figure 2), the signal called "injection", coming from the electrode oscillator prior to prescaling, is sent to the VCO to perform what we call "injection locking" [4]: when an oscillator injects a small amount of energy into another, it tends to drive the second oscillator when certain conditions are met (in particular, their nominal frequencies must be fairly close). The driving oscillator tends to impose its own frequency on the other oscillator. The principle also works on a harmonic, i.e. a multiple of the fundamental frequency. For our 3D-Pad, the effect of this is to influence the phase relationship between the two oscillators, and thereby the shape of the waveform at the P/F comparator output. The advantage lies in terms of the signal-to-noise ratio and the robustness against EMC interference – this is crucial, given the extreme sensitivity of our system.

So even though it is achieved using logic gates, my detector in fact delivers an analog signal. Maybe I'm a bit old-fashioned, but seeing this analog signal directly on the oscilloscope is for me a decided advantage.

When you test this circuit for yourself, once you've built it, here's what one of the electrode signals will give:

- Start by holding your hand far enough away from the 3D-Pad (at least 6 inches / 15 cm): the software has stabilized the analog value at the end of the sequence at 0.7 V.
- Bring your hand closer to the electrodes: the sawtooth amplitude changes from 0.7 V (@ 5 inches / 10 cm) to nearly 5 V @ 0.4 inches / 1 cm!

Our 3D-Pad comprises three boards (**Figure 5**): the Arduino UNO (in version R3 or higher), the main 3D-Pad *shield* board, and lastly the electrode plane. Their different functions are clearly indicated in the block diagram (Figure 2).

Circuit

Now you know all about how it works, here are a few interesting details about the circuit (Figure 3). If you change the geometry of the electrodes by even a tiny amount, or even the technology used to produce the PCB (the influence of stray capacitances is not negligible), you'll certainly have to adjust the values of resistors R9–R14, which let you adjust the electrode oscillator frequencies (IC6, IC7, IC8). If you reduce their value, the frequency goes up: you need to get as close as possible to 100 kHz. Capacitors C13 and C14 are not fitted.

Buffers IC9A and IC9B (4050) adapt the voltage level from the oscillator, powered at 9 V, to that of the downstream components, powered at 5 V. The counter IC2 (4024) acts as a prescaler for the `OSC_SENSE1` signal and supplies a lower frequency (`SIGIN`) to the P/F comparator IC1. Its Q4 output gives a frequency divided by 16. The signal from its Q3 output (`SYNC_SCOPE`) is sent to one input on Arduino. This counter can also (and will) be reset to zero by the (`\ENOSC_SENSE`) signal from the Arduino microcontroller.

The VCO and P/F comparator functions are found together in the same 4046 IC (IC1); their two end frequencies are determined by R2, R5, and C2. The electrode oscillator frequency (prior to prescaling) is injected by R4, as per the injection

I'm not going to go into details about getting started with Arduino. If you need to familiarize yourself with this versatile tool (as I did before this project), I recommend the following books: *Arduino* by Günter Spanner and *Mastering Microcontrollers with the Help of Arduino* by Clemens Valens [5]. The latter saved me a great deal of time and now has pride of place within easy reach, just next to my soldering iron. And not forgetting the other Elektor articles that have already been published on the subject.



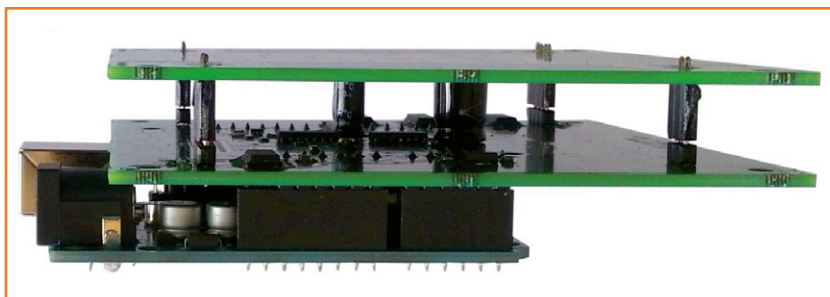


Figure 6. The three boards assembled.

Figure 7. An extract from the code of the `_3Dpad_sensor` sketch, with (on the right) the data sequences corresponding to the movements detected by the four electrodes.

```

_3Dpad_sensor | Arduino 1.0.5
Fichier Edition Croquis Outils Aide

_3Dpad_sensor
//.....
// Elektor Labs - 00T5IDEBOX project
//
// Touchless - Gesture 3D-pad
//
// JN Lefebvre 10 Mars 2014
// https://twitter.com/umshymot
// https://twitter.com/Outsidebox
//.....

#include <SPI.h>
#include <EEPROM.h>
#include <Automate.h>
#include <SendValueByUART.h>
#include <I2Cdef.h>
#include <Primitive@par8.h>
#include <GestionDonneeFlash.h>
#include <Regulation.h>

struct _Automate AutomateData;

void Run(void);
void SetupPct(void);
unsigned int EL[NODE]; //tableau des consignes VCO
unsigned int VAL[NODE]; //tableau des valeurs resultantes
unsigned int FILTER[NODE][FILTERLENGTH];
unsigned int VALF[NODE];

unsigned int Node;
unsigned int ConsigneValue=CONSIGNE;

unsigned int BaseTempRegulation=BTREGUL;
unsigned char Prox=0;
unsigned char OldProx;
unsigned char IntoField=0;

void Run(void);
void Filter(void);
void Regulation(unsigned int BaseTempRegulation, unsigned char Touch);

```

```

COM11
>IN 0221 0323 0623 0239 <
>IN 0222 0321 0621 0243 <
>IN 0223 0321 0619 0248 <
>IN 0222 0320 0619 0248 <
>IN 0220 0320 0621 0244 <
>IN 0217 0321 0624 0240 <
>IN 0216 0321 0626 0236 <
>IN 0216 0321 0626 0236 <
>IN 0220 0322 0625 0238 <
>IN 0220 0319 0622 0243 <
>IN 0221 0320 0621 0247 <
>IN 0222 0320 0622 0251 <
>IN 0222 0320 0623 0249 <
>IN 0220 0323 0626 0245 <
>IN 0219 0323 0628 0240 <
>IN 0219 0323 0631 0236 <
>IN 0220 0323 0629 0237 <
>IN 0223 0324 0628 0239 <
>IN 0226 0322 0627 0243 <
>IN 0228 0322 0629 0247 <
>IN 0228 0322 0630 0249 <
>IN 0228 0324 0633 0245 <
>IN 0225 0324 0635 0241 <
>IN 0224 0325 0638 0237 <
>IN 0224 0325 0638 0237 <
>IN 0227 0327 0637 0240 <
>IN 0228 0325 0636 0245 <
>IN 0230 0325 0635 0250 <
>IN 0230 0325 0636 0254 <
>IN 0229 0325 0636 0252 <
>IN 0224 0328 0637 0247 <
>IN 0222 0328 0640 0243 <
>IN 0222 0328 0643 0239 <
>IN 0222 0330 0641 0240 <
>IN 0226 0331 0639 0245 <
>IN 0226 0328 0637 0249 <
>IN 0226 0328 0638 0253 <
>IN 0225 0327 0638 0254 <
>IN 0223 0328 0640 0248 <
>IN 0221 0329 0641 0245 <
>IN 0218 0328 0642 0239 <
IN 02 02

```

Téléversement terminé
Taille binaire du croquis : 6 478 octets (d'un max de 32 256 octets)

locking principle mentioned above. The VCO's set-point voltage comes from IC3, a DAC8311 14-bit DAC, itself controlled by Arduino. The resistor network R6, R7, and R8 that the DAC output voltage passes through before reaching the VCO set-point input makes it possible to avoid the 4046's loss thresholds—i.e. a minimum voltage below which the frequency hardly reduces any further, and a maximum voltage above which the frequency does not increase any further, or hardly. The P/F comparator (IC1) sees at its CIN input the signal from the VCO and on its SIGIN input the Q4 output from the prescaler. The analog voltage is additionally cleaned by the network

R1, R3, and C2, with its rapid discharge achieved by T1, itself driven by an Arduino output (SCPF_CLEAR). On TP4, we have the usable analog value (CPF_OUT), both for observing operation on an oscilloscope and for the Arduino software, after analog/digital conversion.

For the electrode oscillator, we need 9 V, stabilized by a linear regulator (IC5), itself powered from a DC voltage of at least 12 V. This passes via the Arduino, which we'll be powering from a small 12 V DC power supply (Arduino UNO has provision for this). IC4 is a step-up level-changer (4504) which allows the Arduino controller's 5 V outputs to control the electrode selection, even though the electrode oscillator is powered at 9 V. The connection between our shield and the Arduino UNO board is made via K3, K4, K5, and K6.

Construction

As all the drawings for the UNO board are accessible online, it's not beyond the bounds of possibility for you to build it yourself. But I suggest you'd do better to buy a ready-built and tested UNO board [6]. The same goes for the two double-sided 3D-Pad boards – the track layouts are available on the Elektor website, but I'd recommend ordering them from the ElektorPCBservice [7]. A few details deserve special attention, like capacitors C5 and C12, which must be less than 6 mm high, otherwise you'll have to fit them lying down. You'll need an iron suitable for soldering SMD components, and you'll have to take particular care soldering IC3: the DAC8311 is so small, you need a magnifying glass to spot the pin 1 identifier.

The electrode plane connectors must be fitted on the underside of the PCB, so that the screen-printing on this board is on the top once it is plugged onto the 3D-Pad shield board (Figure 6).

The case must be plastic, not metal. If you're lucky enough to own a 3D printer, or have a Fab-Lab near you, you could have fun designing an original, futuristic case for it. I used a FIBOX ABS 125/35 LT case (RS ref. 498-4306) which has a transparent lid.

Software

The software (sketch) to be downloaded into the Arduino UNO consists of a file named `_3Dpad_sensor.ino` and a library `ElektorLabs3DPad`, to be installed into the Arduino IDE in the usual way. This program sends the data to the terminal via

the USB port, which is used in COM port mode (**fig. 7**). The data sequence contains the four electrode measurement values:

```
>IN/OUT  EL_Gauche  EL_Droite
EL_Haut  EL_Bas    <
```

IN/OUT indicates when the hand is perceived as being (IN) the detection field or not (OUT). The program on the UNO board has two main functions: on the one hand, the scanning and sequential measurement of the electrodes, on the other, the regulation, i.e. maintaining the operating point, taking the two reference electrodes into account. The principle is the same as for Touch detectors, e.g. the ones from Atmel. The level is regulated outside the detection window, which makes it possible to overcome slow variations (mainly caused by temperature changes); then when the detection is active, this regulation is usually blocked. In our case, this implies principally that the VCO set-points for each electrode measurement are continuously being corrected [8]. I can't go into a detailed description of the software here – it would take a whole article at least as long again as this one. However, here are the successive states of the 3D-Pad:

Self-calibration: (when first brought into service or on command via a serial link) the system seeks the operating point for each electrode, then memorizes the set-points into EEPROM on the Arduino.

Setup: each time it is powered up, or on command via a serial link, the system quickly (less than one second) seeks the operating point, starting from the set-points read from the EEPROM. There are also automatic configuration conditions, e.g. in the event of saturation of the electrodes.

Run: the normal operating state, which always follows on from a successful configuration.

The software's other tasks are:

Interpolating the co-ordinates: from the measurement values of the top, bottom, left and right electrodes, it calculates 3D co-ordinates: X, Y, and Z.

Gesture recognition: it recognizes swipes in all four directions (upwards, downwards, left, and

right), rotating movements with turns counting and detection of the rotation direction, and pushing a virtual button (Push).

And for testing, with the `_3Dpad_test.ino` file loaded instead of `_3Dpad_sensor.ino`, we have commands that allow us to activate the oscillators continuously, by choosing the electrode to be used. This lets us check the frequencies and adjust the resistors R9–R14 if necessary. There is also a command to make the VCO work with these set-points: 0, Max and ½ (i.e., at the DAC output, 0 = 0 V, ½ = 2.5 V and Max = 5 V).

As a bonus, I am offering a very simple but fairly comprehensive application program, for PC under Windows (XP and 7): this displays the 3D co-ordinates in the form of a cursor whose position on the screen reflects X and Y, while the Z axis is represented by the diameter of the cursor; it also displays the words "Air Swipes" when it detects a sideways sweep of the hand along one of the four axes, or the word "Push" when you make the movement of pushing a button (Z axis); and lastly, it indicates the rotation direction of the hand or finger it detects and the number of turns (very handy for carrying out settings).

It will be easy for you to draw inspiration from this program in Visual Basic 6 for your own applications.

Set-up

Your Arduino UNO board is operational and you have installed the software into the Arduino IDE. You'll need a voltmeter, an oscilloscope, and a frequency meter (the latter is unnecessary if you can measure frequencies with the 'scope). Once the components have been correctly soldered, I recommend the usual checks: orientation of the integrated circuits and polarized components, examination of the soldering using a magnifier (especially for the DAC IC3).

The tension mounts: Plug together all the boards to form the stack of PCBs (**Figure 6**) then plug the 12 V DC supply into the Arduino UNO external power jack. It's a good idea to check the power rail voltages: 12 V on K3-1, 9 V on TP1, and 5 V on TP2.

All OK? Then upload and run `_3Dpad_test.ino`. Open the connection with the terminal set to 115,200 baud; a menu of the commands available appears.

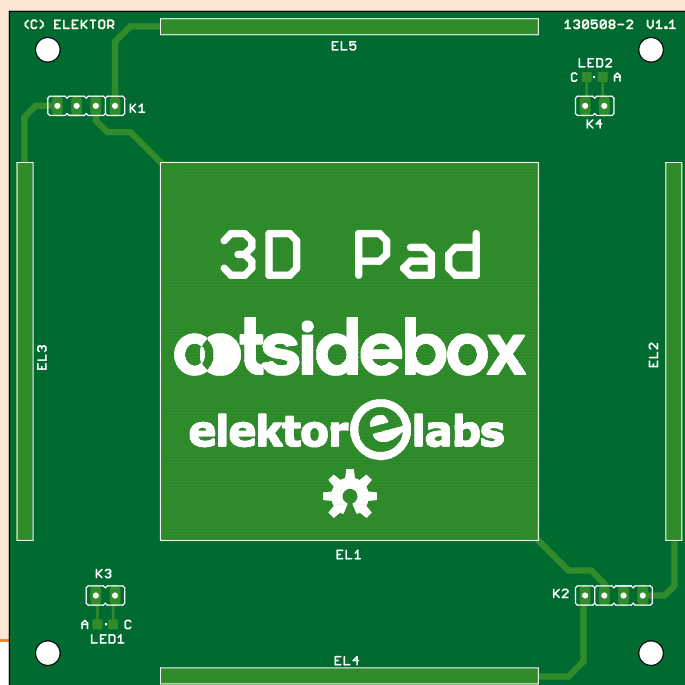
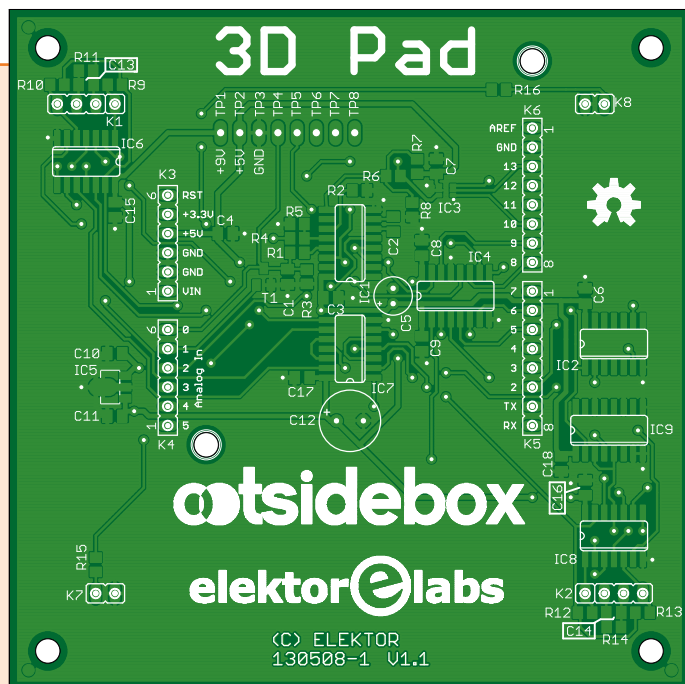
Let's check the oscillators are working:

VCO: enter the command to activate the VCO oscillator with a set-point $\frac{1}{2}$: you should see 2.5 V on TP8 and a nice square wave on TP5 at a frequency of 100–110 kHz (depending on component tolerances).

Electrodes: make sure you clear everything away from around the electrode plane. For each of the six electrodes, use the terminal to send,

via the serial link, the corresponding command (e.g. “T” = top electrode, or “B” = bottom one) and check the oscillation signal on TP6. The frequency should likewise be close to 100–110 kHz. The other commands are documented in the software source code.

If these checks are satisfactory, load the normal program `_3Dpad_sensor.ino` in place of the test



Component List

3D-Pad Arduino shield

Resistors

SMD 0805, 0.125 W)
 R1, R6, R7 = 10k Ω
 R2, R8 = 18k Ω
 R3 = 1M Ω
 R9, R12 = 47k Ω
 R4 = 120k Ω
 R5 = 470k Ω
 R10, R13 = 82k Ω
 R11, R14 = 22k Ω
 R15, R16 = 470 Ω

Capacitors

default: SMD 0805
 C1 = 2.2nF
 C2 = 220pF
 C3, C4, C6, C7, C8, C9, C10, C11, C15, C16, C17, C18 = 100nF
 C5 = 4.7 μ F 16V (pitch 2mm)
 C12 = 100 μ F 16V (pitch 3.5mm)
 C13, C14 = not fitted

Semiconductors

IC1 = HEF4046BT
 IC2 = CD74HC4024M
 IC3 = DAC8311 ou AD5641AKSZ
 IC4 = CD4504BM
 IC5 = 78L09 (SOT-89)
 IC6, IC8 = HEF4011BT
 IC7 = HEF4081BT
 IC9 = CD74HC4050M
 T1 = 2N7002

Miscellaneous

K5, K6 = 8-pin pinheader*
 K3, K4 = 6-pin pinheader *
 K1, K2 = 4-pin pinheader *
 K7, K8 = 2-pin pinheader

Electrode Plane

Semiconductors

LED1, LED2 = LED, green, SMD, Kingbright type KPT-2012SGC

Miscellaneous

K1, K2 = 4-pin pinheader*
 K3, K4 = 2-pin pinheader*
 PCB # 130508-21

* 0.1" pitch



Serial Inventor and Entrepreneur

Jean-Noël Lefebvre learnt electronics with his soldering iron in his hand and through reading magazines and specialist books (including Elektor, naturally) The 3D-Pad circuit described here is based on his own patents. A "DIY" enthusiast, he fully supports the concept of *HackerSpaces* and *Fab-Labs*, in association with the *Makers* movement.

Jean-Noël has been devoting his time for a year now to OOTSIDEBOX, a start-up that is going to be offering devices like the 3D-PAD, and also an accessory for Android tablets that enables them to be controlled by touchless gestures, identical to the 3D-Pad system. This

project will shortly be launched on the Indiegogo crowdfunding website. You can participate in his project and help him. Follow @ootsidebox on Twitter: twitter.com/Ootsidebox or post a LIKE on Facebook: www.facebook.com/ootsidebox

en.wikipedia.org/wiki/Hackerspace
edutechwiki.unige.ch/en/Fab_lab
en.wikipedia.org/wiki/Maker_culture
www.ootsidebox.com
www.indiegogo.com



program. A good way of checking the 3D-Pad in operation consists in connecting a 'scope probe to TP8 (DAC output) and another to TP4. The sync is taken from TP8. At first switch-on, or following the self-calibration command (send "A" from the terminal), the voltage level on TP8 will fall gently: this is the search for the set-points for each electrode. After a few moments, this trace should be stabilized (end of self-calibration) and your oscilloscope screen should look like the one shown in Figure 4b. Bring your hand closer to the electrode plane: you should see the sawtooth on TP4 change and also the measurement values

scrolling on the terminal [9]. You can launch a reconfiguration at any time, either with the command "R" from the terminal, or by saturating the electrodes by bring your hand flat within a few millimeters of the electrode plane.

I'll be there to help you if you need help getting it going or information for a specific application, or if you have any suggestions for developing the project. Don't hesitate to contact me on Twitter [10], I'll be delighted to answer you personally.

(130508)

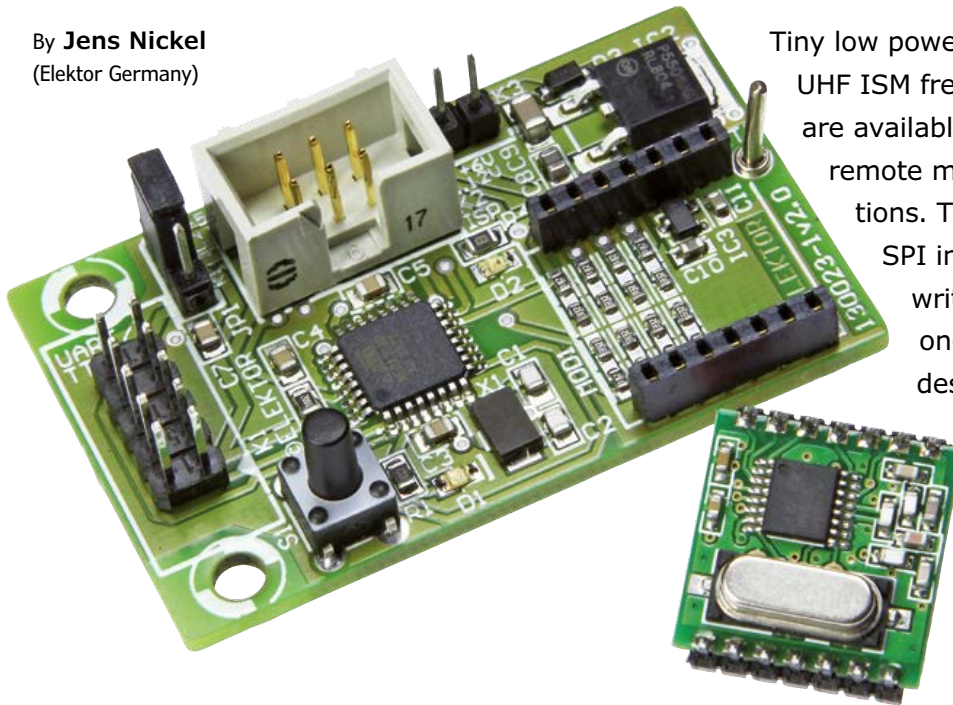
Web Links

- [1] games
 Fruit Ninja: <http://fruitninja.com/>
 Despicable Me - Minion Rush: <https://play.google.com/>
- [2] CC BY-NC-SA 4.0 <http://creativecommons.org/licenses/by-nc-sa/4.0/>
- [3] www.embedded.com/design/prototyping-and-development/4008781/Getting-in-touch-with-capacitance-sensor-algorithms
- [4] injection locking: http://en.wikipedia.org/wiki/Injection_locking
- [5] www.elektor.com/arduino
- [6] assembled, tested UNO board from the Elektor e-shop
www.elektor.com/arduino
- [7] www.elektorPCBservice.com
- [8] www.rtcmagazine.com/articles/view/101626
- [9] video of the sawtooth signal on TP4 <http://youtu.be/rYdyR49qFzU>
 demo video: <http://youtu.be/11QGUXXFYq8>
- [10] @junowhynot + #3DpadElektor: <https://twitter.com/junowhynot>

Wireless Gateways

From serial to radio, and back again

By **Jens Nickel**
(Elektor Germany)



Tiny low power radio (LPR) modules operating at UHF ISM frequencies like 433 MHz and 868 MHz are available on the market that are ideal for remote measurement and control applications. They are usually controlled over an SPI interface. Poring over datasheets and writing a driver is, however, not everyone's delight, and we have therefore designed a small board that carries an LPR module along with an ATmega328 microcontroller and a serial interface. Pre-programmed firmware allows the board to be used as a simple gateway for strings of characters between the serial interface and the ether.

FCC (USA) and ETSI (European) approved and certified LPR modules (**Figure 1**) operating at 3.3 V are available from the Chinese manufacturer HopeRF. Also known as SRDs (short range devices) they are based on the Silicon Labs Si4421 [1] and include the necessary support circuitry and a quartz crystal. The IC makes it relatively straightforward for advanced embedded systems developers to add wireless data transfer to their projects. The transceiver is controlled over an SPI interface: specified byte sequences act as commands that switch the device between transmit and receive modes, alter the data rate, and so on. The SPI interface is also used to transfer characters to be transmitted and characters that have been received.

A designer with enough experience and enough patience could write the necessary firmware to drive the SPI interface, and, on top of that, a small Si4421 driver library to assemble the command byte sequences for the device and send them to it. In the simplest case such a library would provide a function to allow the sending of character strings generated by the application

program over the wireless link; a further function could be provided to store received characters in a circular buffer for subsequent processing by the application program.

Shift those strings

Although example microcontroller-specific SPI drivers and Si4421 libraries can be found on the Internet, adapting these to your own firmware and project will take some time. In this article we take a different route, similar to that adopted with considerable success by FTDI for its USB interface drivers. FTDI's USB-to-TTL converter ICs are popular chiefly because they are so easy to interface to the serial ports on microcontrollers. The designer is freed from having to do any special USB driver programming, and only has to worry about sending and receiving characters using the UART.

Our board is designed so that you can simply plug in a HopeRF wireless module. As well as a power supply, the board includes an ATmega328 microcontroller and a 2-by-5 header that carries the UART RX and TX signals (**Figure 2**). The pin-

out is based on the EEC specification presented in Elektor [2].

Thanks to its firmware the 8-bit microcontroller on the board looks after all driving of the radio module. Elektor supply the board with pre-programmed example firmware, turning the board into a gateway between a UART and the "ether". If a string (of up to 62 characters) is sent to the gateway, terminated by a <CR> character, it will be sent over the wireless link. In the reverse direction a string received over the wireless link is output over the serial interface: the data rate for serial communication in either direction is set at 9600 baud.

The gateway board is based on a design by Elektor author Günter Gerold, who developed his module for telemetry applications for his 'Wheelie'. In the Elektor labs we designed a printed circuit board, which is available either populated or unpopulated [3]. In the populated version the microcontroller comes ready-programmed with the gateway software, but you can of course replace this with your own code. Based in Europe, Elektor Labs used Hope RF's 433 MHz version of the LPR module.

Circuit

The circuit diagram (Figure 3) is not particularly complicated. The central component is the ATmega328 microcontroller, which can be flashed with new firmware over ISP connector K2. The crystal and power supply circuits follow the stan-

dard datasheet arrangement. A LED and a button are connected to port pins PD4 and PD5. These provide a minimal user interface that can come in handy during firmware development on the board, as well as in test and in actual use.

The RFM12B wireless module is connected to the microcontroller over a total of eight wires. Four



Figure 1. The low-power radio module from Chinese manufacturer HopeRF (433 MHz version shown here with header pins soldered on) is controlled over SPI. Versions are also available for the 868 MHz SRD band.

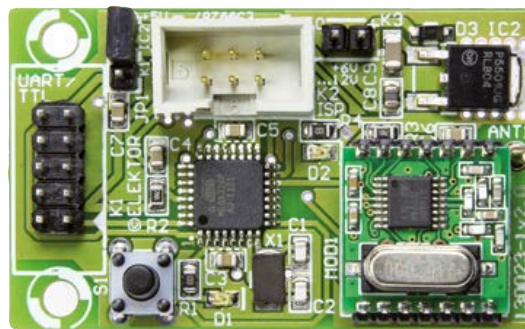


Figure 2. The gateway board with ATmega328, button, LED and a two-by-five connector for serial signals.

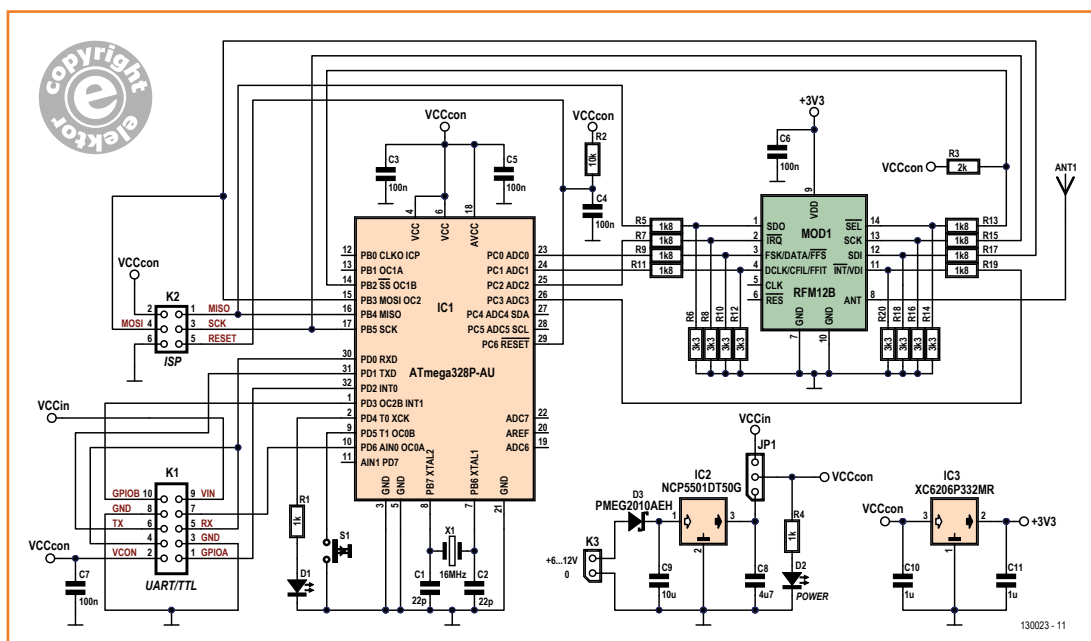


Figure 3. The microcontroller is connected to the wireless module over a total of eight wires. Five of these are used by the pre-programmed firmware.

Component List

Resistors

R1,R4 = 1kΩ (0805)
 R2 = 10kΩ (0805)
 R3 = 2kΩ (0805)
 R5,R7,R9,R11,R13,R15,R17,R19 = 1.8kΩ (0603)
 R6,R8,R10,R12,R14,R16,R18,R20 = 3.3kΩ (1%, 0603)

Capacitors

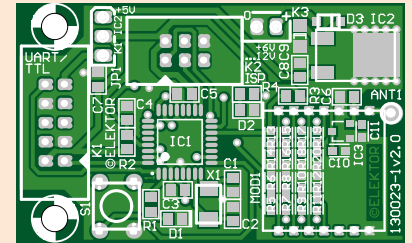
C1,C2 = 22pF (0805)
 C3..C7 = 100nF (0805)
 C8 = 4.7μF (0805)
 C9 = 10μF, 25V (1206)
 C10,C11 = 1μF (0603)

Semiconductors

D1 = LED yellow (0805)
 D2 = LED green (0805)
 D3 = PMEG2010AEH Schottky diode (SOD-123F)
 IC1 = ATmega328P-AU, programmed (TQFP-32)
 IC2 = LDO NCP5501DT50G (DPAK-3)
 IC3 = XC6206P332MR (SOT-23-3)

Miscellaneous

K1 = 10-pin (2x5) pinheader, 0.1" pitch
 K2 = 6-pin (2x3) pinheader, 0.1" pitch
 K3 = 2-pin pinheader, 0.1" pitch
 JP1 = 3-pin pinheader, 0.1" pitch with jumper
 S1 = pushbutton



X1 = 16MHz quartz crystal, 50ppm, 18pF
 MOD1 = RFM12B-433-S1 radio module (HopeRF)
 14-pin pinheader, 2mm lead pitch
 14-way receptacle. 2mm pitch
 PCB # 130023-1 v2.0



of these comprise the SPI interface mentioned above, and these are connected directly to the hardware SPI port on the ATmega328. This is the same interface as the microcontroller uses when it is being programmed: pull-up resistor R3

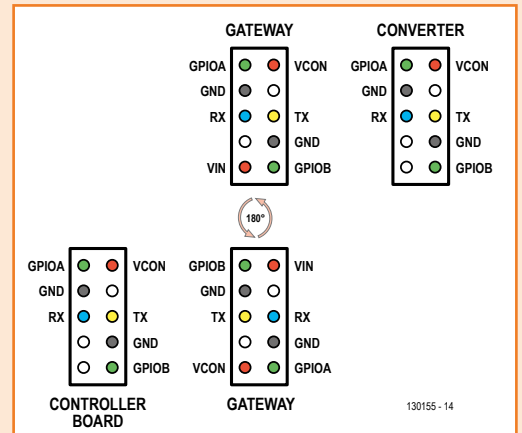
takes the module's select input /SEL high during programming, so that the wireless module does not try to interpret the bytes being transferred. When the microcontroller wants to send bytes to the radio module it must pull this signal low

ECC: connector specification for serial signals

The Embedded Communication Connector (ECC) employed again in this project, is a two-by-five pinheader that carries serial signals. As well as the expected RX and TX we have two general-purpose digital signals or GPIOs (green in the figure). In the March 2014 edition we described a small RS-485 interface adapter that can be connected to a microcontroller board via the ECC. In the planning stage we also have an RS-232 interface adapter as well as modules for USB, WLAN and Bluetooth.

The pin at the top right allows a converter board to be supplied with power from the microcontroller board. In contrast to a 'dumb' converter board, a gateway such as the one described in this article contains a microcontroller with software running on it. The ECC makes it possible to connect such a gateway to a converter as easily as to a microcontroller board. In the former case the gateway can provide the converter with power over the VCON pin; and similarly the gateway can itself be powered from a microcontroller board via the VIN pin. If a ribbon cable is used to connect the two boards together, the plug at the gateway end should be turned through 180 degrees so that the VCON pin of the microcontroller board connects to the VIN pin of the gateway. Conveniently this also happens to swap over the RX and TX signals, so that the microcontroller on the main board and the gateway can communicate with one another. The latter case is probably the standard application—our Gateway effectively equipping the main controller board with a radio link. Consequently pin 1 of the ECC points inwards on the component overlay, so inwards is where the 'nose' of the flat ribbon connector should point also. On all controller boards pin 1 of the ECC connector (and the connector nose) points outwards. Attention: due to the flip around (swapping TX and RX) the pin numbering is not conclusive about the pin function; pin 1 for instance can only be GPIOA once or GPIOB once.

The pins shown in white are uncommitted, and can be used for special purposes. In the project described here we have used one of them as an extra connection for the RX signal and connected the other to a general-purpose port pin of the ATmega328.



in software via port pin PB2.

The pre-programmed software also makes use of the FFIT pin on the wireless module. The signals /FFS, /INT and /IRQ are connected to port pins on the microcontroller as well in order to allow you to make more flexible use of the wireless module in your own code.

K1 is the ECC connector mentioned above. The middle pair of pins provides access to the RX and TX signals of the microcontroller's UART, while the other pins of the connector are taken to port pins PD2, PD3 and PD6 of the microcontroller. These could be used for flow control on the serial port or for dedicated functions.

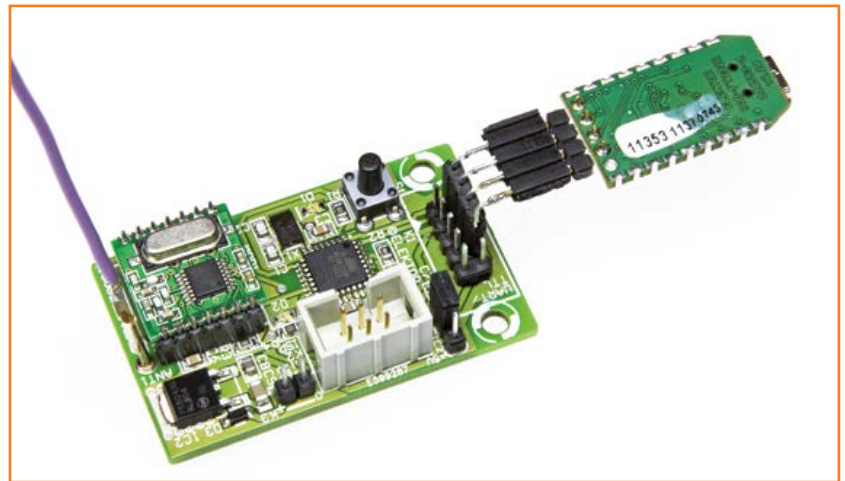
The lower left pin is connected to the 5 V microcontroller supply. The gateway board can supply a small current over this pin, sufficient for an adapter to convert the TTL-level serial signals to RS-232 or RS-485. An RS-485 adapter for connection to the ECC was presented in the March 2014 edition [2].

The top right pin, conversely, allows the gateway board to be supplied with power over the connector, for example from a motherboard. Alternatively the board can be powered via K3 and voltage regulator IC2: the power source is selected using jumper JP1. Green LED D2 indicates when power is present.

The gateway's 5-V operating voltage is used by a small voltage converter to generate a 3.3-V supply for a radio module; at the same time the 5-V signals are converted down to 3.3 V and vice versa. Thanks to the use of SMD components the gateway board is very compact and the artwork is always available for downloading from the project page set up for this article. There you can also find your ready populated and tested Gateway board together with radio module type RFM12B-433. If you use a different radio module, do make sure you have a 3.3-V version. Although there are surplus stocks around, the 5-V version of the radio module is no longer manufactured. On ANT1 connect a suitable piece of stiff wire around 17 cm long (quarter wavelength at 433 MHz).

433 MHz, 868 MHz or ?? MHz?

That depends on the country you live in, bearing in mind that Elektor is published worldwide in English. One or more of the above frequency ranges may be allocated to ISM (industrial/scientific/medical) applications allowing the use of type-approved low-power radio modules by pri-



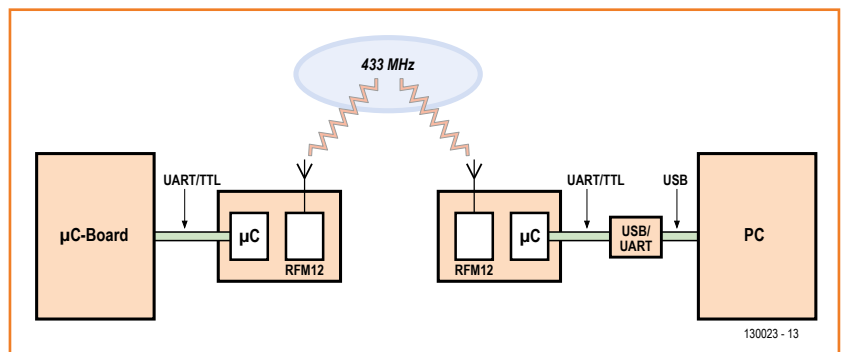
vate individuals within the limits of national, state, or regional legislation.

Applications

The ECC leaves two uncommitted pins for use in special applications (shown in white in the figure in the text box). We have routed the UART RX signal to one of these pins so that RX, TX and ground are available on the connector as three adjacent pins in a straight line. As a result the BOB USB-to-serial converter [3] can be connected directly to the board for test purposes or even in a final project: see **Figure 4**. An FTDI 5 V USB-to-serial cable (also available from Elektor) can be fitted, but the pins in the socket need to be moved around so that the pinout matches [4]. Connect the USB adapter to a PC. Connect a second wireless module to a microcontroller board with RX, TX and ground pins available: connect TX on the microcontroller to RX on the ECC and *vice versa*. You can now send strings of up to 62 characters from the PC to the microcontroller board (**Figure 5**). At the PC end you can use a simple

Figure 4. A BOB USB-to-serial converter can be connected directly to the gateway.

Figure 5. One possible usage scenario for the gateway: strings can be transmitted to and fro between a PC and a microcontroller board.



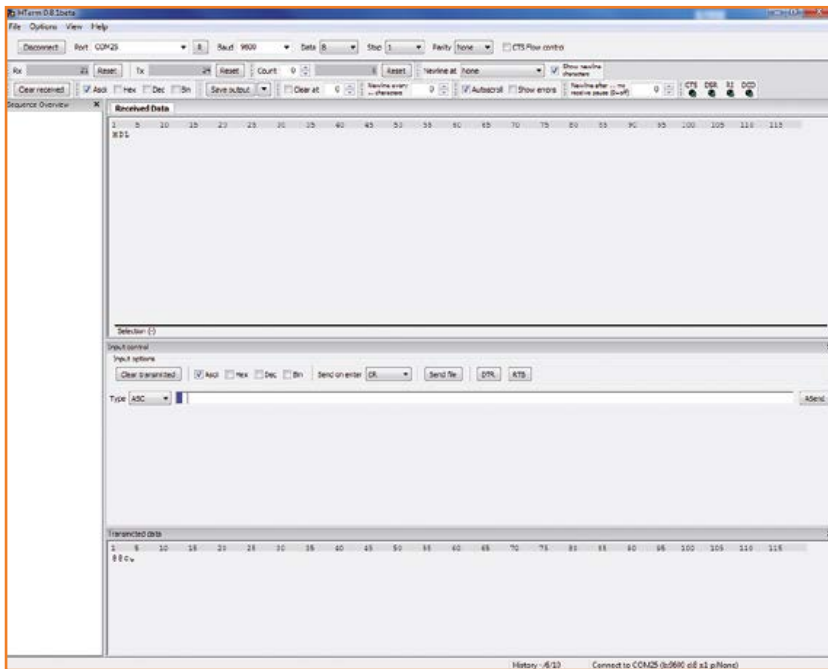


Figure 6. A terminal program such as HTerm comes in handy when testing the radio gateway.

terminal program such as HTerm, which can be downloaded from the Internet free of charge [5]. HTerm should be configured so that a complete line of characters is sent when the 'Return' key on the keyboard is pressed, with a <CR> (ASCII 13) character appended. To send data in the opposite direction you will need to arrange for the microcontroller to output a string using its UART, again terminated by a <CR> character.

The ready-programmed version of the wireless gateway has two modes: 'send' and 'listen'. When power is applied the gateway enters send mode, with the yellow LED not illuminated. In this mode the board will accept strings over its serial connection and send them out over the air, but will not act as a receiver for data. Press the button, and the gateway will switch to listen mode, indicated by the LED lighting. The board will now listen for characters sent over the ether. When a <CR> character is received the entire string is output over the serial interface.

Special commands

A gateway can also be switched between its two modes of operation by sending the sequence '@@c<CR>' to its serial interface (the 'c' stands for 'change'). The command '@@s<CR>' ('s' for 'swap') causes both gateways to switch between one mode and the other. However, the second gateway only changes mode when it receives a

string over the air that indicates that the first gateway has switched to listen mode. A gateway connected over its serial interface responds to the two special commands with the acknowledgement strings 'MDL' ('mode listen') or 'MDS' ('mode send'). In this way we receive confirmation of which mode the gateway is now in and can proceed to test whether the connection is working via the serial interface (Figure 6).

Assuming that only a unidirectional connection is wanted, then initially the remote module will be set up in listen mode and strings can be sent to it. The direction of communication can be reversed at any time by sending the command '@@s<CR>' to the gateway currently in send mode. The gateway forwards this command to the other end of the link, as a message which could be interpreted as meaning 'I have finished transmitting and am now switching to listen mode and will wait for your transmissions'.

This behavior does not cover all possible application requirements, of course. For example, if a sensor board wants to send regular readings to a control system running on a PC, then the module connected to the PC will have to be left in listen mode. However, this means that the PC is unable to send out commands, such as to tell the sensor board to change the interval between readings. The gateway software therefore includes the possibility that both ends of the connection are in listen mode. Despite being in listen mode, characters can be sent to the serial interface on either board to be sent over the wireless link. For technical reasons, in this mode it is necessary to terminate the string to be sent with two <CR> characters (that is, press the Return key twice in the terminal program). In this bidirectional configuration either participant can be sending or receiving at any time and it is up to the application software (running on the PC or on the microcontroller board) to ensure that collisions do not occur.

Software

The firmware that is programmed into the ready-made boards is of course available as a source-code download from the Elektor website in the form of an Atmel Studio 6 project [3]. The code is written in a modular fashion, based on the Embedded Firmware Library (EFL) [6]. The microcontroller file for the ATmega328 sits at the bottom of the software stack, exposing to higher levels functions such as

```
uint8 SPIMaster_TransceiveByte(int8 Handle, uint8 Databyte)
```

which sends a character over the SPI interface. This particular function is used by the low-level driver for the wireless module, which is located in the board file (BoardEFL.c). The low-level driver provides the following functions:

```
void WirelessModule_Command(uint8 WirelessBlockIndex, uint16 Command)
void WirelessModule_SendData(uint8 WirelessBlockIndex,
    uint8* DataBuffer, uint8 DataLength)
void WirelessModule_ReceiveData(uint8 WirelessBlockIndex)
```

In this case `WirelessBlockIndex` is always zero, although in principle it would be possible to connect more than one wireless module to the microcontroller. The `WirelessModule_Command()` function can be used to send a two-byte command to the wireless module as specified in its datasheet [2]. This function simply calls the lower-level function `SPIMaster_TransceiveByte()` twice while holding the /SEL pin low.

The function `WirelessModule_SendData()` is responsible for sending characters. The function first sends the byte `0xB8` to the wireless module; subsequent bytes are then transmitted. The developers of the wireless module specify that a two-byte synchronization pattern (`0x2DD4`) should be sent before the payload data to improve the reliability of transmission. After all the payload bytes for transmission have been transferred over the SPI bus, the function appends a byte `CHAR_WIRELESS_ENDOFTRANSMISSION` to the end of the string: in `BoardEFL.h` this is defined to take the value `0x04`.

The function `WirelessModule_ReceiveData()` is repeatedly called in listen mode. Its most important component is a loop which continuously receives characters. At the head of this loop is a small nested loop in which the FFIT pin is checked. When this pin is high it means that the wireless module has received a character and written it to its internal FIFO buffer. The function now takes /SEL low and sends the command `0xB000` to the module to fetch the received character. The code then returns to checking the FFIT pin. The wireless module starts to write characters into its FIFO as soon as it sees the synchronization pattern.

The characters received are written to a circular buffer dedicated to the wireless module. Like all circular buffers in the EFL it has a default size of 64 bytes. When the character `0x04` is received the receive function exits its main loop. Note that the `0x04` character is not written to the circular buffer. By clearing a flag in a specified register the wireless module is instructed not to write further characters into its FIFO until the synchronization pattern is seen again. This will occur at the start of the next received string.

The receive loop is also exited when the button is pressed or if a byte is received over the serial interface: this allows the bidirectional function described above to be implemented.

Configuring the wireless module

We have now covered all the functions that are used to control the wireless module. As usual for the EFL, the functions are written in a hardware-independent fashion and can be used on other boards with different wiring. The wiring configuration of the gateway board, including its button and LED, are defined in the function `Board_Init()` in the board file [6], which will need to be changed suitably to use the code with different hardware.

Before we can send and receive characters we need to configure the wireless module. What command bytes do we have to send to the board using the func-

BATRONIX

UNBEATABLE

at price-performance ratio. The new Rigol measurement devices



Rigol DS1000Z Oscilloscopes

4 channels, 70/100 MHz, 1 GSa/s sample rate, 12 million measurement points memory, USB, LAN, professional measure & analyse features, opt. built-in waveform generator, 3 years warranty

from € 450,- net
incl. EU wide free shipping



Rigol DG1000Z Waveform Generators

2 channels, 30/60 MHz, 200 MSa/s, 2 Mpts memory, USB, LAN, LXI, analog and digital modulation, arbitrary signal generation, 3 years warranty

from € 467,- net
incl. EU wide free shipping

Make your **LIVE** easier.
Leading **TECHNOLOGY**
with **BATRONIX** satisfaction-
guarantee

- ✓ Attractive prices
- ✓ Expert advice
- ✓ Large selection in stock
- ✓ 30 day trial period
- ✓ Money back guarantee
- ✓ EU wide free shipping for most products

Use our special offers now:
www.batronix.com/go/32

NEW

tion `WirelessModule_Command()` to achieve this? The only way to find out involves painstaking study of the module's datasheet. For that reason we have written the library module `WirelessInterfaceEFL`. If the function `WirelessInterface_LibrarySetup()` is called when an application starts up it will carry out the basic configuration of the wireless module, for example setting its data rate to 9600 baud. The individual commands are commented in the code, including a cross-reference to the relevant section of the datasheet where you can find a more detailed explanation of what is going on.

The function `WirelessInterface_Send()` configures the module for transmission, activating the TX register and the transmitter. This function should only be called immediately before actually sending data in order to conserve power.

The function `WirelessInterface_Listen()` switches the wireless module into FIFO mode, which, among other things, means that the FFIT pin mentioned above will signal the arrival of a new character. The receiver is also enabled.

One-to-one gateway

The remaining piece of the puzzle is a mechanism which in listen mode will periodically read the character received over the air from the circular buffer where they are stored and send them out over the serial interface, and, in the opposite direction, take characters received over the serial interface from the other circular buffer where they are stored (by the UART interrupt code) and transmit them. These tasks are carried out by the library `OneToOneGatewayEFL`, which was described in a previous article [7]. The library inspects the two circular buffers in turn to see if a <CR> character has been received. If so, the characters received up to that point are sent out over the appropriate channel in one go. The `OneToOneGatewayEFL` library has been extended for this project to handle the special commands, which always start with the sequence '@@' and

which are three bytes long. A call to the function `OneToOneGateway_SetSpecialCommandFunction()` is used to tell the library what callback is to be called when a special command is received over the communications channel. In our case we specify the function `SpecialCommandGateway()`, which is implemented in the main part of the code. The parameter to this callback function is the third character of the command, here 'c' or 's'. Depending on which character is received the gateway will switch between its listen and send modes. The command is also sent over the wireless link to the other gateway module: this behavior can be configured using the second parameter to `OneToOneGateway_SetSpecialCommandFunction()`, which we have set to 'TRUE'.

Testing

To test the gateways we can connect two units, each equipped with a BOB or FTDI cable, to two USB sockets on the same host computer. It is now possible to open two copies of the terminal program and send characters to and fro.

If you enter the special command '@@<CR>' in both terminal windows both gateways will independently switch to listen mode. Each wireless module should reply with the string 'MDL' ('mode listen') over its serial interface, and hence in its terminal window. This means that serial communication is working correctly: if this test fails, it sometimes helps to try disconnecting and reconnecting the USB-to-serial adapter.

You can now try sending short messages from one terminal to the other. Don't forget that you need to press the Return key twice at the end of each message!

(130023)

Note:

the firmware running in Elektor programmed microcontroller is purely experimental. The publishers cannot guarantee error-free operation in all situations.

Web Links

[1] www.silabs.com/Support%20Documents/TechnicalDocs/Si4421.pdf

[2] www.elektor-magazine.com/130155

[3] www.elektor-magazine.com/130023

[4] www.elektor-magazine.com/080213

[5] www.der-hammer.info/terminal/ (site in German)

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

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

[8] www.elektor-labs.com/ecc

[9] www.elektor-magazine.com/120126

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

DLP-USB1232H: USB 2.0 UART/FIFO
HIGH-SPEED
480Mb/s



Only \$28.95!

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

DLP-IO8-G
8-Channel Data Acquisition



Only \$29.95!

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

DLP-IOR4
4-Channel Relay Cable

DLP-TH1b
Temp/Humidity Cable

DLP-RFID1
HF RFID Reader/Writer

DLP-FPGA
USB-to-Xilinx FPGA Module



www.dlpdesign.com

SCOPE DEALS

6-IN-1 SCOPE

2-ch 2MSa/s 200kHz 11-bit oscilloscope, 2-ch spectrum analyzer, 2MSa/s 8-bit arbitrary waveform generator, function gen, network analyzer, PWM/Digital I/O. - CGM-101 **\$99.95**



PASSPORT-SIZE PC SCOPES

Great scopes for field use with laptops. Up to 200MHz bandwidth with 1GSa/s, high speed data streaming to 1MSa/s, built-in 1GSa/s AWG/wfm gen. - PS2200A **\$239+**



30MHz SCOPE

Remarkable 30MHz, 2-ch 250MS/s sample rate oscilloscope. 8-in color TFT-LCD and AutoScale function. Includes **FREE** carry case + 3 yr warranty! - SDS5032E **\$289**



60MHz SCOPE

Best selling 60MHz 2-ch scope with 500MSa/s rate + huge 10MSa memory! 8" color TFT-LCD. Includes **FREE** carry case! - SDS8062 **\$349**



100MHz SCOPE

High-end 100MHz 2-ch 1GSa/s benchscope with 1MSa memory and USB port + **FREE** scope carry case. New super low price! - DS1102E **\$399**



INCREDIBLE LOW PRICES, FREE TECHNICAL SUPPORT
GREAT CUSTOMER SERVICE SAELIG.COM

Take out a FREE membership to Elektor.POST

- The latest on electronics and information technology
- Videos, hints, tips, offers and more
- Exclusive bi-weekly project for GREEN and GOLD members only
- Elektor behind the scenes
- In your email inbox each Friday



Register today at www.elektor.com/newsletter

Intelligent Cuelight System for Theaters

Act III, Sc. II: *Storm still, Enter Lear and Fool*

By John Baraclough
(UK)



A cue light is a system of one or more electric light bulbs (usually traffic light colors) used to allow silent cues to be given to technicians and performers at various working locations during the running of a show or play. Typically, the Deputy Stage Manager sends signals to these cue lights at pre-arranged times. In this article we present a low cost microcontroller-driven version, hence “Intelligent”.

Typically, traditional cuelight systems have an “Acknowledge” button that allows feedback on the status of the person at the cue point to the Cue Light Operator. Although headset systems have made cue lights less popular, they are still used in some cases where silence is necessary, or where a headset is not practical.

There are multiple protocols that are used to designate the meaning of each light. Green is usually used to signal a “Go” cue. An optional yellow light or a flashing red generally means “Stand By.” A solid red light can indicate that the standby has been acknowledged, or that no cue is pending, depending upon the protocol used and hopefully

learned by the actors and prop assistants. No light at all can represent that no cue is pending. An alternate scheme with only one lamp uses 'on' as a standby cue and 'off' as the cue.

Why build a cuelight system

when there are commercial products available like [1],[2]. Most of these systems contain multiple channels, each channel having its own Standby, Go and Clear buttons. Meaning a lot of buttons in front of the (Deputy) Stage Manager. The author

designed a system with a different approach: he uses up to four banks (called 'buses') of Standby and Go buttons, and one 'Master' Clear button (or individual Clear for one or more channels).

In principle there's no limit to the number of channel boards in the system—every channel has four DPST switches (S1–S4) to connect it to up to four buses. Every channel board controls one 'remote' PCB with a red and a green LED and an optional signaling button.

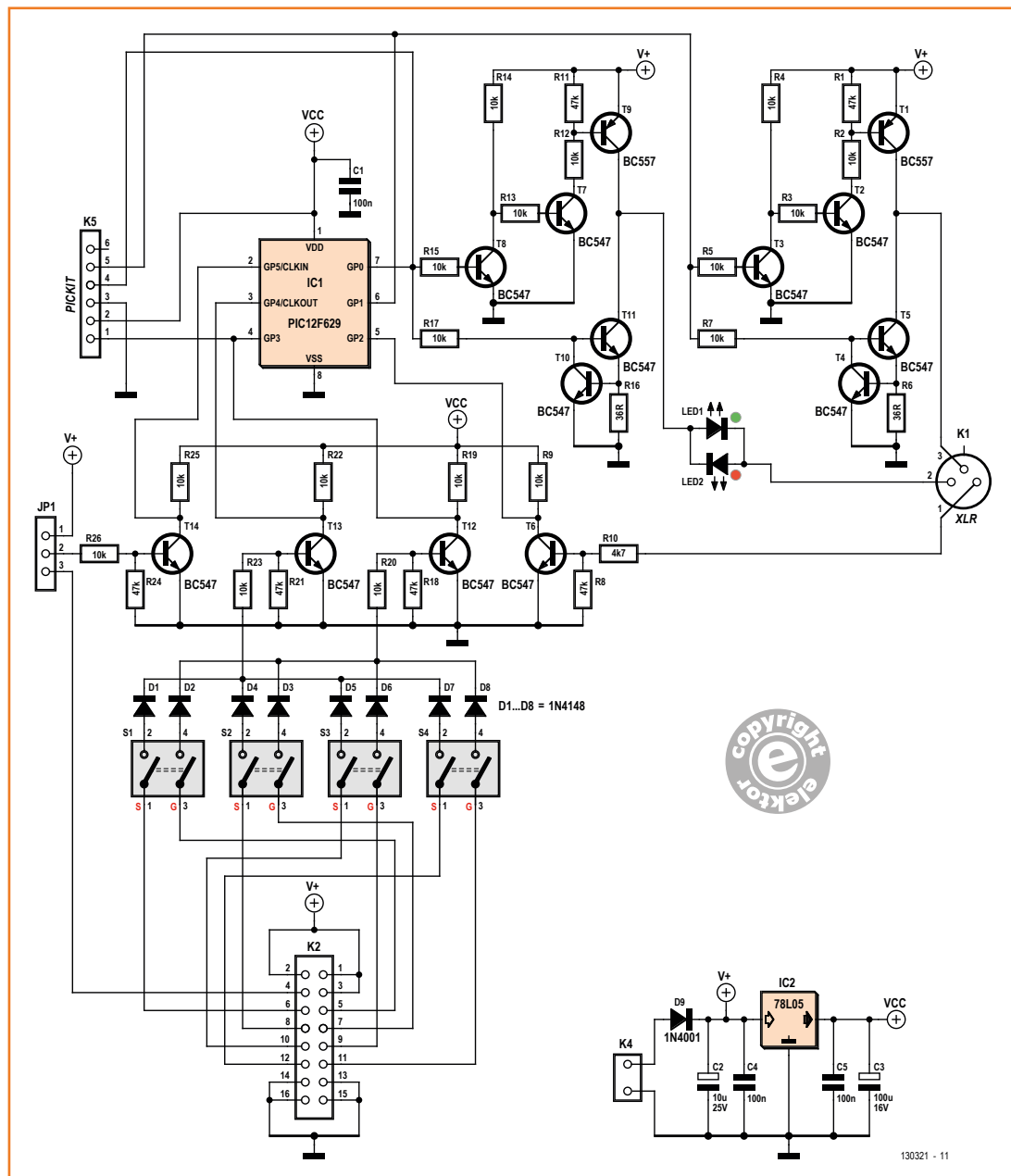


Figure 1. Schematic of the Channel board. Intelligence supplementary to that of the (Deputy) Stage Manager comes from a PIC micro. Communication with other Channel boards is through connector K2.

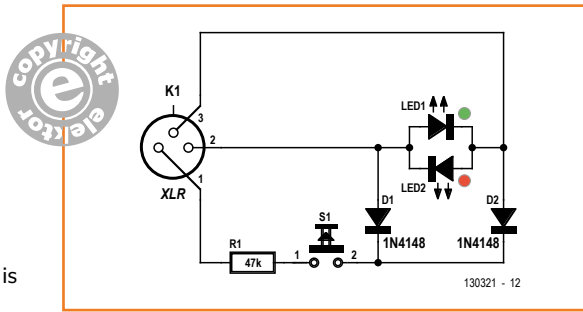


Figure 2. The Slave (Remote) board is entirely passive.

tion and just expect the appropriate people to react correctly at the right time. With less experienced cast members and stage crew it is definitely an advantage to have the “Go” light stay on until the SM can see that the people have responded. With the cast and crew in our local Community Theatre it certainly makes a difference having the “Go” light on for a longer period. Enabling the “Clear” function means that the SM doesn’t have to hold down the “Go” button until he or she sees some action on stage! In the original prototype one of the ADC channels was used to read an external pot which controlled the on-time delay of the “Go” LED after the “Go” button was released. The mock-up was shown it to a couple of retired stage managers living near

To clear or not to clear...

The option of using the “Clear” button is down to the preference of the Stage Manager (SM). With professional casts and stage crews he/she would probably not use the “Clear” func-

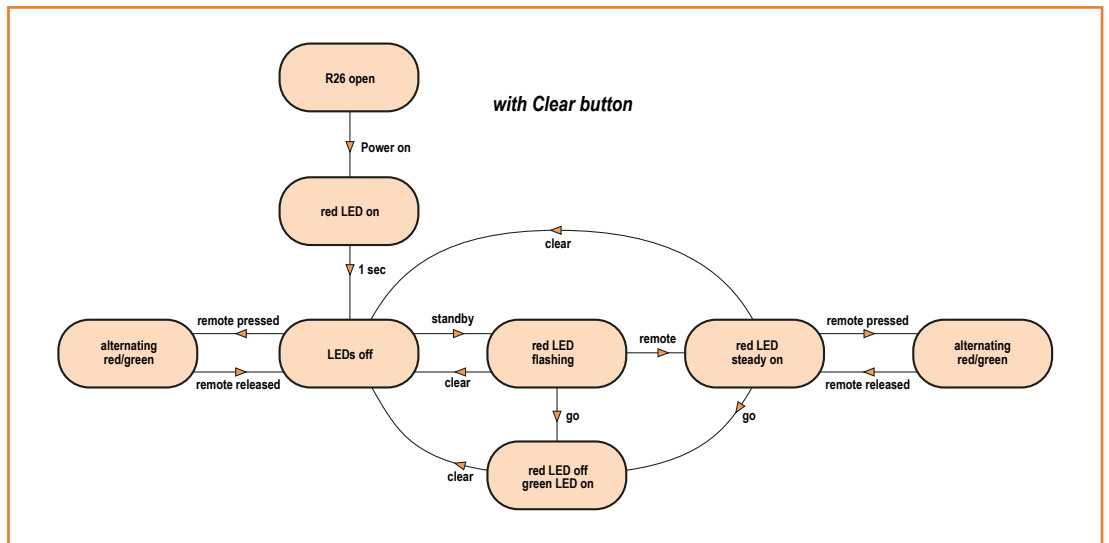


Figure 3. State diagram for “Clear-included” mode of operation.

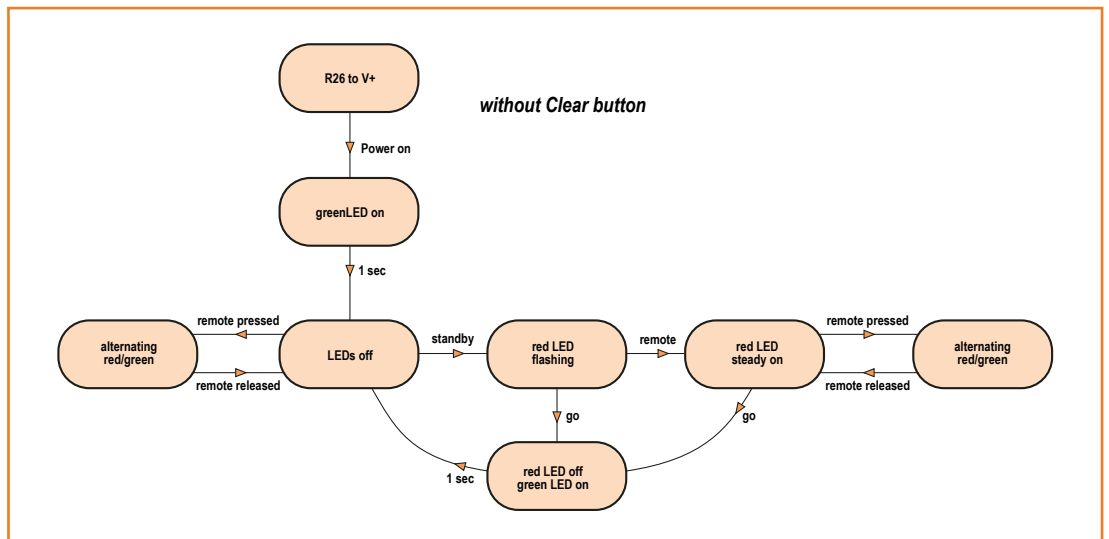


Figure 4. State diagram for “Clear-not-included” mode of operation.

the author and both said that they didn't think that function was a very useful feature, and having the option of keeping the "Go" LED on and using a "Clear" button would be much better. It was easily changed with a simple software modification so was built-in. The difference between the PIC12F615 and -629 is the absence of A/D. In the first prototype the -615 was needed.

Circuit description; software creeping in

The schematic of the Intelligent Cuelight System comes in two parts: **Figure 1** for the Channel board, and the much smaller **Figure 2** for the Slave board.

Pin 2 on JP1 (effectively pin 2 on the PIC) is the "Clear" input of a channel board. It is read by the software at startup and the subsequent behavior depends upon its level. If the pin is tied to V+ at startup with a jumper across pins 1 & 2 (i.e. pin 2 on IC2 effectively pulled Low) the software will ignore that pin from then onwards, and the green

"GO" LED will extinguish a second or so after the assigned "GO" button is released. If the jumper pin is floating at startup (i.e. pin 2 on IC2 pulled High) then the "CLEAR" function is activated. This can be done in either with a "CLEAR" button per channel connected between pins 1 & 2 on JP1 or with a common "CLEAR" button connected between V+ and pin 4 on the ribbon cable bus on K2. In the latter case a jumper between pins 2 & 3 on JP1 is required.

Instead of using links an SPDT switch can be connected to the jumper JP1. In one position it will connect pins 1 & 2, disabling the "CLEAR" function. In the other position it will connect pins 2 & 3 on JP1, allowing the common bus line to be used for the "CLEAR" function. Two state diagrams show how a channel board and remote board work, depending on the setting of JP1 at power-up: with Clear, **Figure 3**, and without Clear, **Figure 4**.

In terms of the current source constellations T1-T5 and T7-T11, the design could have been

Figure 5. Component mounting plan for the Channel board.

Component List

Channel Board, no. 130321-1

Resistors

R1,R8,R11,R18,R21,R24 = 47kΩ 5% 250mW
 R2,R3,R4,R5,R7,R9,R12,R13,R14,R15,R17,R19,R20,R22,R23,R25,R26 = 10kΩ 5% 250mW
 R6,R16 = 36Ω 1% 250mW
 R10 = 4.7kΩ 5% 250mW

Capacitors

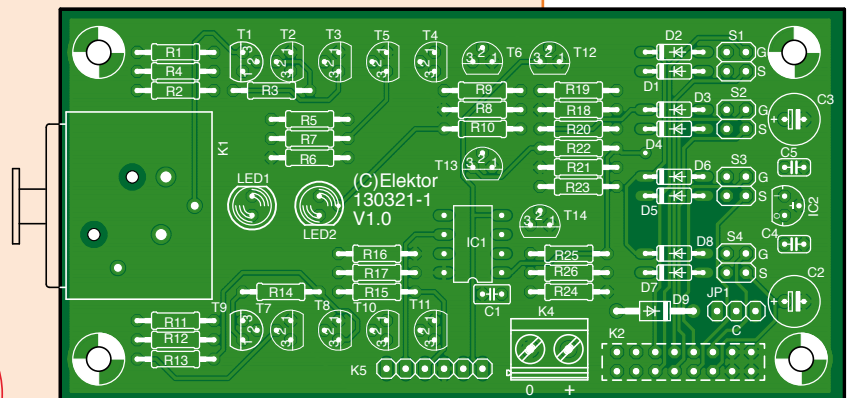
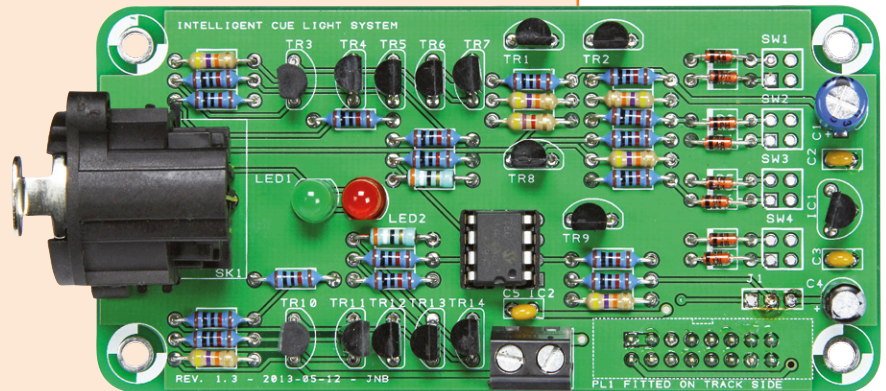
C1,C4,C5 = 100nF 50V
 C2 = 10μF 16V radial
 C3 = 100μF 25V radial

Semiconductors

D1,D2,D3,D4,D5,D6,D7,D8 = 1N4148
 D9 = 1N4001
 LED1 = low current, 5mm, green
 LED2 = low current, 5mm, red
 T1,T9 = BC557
 T2,T3,T4,T5,T6,T7,T8,T10,T11,T12,T13,T14 = BC547B
 IC1 = PIC12F629-I/P, programmed, Elektor Store # 130321-41 [3]
 IC2 = 78L05

Miscellaneous

K1 = 3-pin XLR socket, PCB mount
 K2 = 16-pin (2x8) boxheader, 0.1" pitch
 JP1 = 3-pin pinheader, 0.1" pitch
 K4 = 2-pin pinheader, 0.1" pitch
 K5 = 6-pin pinheader, 0.1" pitch
 PCB # 130321-1 [3]



Listing 1. Excerpt of assembly code (.asm) file

```

;
; Main code starts here
;
MAIN

    BSF    STATUS, RP0           ; Select Bank 1

    MOVLW TRISIO_DATA          ; Set port direction register
    MOVWF TRISIO

    MOVLW OPTION_DATA          ; Set OPTION register.
    MOVWF OPTION_REG

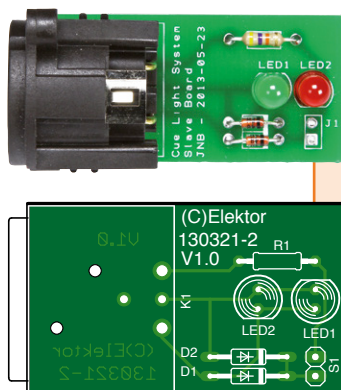
    MOVLW ANSEL_DATA           ; No analog inputs.
    MOVWF ANSEL

    BCF    STATUS, RP0         ; Select Bank 0

    MOVLW GPIO_DATA           ; Initialize GPIO.
    MOVWF GPIO

    CLR    FLAGS               ; Initialize the file registers.
    CLR    STATE_MACHINE
    CLR    ACK_BUT_REG
    CLR    GO_BUT_REG
    CLR    SBY_BUT_REG
    CLR    CLR_BUT_REG
    CLR    PULSE_RELOAD
    CLR    GPIO_VALUE_1
    
```

Figure 6.
Component mounting plan
for the Slave board.



a bit less complicated if the LEDs would simply be switched on and off using one bipolar transistor or FET. But the constant current ($\approx 0.6 \text{ V}/36 \Omega = 17 \text{ mA}$) drive concept used in this design actually only uses two extra transistors but has a couple of benefits. Firstly the length of the connections makes no difference to the LED brightness and secondly the power supply can be any-

thing which comes to hand. It could be as low as 9 V and as high as the high voltage limit for the 5-V regulator (usually about 36 V). For a portable system that's quite an advantage as, if the power supply is lost or fails, any handy one can be used as a replacement.

The complexity in the cable driver comes from the need to always have the supply voltage available

Component List

Slave Board, no. 130321-2

Resistor

R1 = 47k Ω 5% 250mW

Semiconductors

D1,D2 = 1N4148

LED1 = low current, 5mm, green
LED2 = low current, 5mm, red

Miscellaneous

K1 = 3-pin XLR plug, PCB mount
S1 = pushbutton
PCB # 130321-2 [3]




```

CLRF GPIO_VALUE_2
MOVLW TICKS_PER_50MS      ; Timer ticks for 50mS.
MOVWF TICK_COUNT

INT_ENABLE

MOVLW (1 << GIE)|(1 << T0IE) ; Enable Timer 0 & Global interrupts.
MOVWF INTCON

CHECK_CLEAR_BUTTON

BTFSC GPIO, CLR_BUT_BIT    ; Check if CLEAR button pressed (link is closed).
BSF  FLAGS, USE_CLR_BUT_FLAG ; Pin is high, so set the flag to use it.

MOVLW RED_LED_ON          ; Turn RED LED on.
BTFSS FLAGS, USE_CLR_BUT_FLAG ; If using CLEAR button then show RED LED.
MOVLW GREEN_LED_ON        ; Otherwise show GREEN LED.

MOVWF GPIO                ; Write value to LEDs.
MOVLW TWO_SECONDS         ; Show the appropriate LED for two seconds.
MOVWF PULSE_COUNT

BSF  FLAGS, TIMER_FLAG

CHECK_CLEAR_BUTTON_2

BTFSC FLAGS, TIMER_FLAG    ; Wait for timer to finish.
GOTO CHECK_CLEAR_BUTTON_2

MOVLW BOTH_LEDS_OFF        ; Turn off LEDs.
MOVWF GPIO

```

at the remote terminal for signaling. When the system is inactive the positive supply is routed to both legs of the LEDs so that pressing the Acknowledge button when no LEDs are lit still provides signaling capability (through the isolating diodes). One of the positive supplies must be switched off when the LEDs are lit, which requires the extra transistors.

For a completely wired channel board four DPDT switches would be needed for selection of the SB (standby) / GO bus(-es) it should respond to. Plus one SPDT for the Clear setting and (optional) one pushbutton if the channel must be cleared individually. All channel boards are interconnected with a 16-way flatcable on K2 (mounted on the bottom side of the PCB!), which not only connects the channels to the four SB/GO buses and the

Master Clear bus, but also links all boards to the power supply; only one board will have a direct link to the power supply on K4. In many practical situations not all switches will be needed and/or used, and the complete setup will depend on the performance, play, number of actors, theater, and so on.

Finally referring to the schematic in Figure 2, every channel has its Slave/Remote board, connected via a 3-wire XLR cable. The remote boards will only need a pushbutton S1 if the performers should acknowledge that they are ready or if they—for some reason—must be able to draw the attention of the Stage Manager (in cases of stage fright, collapsing stages and such...). If this button is not needed, resistor R1 and 1N4148 diodes D1 and D2 can also be omitted on the remote board.

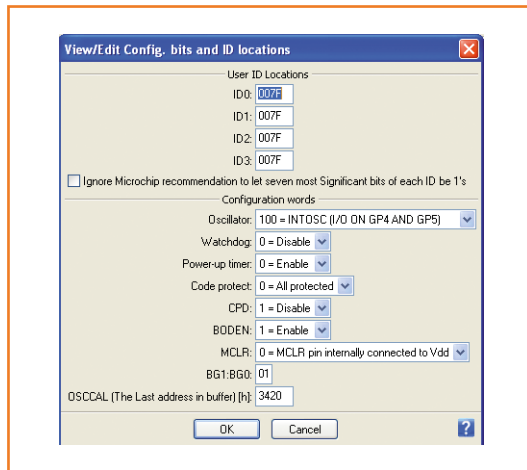


Figure 7. DIY programmers! Abide by these fuse settings and you're okay for burning your own PIC12F629(s) for the project.

Firmware

Every single channel board has a microcontroller (PIC12F629), ready programmed available from the Elektor Store as no. 130321-41. Another option is to buy blank microcontrollers from your local supplier, download the firmware or even the assembler code file from the Elektor Magazine website [3] *free of charge* and do the programming with a programming/debugging interface like the Microchip PICKit or ICD connected to K5. A code extract is shown in **Listing 1**, and the ever popular programming fuse settings are shown in the screendump in **Figure 7**.

(130321)

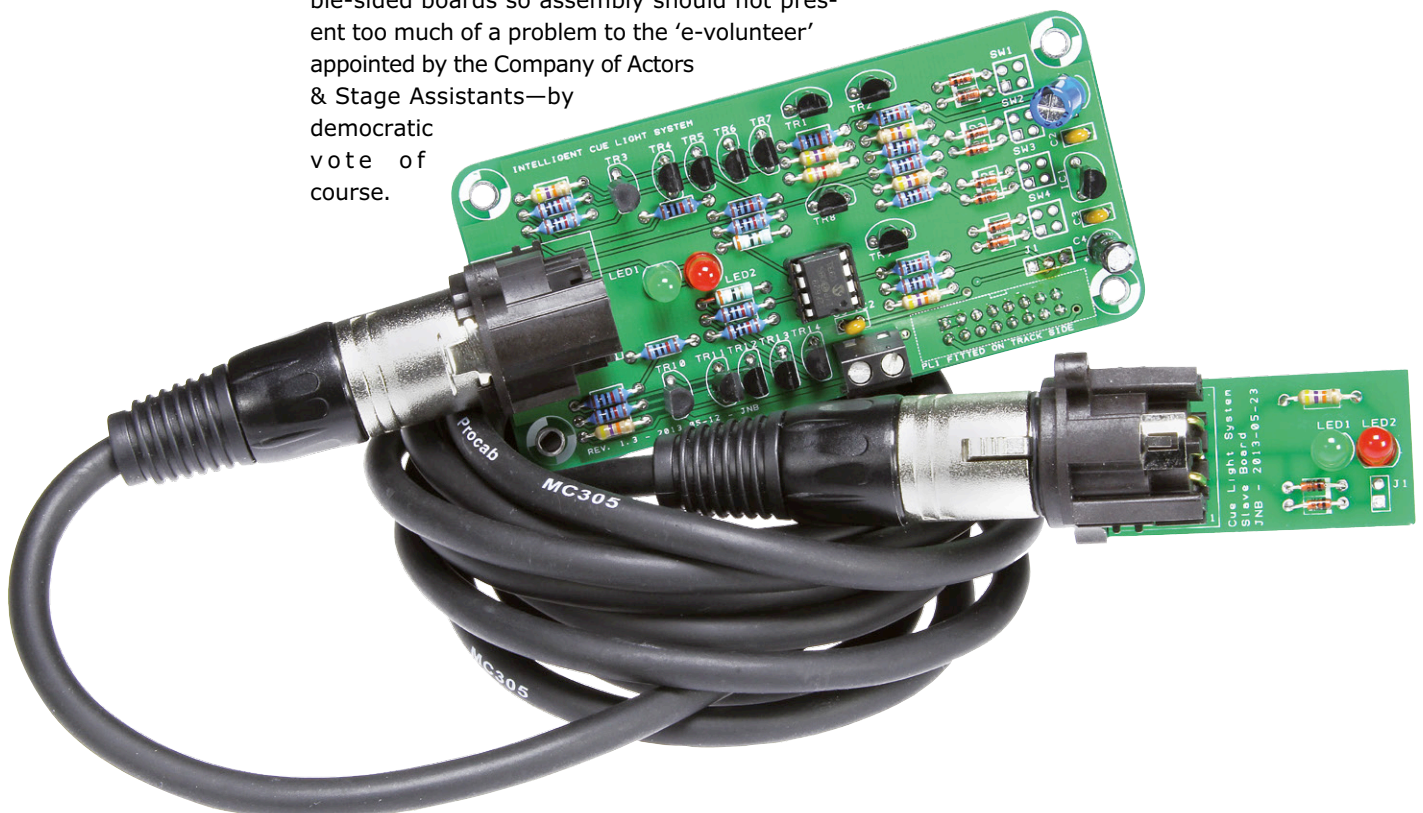
Web Links

- [1] www.orbitalsound.com/sales-paging-gds.asp
- [2] www.leonaudio.com.au/16_channel_cue_light_mk4.pdf
- [3] www.elektor-magazine.com/130321

In addition to that, there will be a maximum of four SB and four GO buttons connected to the buses, plus one Master Clear. All in all quite some wiring to do if one wants to build a complete system with a lot of channels.

Assembly

The system consists of one or multiple Channel boards and associated Slave boards. The board layouts designed by Elektor Labs are shown in **Figure 5** and **Figure 6** respectively. Here we're looking at plain old through-hole parts on double-sided boards so assembly should not present too much of a problem to the 'e-volunteer' appointed by the Company of Actors & Stage Assistants—by democratic vote of course.



DVD'ed: The Full Range of 2000-2009 Volumes of Elektor Magazine!

NEW!



**A Must-Have
for All Elektor
Fans!**

- Electronic Archive, Article PDFs, Quick Search Function
- 110 Elektor Editions, Over 2500 Articles, Easy Print Function
- Ideas, Circuits and Projects for Electronics at Work, in College, at Home

- at Work, in College, at Home
- Ideas, Circuits and Projects for Electronics
- Easy Print Function
- 110 Elektor Editions, Over 2500 Articles
- Quick Search Function
- Electronic Archive, Article PDFs

ISBN 978-1-907920-28-8
£77.95 • € 89.00 • US \$121.00

Order Today at www.elektor.com/zeroes

log inputs, so the corresponding terminals on the Uno board are labelled A0 to A5. Port pin PC5 is connected to A5 at the edge of the board diagonally opposite the USB connector, which makes it easy to find. That's actually why we chose this particular port pin for our first program. **Figure 2** shows all the pins of the ATmega328 with their official pin names (PC6, PD0, etc.), the special functions of individual pins (which we will discuss in due course), and the abbreviated Arduino pin names (at the outside edges).

After a restart or reset, every pin of port C is configured as a high-impedance CMOS input with the same properties as the familiar 4000 family of CMOS ICs. Accordingly, it's not a bad idea to use 100 kΩ resistors here for protection against excessive voltages and static discharges.

Protection diodes

All modern CMOS devices have a pair of protection diodes on each input, with the one connected to ground and the other to Vcc. Their job is to limit excessive voltages. The first CMOS ICs in the 4000 family, which came on the market a long time ago, did not have input protection diodes. Premature IC failures due to improper handling were therefore fairly common at that time. Static discharges from people to grounded objects at voltage levels from 100 to 1000 V happen all the time, but MOSFET gates don't like them at all and may break down at voltages as low as 50 V. Now that protection diodes are integrated in all ICs, failure due to static discharge breakdown has become rare. You can detect these protection diodes with an ordinary ohmmeter. This is often very useful because it is an easy way to check the connections to the pins of an IC in a circuit to see whether or not they are okay.

The protection diodes are what make it safe to touch an open CMOS input with your finger through a 100 kΩ resistor, despite the fact that your body may have static charges and carry induced AC voltages up to 100 V under no-load conditions. The human body is a sort of node with numerous capacitive links to all sorts of cables and power outlets. As audio designers and troubleshooters know, a finger is an excellent source of a hum signal. The protection diodes (**Figure 3**) limit the high-amplitude sinusoidal signal to a low-amplitude square wave (5 V peak to peak). You can see this for yourself by measuring the signal with an oscilloscope. Actually the amplitude of the square wave is 6.2 V peak to peak, since the protection

diodes only start conducting at -0.6 V and +5.6 V (assuming a supply voltage of 5 V).

Reading input states

The program *UNO_Input1.Bas* in **Listing 1** (the code can be downloaded from the Elektor website [2]) reads the state of the signal on port pin PC5 and outputs it to the LED on port pin PB5. First all pins of port C (PC0 to PC5) are initialized as digital inputs. In the interest of readability, the instruction *Config Portc = Input* is used for this purpose. Actually you could omit this instruction, since as previously mentioned port C is automatically configured with all pins as inputs after power-up.

ATmega microcontrollers have output registers that determine the states of the output port pins. When you program the software to write a particular value to an output register, the corresponding port pins are set to 0 or 1 according to the content of the register. There is a separate input register that holds the actual states of the pins. This register

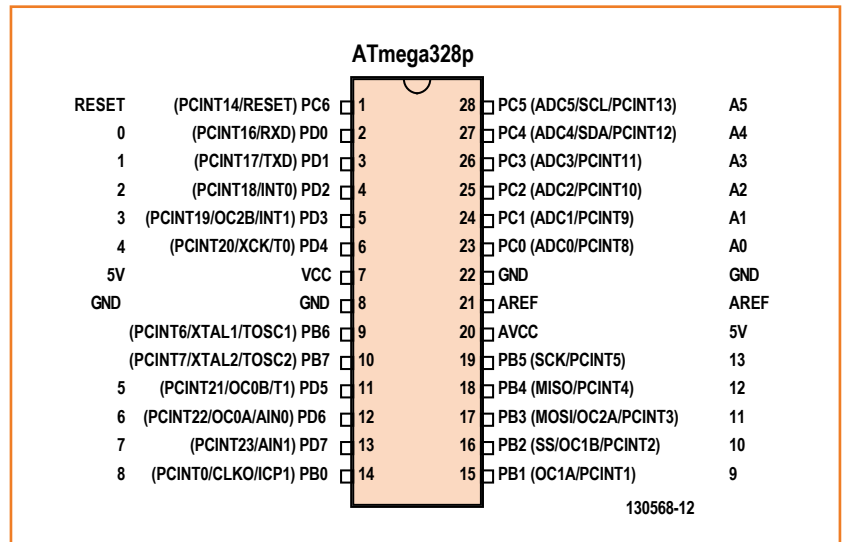


Figure 2. The ATmega328 pinout and the corresponding Arduino names.

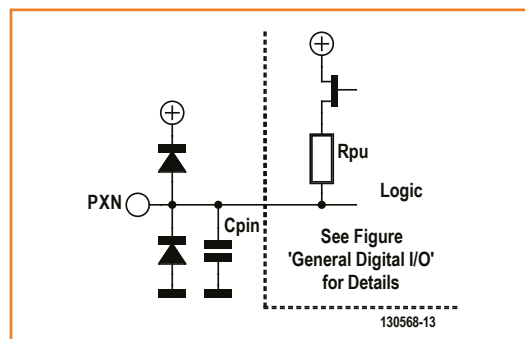


Figure 3. Input protection diodes.

Bits and Bytes

Each register in the AVR microcontroller holds eight bits, or one byte. A bit can only have the value 1 or 0. A byte (in a register or otherwise) consists of eight bits, so a byte register can hold values from 0 to 255 in decimal notation. If you write the values in binary notation instead, you can see the individual bit values directly. For example: 00000000, 11111111 or 100110010.

In many registers each bit is directly associated with a specific pin. One example is the PORTD register. Bit 7 (at the left end) determines whether port pin PD7 is high or low. Similarly, bit 0 (at the right end) determines the state of port pin PD0.

Many programming languages have instructions for setting or clearing individual bits in a register, which are accordingly called single-bit instructions. There are also instructions that can write a specific value to a register and therefore affect all of the bits at the same time.

To take an example from Bascom, *Portb.5 = 1* sets bit 5 of the eight-bit register PORTB to 1, which means that the signal level on PB5 is high. None of the other bits is affected, since this is a single-bit instruction. However, you can also access all bits of the PORTB register at the same time with the Bascom instruction *Portb = &B00100000*, where *&B* is the Bascom notation for a binary number. This has the same effect on bit 5 as the previous instruction: in both cases the yellow LED on the Arduino board lights up. However, the difference can certainly be significant for the other seven port pins. In the first case, any of these bits that were previously

set to 1 remain in that state, but in the second case they are all set to 0.

To take another example, in Listing 1 you could replace the line *Portb.5 = Pinc.5* with *Portb = Pinc*. If you only consider bit 5, you have the same circuit and the same result, but there's a big difference because now you're moving eight bits at a time instead of just one. As a result, any change on input PC0 will affect output PB0.

Along with binary notation (such as *&B00100000*), Bascom supports two other commonly used notations. Instead of *&B00100000*, you can write the value in hexadecimal notation, in which case the instruction takes the form *Portb = &H20*. The other alternative is decimal notation: *Portb = 32*.

The best way to understand how these different ways to represent numbers work is to consider the example of a four-bit number. The smallest possible value is 0000 (all bits off), and the largest possible value is 1111 (all bits on). In mathematical terms, this is called a base-2 number system. Each bit has a specific weight, which increases from right to left: 1, 2, 4, 8. Bit 0 has the weight $2^0 = 1$, bit 1 the weight $2^1 = 2$, bit 2 the weight $2^2 = 4$, and bit 3 the weight $2^3 = 8$. As you can see, the value doubles with each bit position. You can obtain the corresponding decimal number by adding up the values of the individual bits. In hexadecimal notation, the numbers 10 to 15 are represented by the letters A to F.

can be read by the software to determine whether the pins are high or low. Each register holds one byte (eight bits) with a value range from 0 to 255 (see the **Bits and Bytes** inset). Each bit shows

the state of a single port pin.

In the case of port B, the output register (write register) is called PORTB and the input register (read register) is called PINB. The software sees these registers as memory locations that can be read and written. However, in hardware terms they are physical circuits with external connections. There is another register for each port called the Data Direction Register, which allows the port pins to be configured as inputs or as outputs. For port B, this register is called DDRB. There is no obvious way to write data to this register in a Bascom program, since this operation is performed indirectly by the Config instruction. For example, *Config Portb = Output* sets all bits of DDRB to 1, which causes all pins of port B to act as outputs.

If you run the program and connect the input to ground, the LED will be off (dark). If you connect the input to +5 V, the LED will light up. As you can see, the input state is copied to the output. Now try something different: first make sure that you are standing or sitting or standing completely insulated from any conductive objects, and then touch the input with your finger. Remarkably

Listing 1. Copying an input to an output.

```
'-----  
'UNO_Input1.BAS  
'-----  
$regfile = "m328pdef.d"    'ATmega328p  
$crystal = 16000000      '16 MHz  
  
'-----  
  
Config Portb = Output  
Config Portc = Input  
  
Do  
  Portb.5 = Pinc.5        'Copy Input to Output  
  Waitms 19              '19 ms  
Loop
```


8	4	2	1	Decimal	Hexadecimal
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

For comparison (and for those of you who have forgotten your first-grade math): our usual number system with base 10 comes from the more or less random biological circumstance that we have a total of ten fingers and thumbs. The weights of the individual digits are 1, 10, 100, 1000 and so on.

The eight bits of a byte can be divided into two sets of four bits. Each set of four bits can then be expressed

directly as a hexadecimal number. For example, the binary number &B00100000 (0010 0000) is the same as &H20 in hexadecimal notation.

To convert a hexadecimal number into a digital number, you only need to know that the hexadecimal system is based on 16 as the name indicates (of course, in that regard it helps if you know a bit of classical Greek and Latin). The weights of the digits from right to left are therefore 1, 16, 256, 4096 and so on. For example, the number &H20 corresponds to $2 \times 16 = 32$.

If you often need to convert binary numbers into decimal numbers, you should memorize the weights of the eight binary digits of a byte. Armed with that knowledge, all you have to do is to add up the individual bit values.

Position	7	6	5	4	3	2	1	0
Value	128	64	32	16	8	4	2	1

Now for a question: what is the largest number that can be represented by a byte? If you said '255', you're right. Once you know this, the world of numbers is an open book for you and it is no longer a mystery why you can replace the instruction *Portb = Output* in Bascom with the simple instruction *Ddrb = 255*. This says that all bits in the data direction register for port B (DDRB) are set to 1 and all pins from PB0 to PB7 are outputs.

enough, you will see that the LED blinks. At first this may seem very strange, but when you think about it the reason becomes clear: a stroboscope effect. Your finger couples a signal into the input – the well-known 50-Hz AC hum signal with a period of 20 ms, or 16.6 ms for 60 Hz. By contrast, the wait time in the program is 19 ms. The 5% difference between the two times yields an output signal with a frequency of 2.5 Hz (50 Hz x 0.05). If you live in an area where the AC line frequency is 60 Hz, the output frequency—and thus the blinking rate—is correspondingly higher.

What makes a high level high?

Like all digital circuits, microcontroller output pins have only two states, which are variously called on and off, high and low, 1 and 0, or whatever. Furthermore, microcontrollers can only detect these two signal states on their input pins. However, in practice any voltage from 0 V (ground) to the supply voltage (Vcc) can be applied to an input pin. You might think that anything above 2.5 V is a high level and anything below 2.5 V is a low level. However, the ATmega328 data sheet states somewhat circumspectly that the minimum

value where the pin is guaranteed to be read as high is 0.7 Vcc (corresponding to 3.5 V) and the maximum value where the pin is guaranteed to read as low is 0.3 Vcc (corresponding to 1.5 V). That sounds a bit like a disclaimer intended to prevent unjustified complaints about supposedly incorrect operation: if you connect a signal between 1.5 and 3.5 V to an input, whatever happens is your responsibility. But what actually happens in that case? The only way to find out is to simply do it and observe the results.

Go ahead, try it! Use an external power supply for this so that Vcc is exactly 5 V, since the voltage available from a USB port is usually only 4.5 V or so. Connect a potentiometer as shown in **Figure 4** and vary the input voltage over the range of 0 to 5 V. Here's what you should measure with the microcontroller: If you gradually raise the voltage from 0 V, the output state changes from low to high at 2.7 V and the yellow LED lights up. If you then slowly reduce the voltage, the output state changes back to low at 2.2 V. You can repeat this process as often as you like. The switching points are 0.5 V apart, and no change occurs in the range between 2.2 and 2.7 V. This means

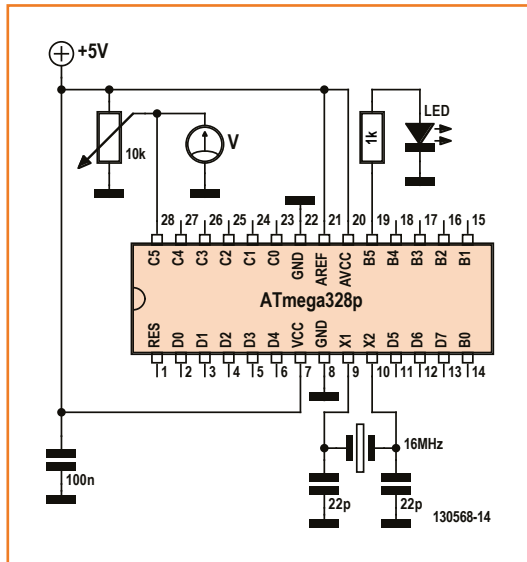


Figure 4.
Measuring the input voltage.

that there is a hysteresis of 0.5 V. The data sheet doesn't actually say this, perhaps because nobody at Atmel wants to claim that you can rely on this property. However, it's a useful property because it means that the inputs act like Schmitt triggers. As you may know, the purpose of Schmitt triggers is to convert analog signals with variable voltage levels into digital signals.

Switching back and forth

You can also automate the above process and dispense with the potentiometer. You already have the necessary hardware in the form of the microcontroller, and the idea is to have it measure its own switching points. In the circuit in **Figure 5**, the capacitor connected to the input pin is charged by the output pin through a series resistor. The output should go to 0 when a 1 is read at the input, and it should go to 1 when a 0 is read at the input. This circuit is effectively a sort of on/off controller with hysteresis. You need to change the program slightly for this purpose, as shown in **Listing 2**. The *Not* function inverts a digital state, in the same way as an inverter gate in hardware logic. The wait time is not necessary because the program should respond immediately when a new state is detected. In this case, "immediately" means within a microsecond or so.

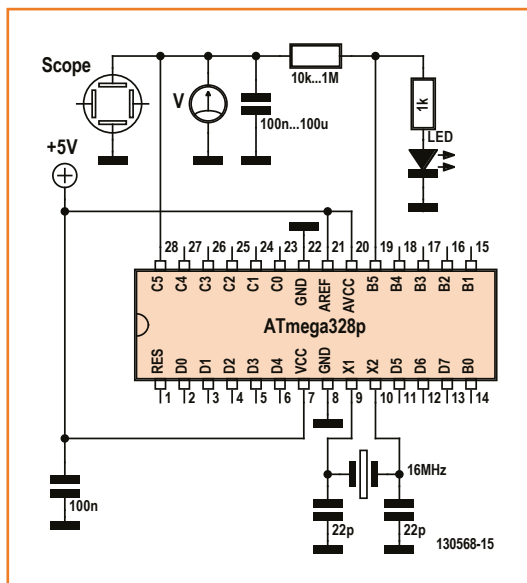
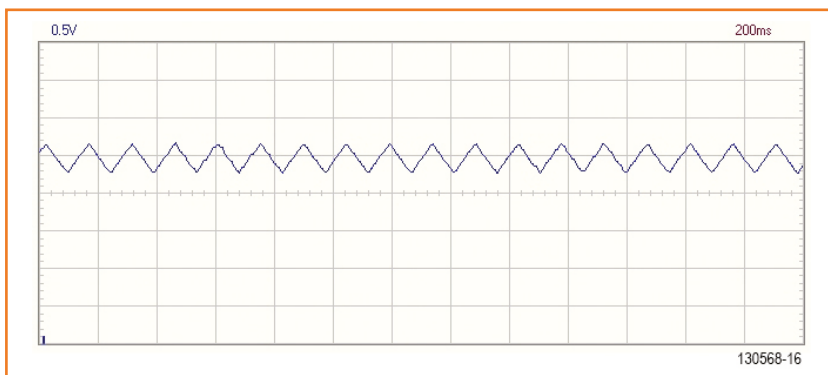


Figure 5.
A control loop.

The potentiometer of the previous circuit is replaced here by an RC network. The component values of this network (and the corresponding time constant) can be chosen freely over a wide range. For instance, you could use 100 nF and 10 kΩ if you have an oscilloscope available. In that case you can view the triangular waveform and see the hysteresis directly (**Figure 6**). If you use a digital multimeter with high input impedance to make your measurements, it's better to use 100 μF and 1 MΩ. In that case the voltage changes so slowly that you can easily read the minimum and maximum levels from the meter.

Figure 6.
Voltage waveform at the input.



Actually you don't need a measuring device for this process, since you can use the analog inputs of the Arduino board to measure the voltage. Any voltage in the range of 0 to 5 V can be measured digitally with a resolution of approximately 5 mV. We'll focus on how to do this in the next part of this series.

Incidentally, if you want to examine the response time of the program in more detail, you can simply omit the RC network and connect pin PB5 directly to pin PC5. If you then measure the signal with an oscilloscope, you will see a rectangular waveform

with a frequency of approximately 500 kHz. This means that the circuit responds to changes in the input state roughly once per microsecond.

There's only one strange thing about this circuit: with a symmetrical 500-kHz square-wave signal, the yellow LED on the Arduino board should be lit at half its normal brightness, but it remains dark. This reason for this can only be determined by examining the circuit diagram of the Arduino Uno in more detail. The built-in LED on the Uno board is connected to PB5 through a buffer consisting of one half of an LM358 dual opamp. This is unusual on a digital circuit board, and it comes from the fact that the other half of the LM358 is used as a comparator for the supply voltage. If you connect an external power supply, the Uno board automatically disconnects the USB supply voltage—very user-friendly and reliable. The other opamp is redundant and actually shouldn't be used, but the designer took advantage of its high input impedance to use it as buffer that allows port pin PB5 (Arduino pin 13) to also be used as an input. However, a 500-kHz signal is simply outside the frequency range of an ordinary opamp, so the LED on the board remains dark. If you want to actually see something, you have to connect your own LED to the board as shown in Figure 1.

Branching

Microcontrollers can make decisions by following program branches. In practical terms, this means that parts of a program are either executed or not executed, depending on specific conditions. The program structure used for this in Bascom is the If block, which takes the form *If <condition> Then <instructions> End If*. The instructions inside the If block are executed if the condition is fulfilled, and otherwise these instructions are not executed and the program jumps directly to *End If*.

In the program in **Listing 3** you can also see another new instruction called *Toggle*, which means “switch to the opposite state” and causes the output to change state each time the instruction is executed. The blinker runs as long as the signal level on port pin PC5 is logic 1, which means 5 V or at least something higher than 2.7 V. When the input pin is connected to ground, which corresponds to logic 0, the LED remains in its current state – either constantly lit or constantly dark.

The circuit is the same as in Figure 1. Here as well, you can test the circuit by connecting the input to ground or +5 V, or by touching it with your finger.

An interesting aspect in this regard is that your finger acts like a high (logic 1) signal. This means that if you touch the input pin when nothing else is connected to it, the LED will blink at a steady rate. This results from the *Wait* instruction in the If block. When the program sees a 0 level on PC5, it immediately jumps to *End If* and then back to the start of the loop. This means that a 0 level causes the If condition to be tested repeatedly in rapid succession. If you apply a 50-Hz signal to the input by touching it with your finger, after at most 10 ms a 1 level is detected at the input. Then the

Listing 2. Inverting the input signal.

```
'-----
'UNO_Input2.BAS
'-----

$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000      '16 MHz

'-----

Config Portb = Output
Config Portc = Input

Do
  Portb.5 = Not Pinc.5    'Inverted Input to Output
Loop
```

Listing 3. An If block.

```
'-----
'UNO_Input3.BAS
'-----

$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000      '16 MHz

'-----

Config Portb = Output
Config Portc = Input

Do
  If Pinc.5 = 1 Then
    Toggle Portb.5
    Waitms 250
  End If
Loop
```

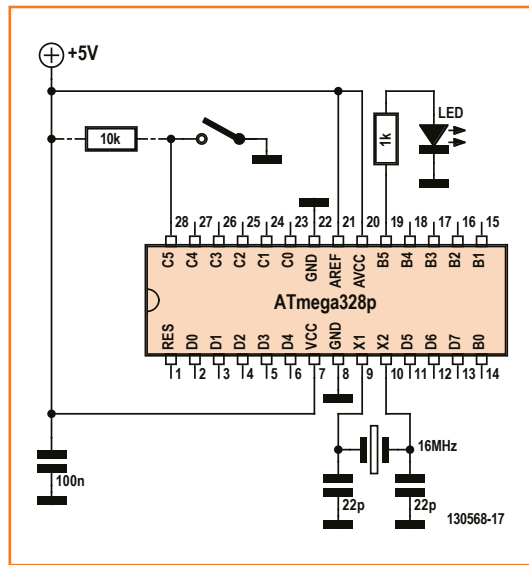



Figure 7.
A pull-up resistor.

LED changes state and the program waits 250 ms before testing the If condition again.

This aspect of the program also shows you what you shouldn't do. Open inputs are bad news and cause nasty problems, especially if you overlook them. In the case of this simple program, it's totally unpredictable what will actually happen when nothing is connected to the input. The input may be at the high level, in which case the will LED blink, or it may be at a low level, in which

case the LED will not blink – but you can't say in advance whether or not the LED will blink. If you hold your hand close to the input and move your feet, the static charge on your body can cause the port to change states as a result of capacitive coupling through the air. An outside observer who doesn't know all the details might start looking for a pesky intermittent contact, and perhaps start to doubt the reliability of microcontrollers in general. Many experienced designers have learned about this the hard way by spending hours looking for a suspected software bug or hardware error, when the actual problem was an open input.

Reading switch states with a pull-up resistor

From the above, we can conclude that open inputs cause problems. Suppose you want to read the state of a switch. This means that the input must have a defined state when the switch is open. The usual way to achieve this is to use a pull-up resistor, which is a resistor connected to Vcc to pull the input high when it is open. If the microcontroller sees a low level at the input, it knows that the switch connected to the input is closed.

The pull-up resistor in **Figure 7** is shown connected by dashed lines, which means that the resistor is optional. This is because the microcontroller has built-in pull-up resistors; all you have to do is connect them. Many microcontrollers have a separate register for this purpose. However, AVR microcontrollers manage to handle everything with a total of three registers: PORTC, PINC and DDRC. In this case you set all bits of DDRC to 0, which configures all pins of port C as inputs. You also set all bits of PORTC to 1, as though you wanted to set all the pins to high outputs. What actually happens in this case is that each port pin is connected to Vcc by an internal pull-up resistor (marked "Rpu" in Figure 3) with a resistance of approximately 30 kΩ. This causes the input pins to have a high signal level and a relatively low input impedance, instead of a high impedance. As a result, all bits in PINB will be read as 1 unless they the corresponding pin is connected to ground by an external switch.

In **Listing 4** you only need the instruction *Portc.5 = 1*, which connects the internal pull-up resistor to pin PC5. The other inputs of port C are left in the high-impedance state without a pull-up resistor. That doesn't matter here because they are not used in the program. With this change, the program response is unambiguous. An open

Listing 4. Polling a port with a pull-up resistor.

```

'-----
'UNO_Input4.BAS
'-----
$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000      '16 MHz
'-----

Config Portb = Output
Config Portc = Input
Portc.5 = 1             'Pullup

Do
  If Pinc.5 = 1 Then
    Toggle Portb.5
    Waitms 250
  Else
    Portb.5 = 0
  End If
Loop

```

input is always read as a high level (logic 1), and the LED blinks. Another change is necessary to ensure that there is no ambiguity when the switch is closed. This consists of adding an *Else* section to the *If* block. The instructions between *Then* and *Else* define what has to be done when the switch is open, and the instructions between *Else* and *End If* define what has to be done when the switch is closed. In this case, the LED should be off then. By the way, you might want to measure the actual value of the internal pull-up resistor. The data sheet says that it is somewhere between 20 kΩ

and 50 kΩ. To check this, connect an ammeter to the input pin instead of the switch, or in parallel with the switch. In our case, we measured 140 μA. This corresponds to a resistance of 35.7 kΩ ($5\text{ V} \div 0.14\text{ mA}$), which is exactly in the middle of the range specified by Atmel.

(130568-1)

Web Links

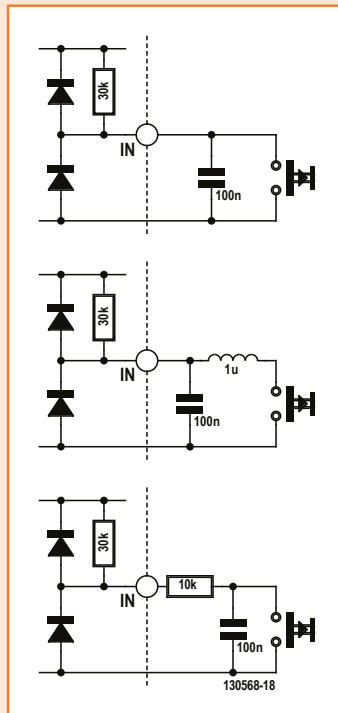
[1] www.elektor-magazine.com/120574

[2] www.elektor-magazine.com/130568

Latch-up

You can use the input protection diodes of the port pins intentionally in your circuits to limit input voltages higher than V_{cc} . To take an RS232 interface as an example: the signal voltage on the output line of a PC serial port is often -12 V and $+12\text{ V}$, or in some cases somewhat less. This is a complete mismatch to CMOS inputs with 5 V signal levels. The quick and dirty solution is to connect the signal line directly to the input through a protective series resistor (value $10\text{ k}\Omega$ or $100\text{ k}\Omega$) as illustrated in Figure 1. Here the protection diodes take care of the rest.

However, there's one thing you have to watch out for: the current through the protection diodes should never exceed a few milliamperes. This is because every CMOS IC has a parasitic thyristor that can be triggered unintentionally. It basically consists of an NPN transistor and a PNP transistor, which are a sort of undesirable side effect of the CMOS structure. This thyristor can be triggered by the current through the protection diodes if it rises above a certain value, which might be 100 mA or as high as 1 A —but you never know the precise value. The worst thing about this is that even very short current spikes can trigger the thyristor. After the device enters the latch-up state, what happens next depends on how much current the power supply can deliver. The latched CMOS IC draws as much current as it can and gets blistering hot. So if you get your fingers get burned, switch off the power right away and wait a few minutes. There's still a small chance that the IC will have survived, but in most cases it's game over. If you measure the resistance between the V_{cc} and ground pins, you'll only read a few ohms.

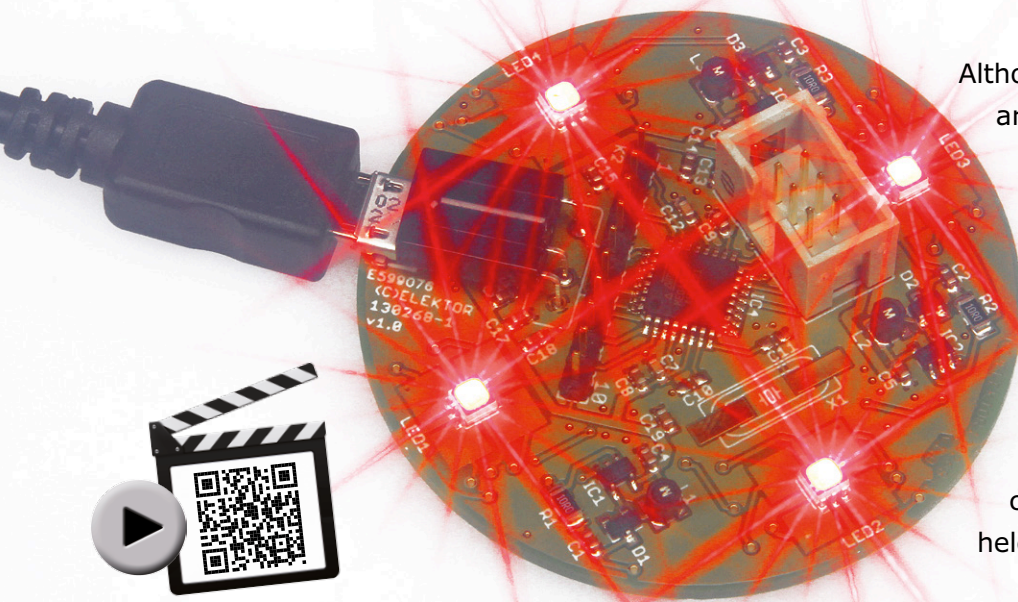


If your body has a static charge and you touch an IC input, the current pulse may be large enough to trigger latch-up. Many modern ICs can withstand static discharges up to 15 kV , based on a human body model with a contact resistance of $1\text{ k}\Omega$. In that case the ignition threshold is about 15 A , and you can certainly feel static discharges of that magnitude. However, no IC can survive contact with a 12 V power lead lying loose on the bench, especially if the power supply has a hefty output capacitor. If the power lead touches an IC pin, the microcontroller is guaranteed to be dead. Have a look at the circuit diagram in the above figure. Do you see anything to be worried about? To all appearances, the only thing the capacitor on the input does is to debounce the switch contacts. This is common practice to ensure that the microcontroller only sees a single level change when the switch contacts bounce back and forth a few times. What the circuit diagram doesn't show is the length of the

wires and their inductance. If you imagine an inductor in the circuit as shown in the middle diagram, you have something that looks a lot like Marconi's spark-gap transmitter. When the switch is closed, there is an excited resonant circuit that oscillates at 500 kHz , assuming a capacitance of 100 nF and an inductance of $1\text{ }\mu\text{H}$. There's also enough energy available, since the capacitor has previously been charged through the pull-up resistor. The first peak of the oscillation waveform hits the protection diode with full force, and under unfavorable conditions it can trigger that nasty thyristor. The circuit diagram at the bottom shows how you can prevent this. All you need is a current limiting resistor that soaks up the excess energy.

An RGB LED Lamp

With remote control!



Although the technology has been around for a while, RGB LED lamps are still state of the art for interior lighting. Traditional incandescent light sources found in night clubs, stage lighting, car instrument panels and interior room lighting have all benefited from a semi-conductor makeover. Here we add the convenience of a ready made hand-held remote controller.

By **Martin Christoph**
(Germany)

Interior designers will tell us that the color of our surroundings has a big influence on our mood. White spaces fill us with a sense of calm sobriety. The redder the light, the warmer we feel. Blue light gives an impression of coolness; you are more likely to start shivering in a blue room than you are in a white room. With space at a premium our living areas must now be multifunctional. With the appropriate background color, the room atmosphere can be adapted to enhance its suitability for your chosen activity. With the help of a little technology we can now achieve this goal in seconds without resorting to DIY overalls, dust sheets and a paint roller.

Instead we can set the background lighting color almost instantly to achieve the desired atmosphere. When the dinner table is used by your kids to do their homework you can select a daylight color temperature. This has a hint of blue which will suppress production of the sleep hormone melatonin so that they remain alert and able to concentrate (explain this to them carefully... they now have no excuses). As the time comes for supper, adjust the color of the eating area to give an orange cast to the light; this is not only cozy but is said to stimulate your appetite and boost metabolism. In the living room a

yellow light gives a sunny radiance and helps lift the mood. When it's time to break out your Barry White albums just lower the lights and dial the mood to red.

You can achieve these effects with the lighting unit described here. It consists of a 57 mm diameter PCB fitted with four programmable RGB LEDs. To set the light color you don't need to fumble around with switches or controls on the unit; you can use a TV remote controller handset to give you just the right lighting effect.

Special components

The complaint you hear most often about RGB type LEDs is that the blue emitter is less intense than the other two colors. For some time now Osram have had their MultiLED devices on the market. These come in an SMD package with six connection pads and achieves an output of 370 mcd from the blue emitter (even up to 560 mcd in the upper blue wavelength). This is a distinct improvement compared to other multi color LED devices. The increased brightness in this range is beneficial because our eyes are less sensitive to the blue end of the spectrum than they are to the yellow region for example. In

In addition the short blue wavelength gives a good impression of a fully saturated blue light. Using the MultiLED gives you a more even coverage of the complete RGB spectrum. The MultiLED type LRTB G6TG contains three independently controllable LEDs emitting wavelengths of 632 nm (red), 523 nm (green) and 465 nm (blue). This series also contains other types with different brightness relationships [1], here we use the type given in the parts list.

The circuit diagram, **Figure 1** shows four LED units with the same color LED in each unit connected in series. An LED boost driver drives each of the three colors. There are a number of ICs

available for this purpose; here we have chosen the NCP5007 from ON Semiconductors [2]. This particular IC is relatively low cost and can drive up to five series-connected LEDs. The NCP5007 needs just five external components mostly for power decoupling. A resistor connected to the FB pin defines current through the LED chain. Two methods can be used to control LED brightness: an analog voltage (or PWM signal with an R/C filter) applied to the FB input or pulse width modulating the IC Enable input which we have used here.

A standard IR receiver module type TSOP31236 (for a 36 kHz carrier frequency) is used to demodulate the IR signals. An Atmel ATmega328 (as

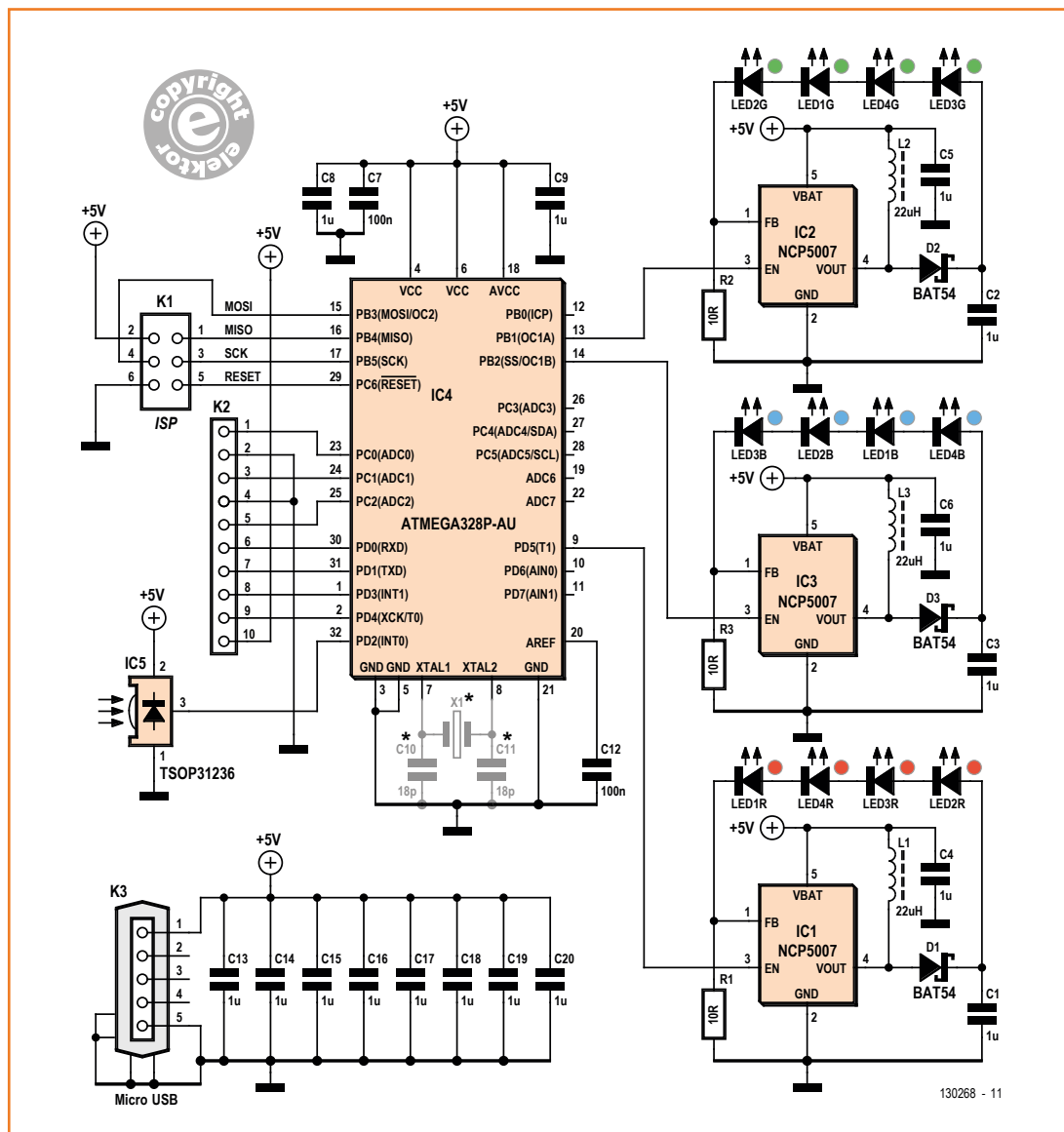


Figure 1. The controller receives IR commands and controls the LEDs using PWM signals.

used in the Arduino) decodes the IR command messages, converts them into RGB values and outputs them to the hardware PWM pins on the LED boost driver chips. An 8 MHz clock is more than adequate for this work so the microcontroller's own internal clock is used here.

The board's low power requirements allow the use of a micro USB socket for connection of an external USB power adapter. All components in the circuit have an operational voltage range between 2.7 and 5.5 V so there is no need for any additional on-board power regulator.

Altogether there are 28 components (11 different

types). The PCB layout includes some expansion options for future use but aren't necessary for operation of the unit. These measures include pads on the PCB to allow an external crystal to be fitted and also some GPIOs available at the SIL pin header K2.

RC5 to HSV to RGB to PWM

The special feature of this RGB LED controller is that it uses a standard TV remote controller to send color and brightness information to the lamp. The circuit receives IR signals in RC-5 format, converts the commands to HSV color

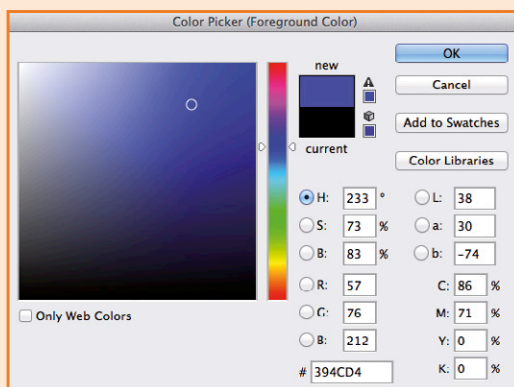
A color model primer

The work most engineers get involved with rarely requires them to have an appreciation of color manipulation and representation.

This is an area where the worlds of art and science touch. It's probably fair to say most are already aware of the red green and blue (RGB) color concept in connection with computer monitors and camera pixels and also that Cyan, Magenta and Yellow are used for color representation in printing. But

most people won't be familiar with the concept of a color wheel (or is that a color sphere?) which contains all the base colors and the mixes of base and complementary colors. And what's all that about white/black and achromatic?

Open up the color picker in Photoshop (or the corresponding tool in a similar graphics program) and have a play around with the options. On the left is shown a section from the color field which is represented on the vertical color slider in the centre. When you click here the selected color is shown in a box and beneath it are the corresponding RGB values for the color. Each of the three colors can have a value from 0 to $2^8-1 = 255$, when RGB = 255,255,255 this indicates the color white and RGB = 0, 0, 0 represents black. Underneath these values are the RGB values in hexadecimal and on the right are the CMYK values (in percentage) of the color used by printers (K stands for Key and refers to the black value). While the color information can be given by these three



values it doesn't give us the whole picture because although we have color information there is nothing about brightness or amount of black or white. Above the RGB and CMYK values in Photoshop there are also HSB and Lab values for the selected color, these two systems do not suffer from the same shortfall and are preferred by the professionals. The RGB lamp controller uses the HSV color model (Hue, Saturation and Value) also known as the HSB

(Hue Saturation and Brightness) to control the LED light output: (photo source: Wikipedia [3]):

Hue

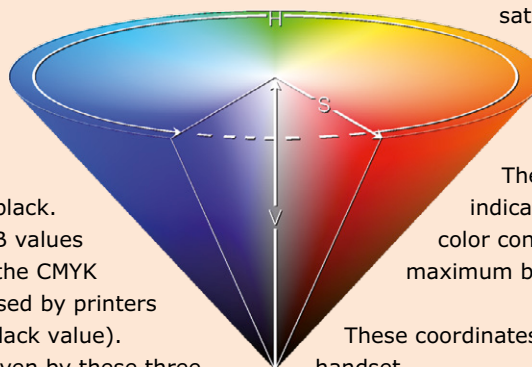
The hue is defined with an angular value where Red is (0°), yellow (60°), Green (120°), Cyan (180°), Blue (240°), Magenta (300°)

Saturation

The color saturation is given as a percentage. The color saturation is greatest at the edge of the circle (100 %) and at the centre (0 %) only grey (white to black)

Brightness

The brightness is given as a percentage indicating how much white/black the color contains. 100 % indicates none, 0 % is maximum black.



These coordinates are sent by the remote controller handset.

Undergo the Red / Green / Blue Miracle

model information to control the LED light output. HSV is also sometimes referred to as HSB which substitutes the word Brightness for Value. If the color wheel concept is completely alien to you, take a look at the color wheel primer in this article. This should give you a quick introduction to phraseology graphic designers use for color representation.

The firmware decodes infra red RC5 signals from the remote controller (for TVs with a device code ID 0) and converts these values internally to levels of hue, saturation and brightness. Next comes a HSV/RGB conversion in accordance with the algorithm described in [3], finally the new values of PWM signals are calculated and sent to the boost-converter IC's. The firmware uses in-phase 8-bit PWM and the controller's PWM hardware functions. The saturation and brightness levels are defined with almost 8-bit resolution. The LED controller can be configured to learn the RC5 codes generated by your TV remote controller. First short together pins 1 and 2 of on the expansion header K2 then power up the LED controller. In this 'learn' mode the LED color will be green (instead of blue which indicates normal operation). Now press the twenty pushbuttons on the RC-5 remote controller in the sequence given in **Table 1** for the controller to learn all the codes. Each button pressed is acknowledged by a green flash from the LED controller. All other push buttons on the remote controller have no function in this application.

Use full saturation of a base color (not white) when setting the hue. You will soon get a feel of how it functions when you have played with the controller for a bit and tried out all the lighting and color options.

A standby function is implemented in the firmware; pressing the eleventh button (POWER) turns out the light (brightness = 0) and current consumption of the LED controller drops to a few hundred microamps. This low current standby mode is necessary to keep the IR receiver active so that signals from the remote controller can wake up the LED controller. When the controller goes to standby the most recent color settings are stored in EEPROM and reloaded on wakeup. The same procedure is followed when a power outage occurs. Unknown IR signals can bring the

controller out of standby but if no valid command is received it quickly returns to standby mode. At this point we should point out a problem we discovered: To test the LED lamp controller we bought a 13-1 universal replacement IR remote controller (type IM-1313) which we configured to send type RC-5 device ID 0 (TV) control signals. The all-in-one controller however caused a problem: when the button on the right was pressed to increase the hue angle it sent more than one RC-5 command which set the saturation level to zero at the same time. When this happened the hue button had no further effect on the color. If you get this problem try to find an original Philips TV remote controller from a boot sale or on-line auction site. That is guaranteed to send the correct codes.

Using a needle tip

Just looking at the PCB in **Figure 2** you can see that your large soldering iron bit isn't going to

Table 1. The remote controller operating keys

Key	Function
1	Brightness 1
2	Brightness 2
3	Brightness 4
4	Brightness 7
5	Brightness 11
6	Brightness 16
7	Brightness 22
8	Brightness 32
9	Brightness 252 (maximum)
0	Brightness 0
Power	Sleep-Mode, Brightness 0
red	Switch hue to red (0°)
green	Switch hue to green (120°)
yellow	Switch hue to yellow (60°)
blue	Switch hue to blue (240°)
up	Increase saturation
down	Reduce saturation
left	Reduce hue angle
right	Increase hue angle
OK	Switch saturation from max to min

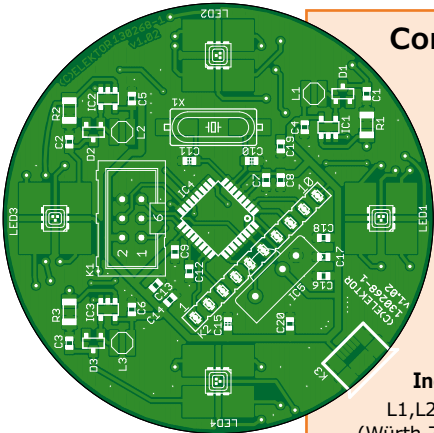


Figure 2.
The circular PCB populated with SMD components.

Component List

Resistor

R1,R2,R3 = 10Ω, 0.25W, 1%, SMD 1206

Capacitors

C1-C6,C8,C9,C13-C20 = 1μF 16V, 10%, X7R, SMD 0603

C7,C12 = 100nF, 25V, 10%, X7R, SMD 0603

C10,C11 = 18pF, 50V, 5%, COG/NP0, SMD 0603 (not used)

Inductors

L1,L2,L3 = 22μF, 0.5A, 20%, R_{DC} 0.575 Ω (Würth 744025220)

Semiconductors

D1,D2,D3 = BAT54

LED1-LED4 = LRTBG6TG-TU7 1+V7AW-36+ST7-68 (Osram) Mouser # 720-LRTBG6TGTU71V7AW

IC1,IC2,IC3 = NCP5007

IC4 = ATmega328P-AU, programmed, Elektor Store # 130268-41

IC5 = TSOP31236

Miscellaneous

K1 = 6-pin (2x3) pinheader, 0.1" pitch

K2 = 10-pin pinheader, 0.1" pitch

K3 = Micro-USB Type B, right angled, SMD, Molex 47346-0001

PCB # 130268-1

Alternative for LEDs:

LRTBG6TG-TV-1+VAW-36+ST7-69-20-R18-IB (Osram) RS Components # 697-3682



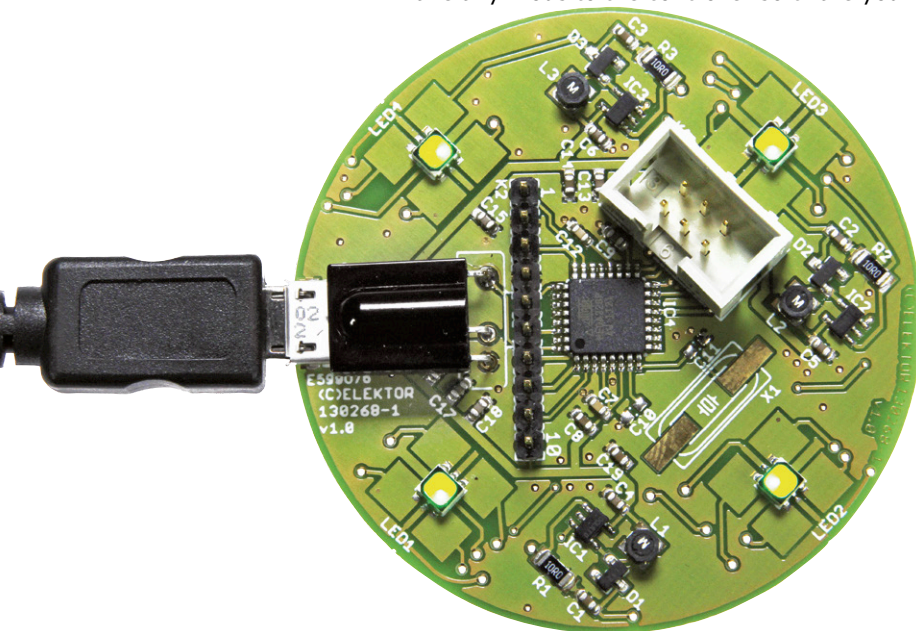
be of much use here, to fit the SMD components you will need a fine needle bit. The 57 mm diameter PCB populated with all SMD components (except the IR receiver and headers) and the pre-programmed controller can be ordered from the Elektor shop [4]. If you are not planning to make any mods to the controller software your-

self (The source and hex files for the project together with programming tips can be found at [4]) or need to alter the code (this would be necessary if you were planning to use a different remote controller which has a different device ID) then you can leave out the ISP connector K1. The same goes for the expansion connector K2, although you will need to use a two pin header between positions 1 and 2 to enable the learning procedure (alternatively just use a temporary wire bridge here).

To get a more diffuse lighting effect, try fitting a frosted acrylic half-globe over the PCB. It should be possible to source such an item from a hobby supply outlet or on-line.

The circuit takes a maximum of 170 mA with all the LEDs at maximum intensity. The majority of USB ports will be able to source this level of current without any problem. The USB connector is only used for supply of power; the data signal lines are not connected. Pin 10 (+) and Pin 4 (-) of connector K2 provides an alternative point to power the circuit.

(130268)



Web Links

[1] www.mouser.com/ds/2/311/RTB_G6TG_Pb_free-258912.pdf

[2] www.onsemi.com/pub_link/Collateral/NCP5007-D.PDF

[3] http://en.wikipedia.org/wiki/HSL_and_HSV

[4] www.elektor-magazine.com/130268

Display Week is the premier, must-see showcase for global information display companies and researchers looking to unveil cutting-edge developments in display technology. Not only did Display Week give the world its first glimpse at technologies such as LCDs, plasma and OLEDs that have shaped today's display industry - but it's also where emerging industry trends such as 4K, touch and interactivity, flexible and e-paper displays, solid-state lighting and plastic electronics will catch your eye. No other display event in the world offers Display Week's unique combination of qualified attendees, technologies and exhibitors, networking and educational opportunities, vital trade information, and state-of-the-art symposium presentations. You won't want to miss this year's event, so set your sights on **June 1-6, 2014 at the San Diego Convention Center.**

SID
**DISPLAY
WEEK '14**
SAN DIEGO



FLEXIBLE
DISPLAYS
LIGHTING
4K x 2K
TV



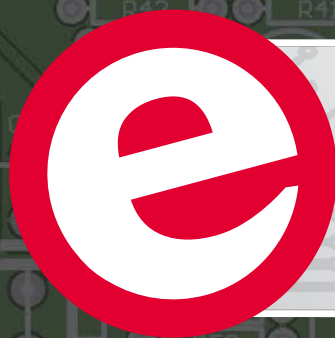
OLED
TV

OXIDE
TFT

TOUCH
& INTERACTIVITY



Register today! Early bird rates end May 16, 2014 at midnight PST!



elektor
PCB SERVICE

powered by Eurocircuits

25% Discount on new Elektor PCBs

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

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



Current Probe with Transimpedance Amplifier

By **Martin Ossmann**
(Germany)

Wideband alternating current measurements

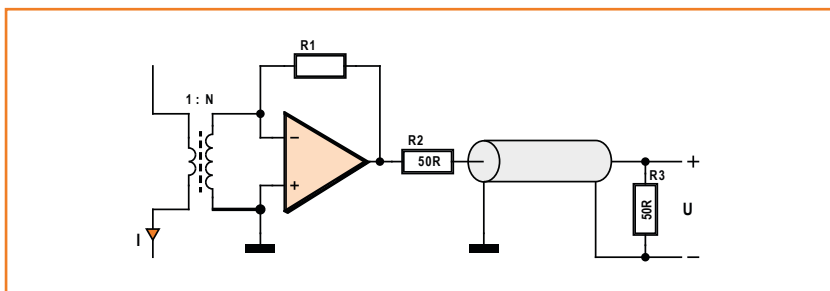


Figure 1.
Current measurement using a transimpedance amplifier.

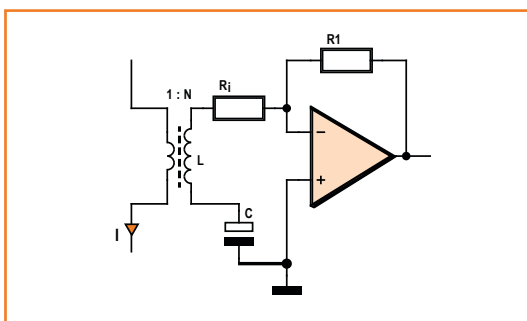


Figure 2.
AC coupling eliminates the effect of the offset voltage.

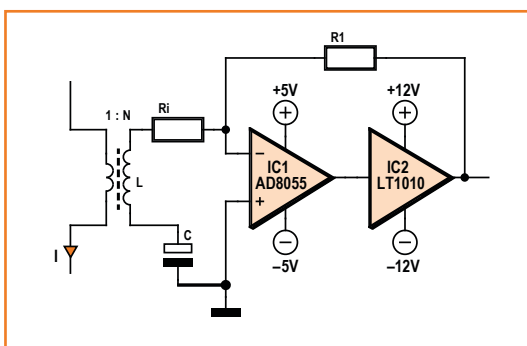


Figure 3.
The LT1010 can deliver a higher current.

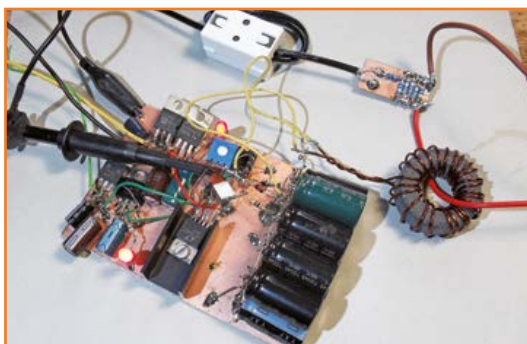


Figure 4.
The circuit of Figure 3 in use.

This article describes how easy it can be to use a current transformer to make precise measurements of alternating currents. We use a wide variety of readily-available ferrite cores in conjunction with a simple circuit incorporating a transimpedance amplifier.

In a previous installment [1] we saw how the lowest operating frequency of a classical current transformer is determined by the shunt impedance on the secondary side: if this is reduced, the lowest operating frequency also goes down, but the sensitivity of the device is also correspondingly lower. The question therefore arises whether it is possible to extend the lower frequency limit with the help of a bit of electronics. Ideally we would like to be able to work with a shunt having $R = 0$, and this can be achieved using a transimpedance amplifier. The outline of the circuit of such a current probe is shown in **Figure 1**.

The operational amplifier controls its output such that the current through $R1$ exactly compensates for the current through the secondary coil, and hence the differential input voltage to the amplifier is zero. Since the voltage across the secondary coil is zero, it is in effect short-circuited. The output of the circuit is matched to a $50\text{-}\Omega$ impedance, and the effective transfer impedance is thus given by:

$$R_{tr} = \frac{R1}{2N}$$

Unfortunately we cannot realize the circuit directly in the form shown. The reason is that, from a DC point of view, the input to the operational amplifier is effectively short-circuited by the coil and so no negative feedback is provided by $R1$. The quiescent output of the operational amplifier will

therefore be equal to its offset voltage multiplied by its open-loop DC gain. If, for example, the offset voltage is 1 mV and the open-loop gain is 100,000, then the amplifier will go into saturation. To solve this problem we add a capacitor C in series with the secondary coil, as shown in **Figure 2**.

The lowest operating frequency is now determined by the resonance of the circuit formed by L and C . If we let $N = 25$ and select a ferrite core with $A_L = 3 \mu\text{H}/N^2$, then we have $L \approx 2 \text{ mH}$. To achieve a resonant frequency $f_{\text{res}} = 20 \text{ Hz}$ we need $C = 35,000 \mu\text{F}$, which is rather large. **Figure 3** shows an assembled prototype, where $C = 33,000 \mu\text{F}$ is formed of ten electrolytics in parallel.

A type AD8055 operational amplifier is used, with a bandwidth of 350 MHz. Its output is capable of delivering up to around 50 mA, which allows currents of up to about $N \times 50 \text{ mA} = 1.25 \text{ A}$ to be measured. Adding an LT1010 buffer after the AD8055 (**Figure 4**) increases the maximum measurable current to 5 A.

The main disadvantage of this circuit is the requirement for a large capacitor, needed to allow negative feedback at DC for the operational amplifier. An alternative would be to provide an 'active' negative feedback path using another operational amplifier. The complete circuit that results is shown in **Figure 5**.

Regulator IC1 provides the 10-V supply for the operational amplifiers. A mid-level 'ground' is generated by IC2a. The transimpedance amplifier is built around IC4, with IC2b in effect providing a reference voltage for the secondary coil of the current transformer. IC2b adjusts this voltage to ensure that the DC level at the output of IC4 is zero. The gain can be set using R8.

Practical use

For a first example we measured the primary and secondary currents in the transformer of a small switching supply (**Figure 6**). The two upper traces in **Figure 7** show the primary current, the first measured using a Tektronix TC202 current probe and the second using our transimpedance probe. There is good agreement between the traces. The lower trace shows the secondary current, measured using a Vitroperm ring core (see [1]). Since the unit under test is a forward converter, primary and secondary currents are aligned with one another.

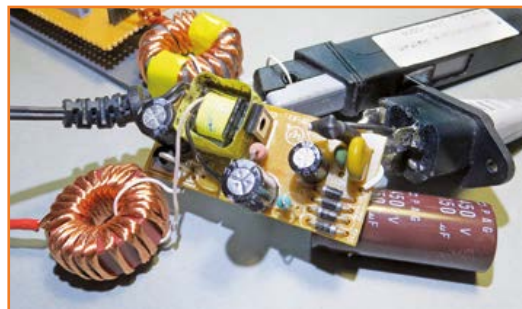
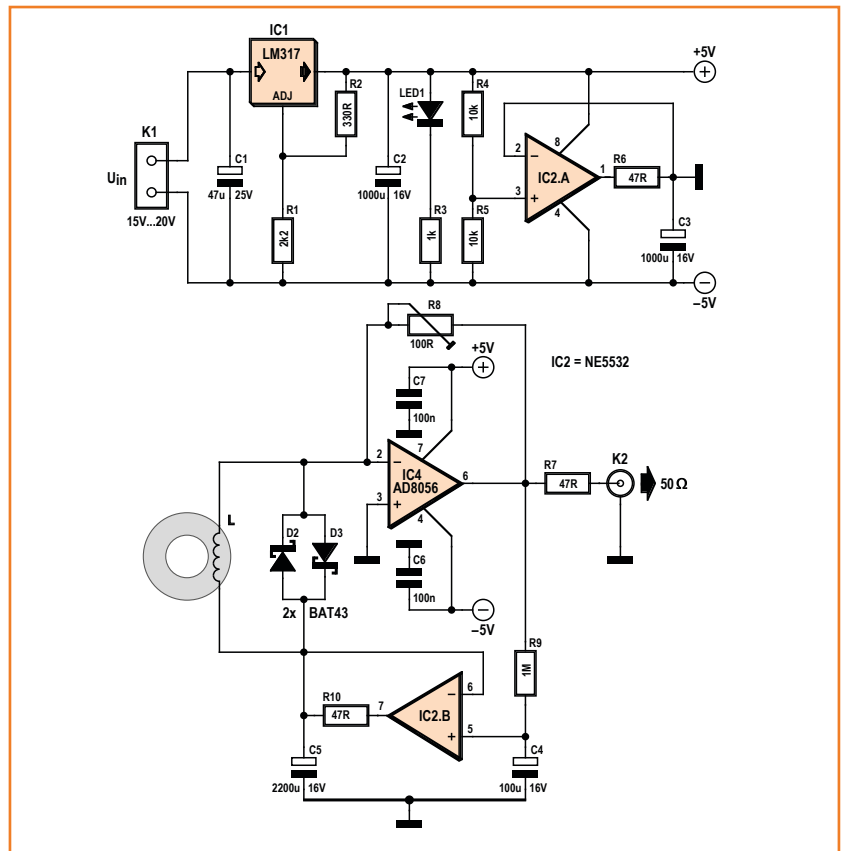


Figure 5. Circuit with offset voltage control.

Figure 6. Making measurements on a switch-mode power supply.

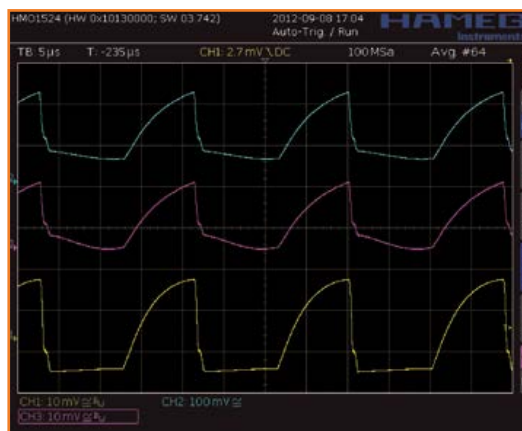


Figure 7. The upper two traces show the primary current (measured with a Tektronix TC202 current probe and with the transimpedance current probe); the bottom trace shows the secondary current.

Figure 8.
Measuring the current in a coil in near-saturation.

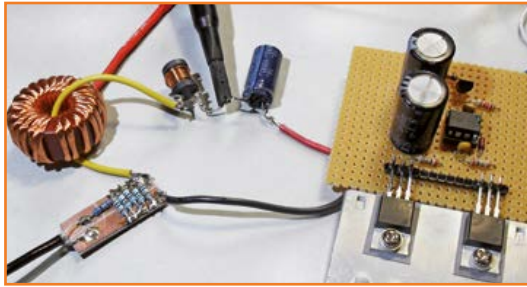
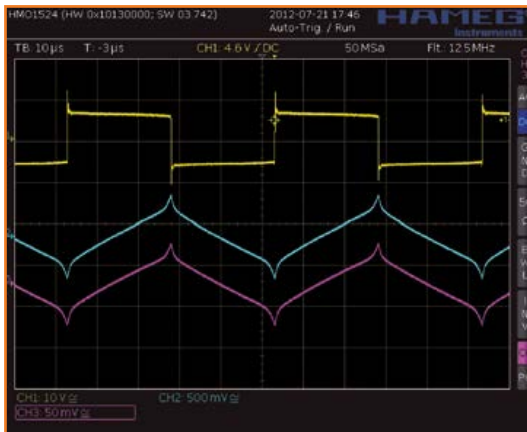


Figure 9.
Voltage (above) and current (below) measurements on a drum-core inductor.



Measuring saturation

Having a way to measure currents accurately lets us look at the behavior of coils at higher currents. **Figure 8** shows measurements being carried out on a drum core inductor. The current was measured using a shunt and then using the current transformer.

Figure 9 shows the coil voltage (above) and the coil current (below). The current waveform is triangular and the peaks show that the core is already in saturation. To generate the high currents needed for this experiment a half-bridge driver circuit (**Figure 10**) was used.

Other core shapes

It is of course possible to use cores with other shapes to make current transformers. Sometimes there is a requirement to make measurements on circuits where the lowest frequency present is only in the tens of kilohertz, and in these cases the lower frequency limit on the probe is less critical. For example, a ferrite bead with ten turns of wire can give $R = 2 \Omega$ (see **Table 1**), a transfer impedance of 0.1 V/A, and a lower frequency limit of around 5 kHz.

Figure 10.
Half-bridge driver circuit for producing higher currents.

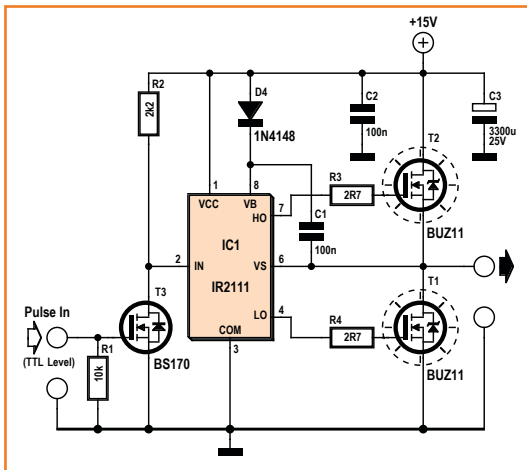


Figure 11.
Measuring current using a ferrite bead.



Figure 11 shows a ferrite bead being used to measure the current on the primary side of a switching supply based on a flyback converter. The results were compared with the Tektronix TC202 probe, and as **Figure 12** shows we get good agreement between the two measurements.

Binocular-core ferrites

Leafing through the manufacturer's datasheets we discover the BN73-202 binocular core with an A_L value of $14 \mu\text{H}/N^2$. This is surprisingly high, and makes possible the construction of a current transformer with a lower frequency limit of around 10 Hz (see **Table 1**). **Figure 13** shows the binocular core being used as a current transformer. The core was used to measure the input current of a switch-mode mains power supply. **Figure 14** shows a comparison against the results from the Tektronix probe, and again we get good agreement between the curves.

Split-core ferrites

The core shapes described above have the disadvantage that the wire carrying the current to be measured has to be threaded through a hole. This can be avoided using a split-core ferrite (**Figure 15**) of the type used for suppressing electro-

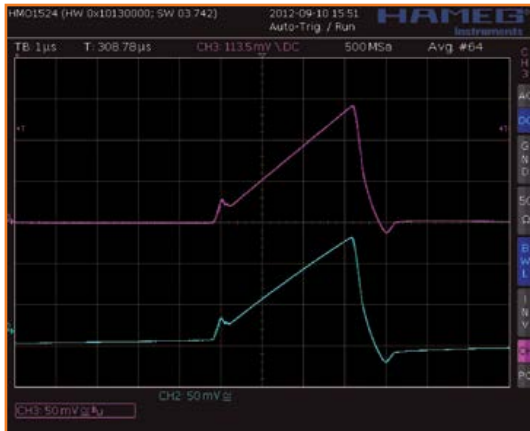


Figure 12. Measuring the current in a transistor in a switching power supply.

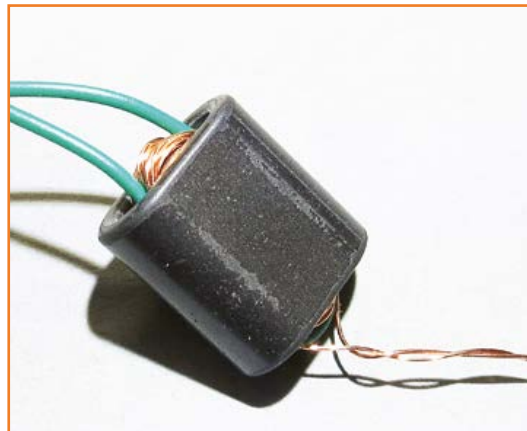


Figure 13. A binocular core can be used as a current transformer.

magnetic interference. The low A_L value of these cores makes them really suitable only for higher frequencies, but they can nevertheless be used in conjunction with the transimpedance amplifier.

Conclusion

The circuits, experiments and measurements we have described show how simple equipment can be used to replace an expensive professional current probe and still obtain good results. We hope that this article will help you to identify where the most significant potential sources of error can lie, and to know how to avoid them.

(130411)

Web Link

- [1] Current Transformer Calculations, Elektor April 2014, www.elektor-magazine.com/130410

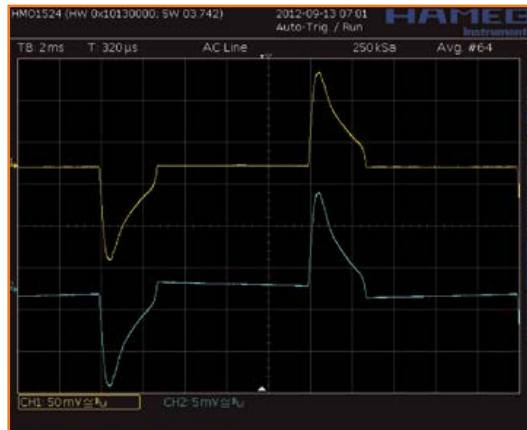


Figure 14. Measuring a current at 50 Hz using the binocular core of Figure 13.

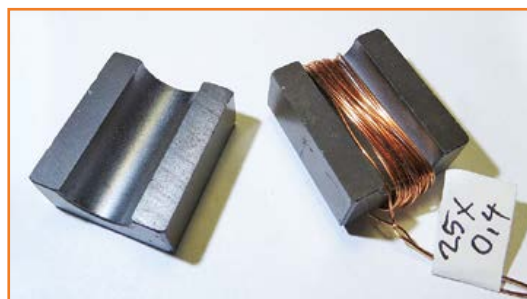


Figure 15. A split-core ferrite can be used in a current transformer.

Core	A_L value $\mu\text{H}/\text{N}^2$	Windings N	Resistance R [Ω]	Transfer impedance R_{tr} [Ω]	Frequency limit f_g [Hz]
Pollin	3.0	25	0.50	0.01	42.4
VITROPERM	80.0	25	0.50	0.01	1.6
Ferrite bead	0.56	10	2.00	0.10	5684
Binocular core	14.0	25	0.50	0.01	9.1
Split core	2.2	25	0.50	0.01	57.9



Grid Frequency Logger

Record frequency variations on the electricity grid

By **Wolfgang Borst**
(Germany)

Variations in frequency on the electricity grid often arise as a result of differences between supply and demand. Although small, these changes can be measured and, using a neat combination of two existing Elektor projects, can be monitored and recorded automatically.

Most people are familiar with the fact that the grid frequency in Europe is 50 Hz, while in other countries such as the USA, 60 Hz is used, and sophisticated control mechanisms are used to maintain

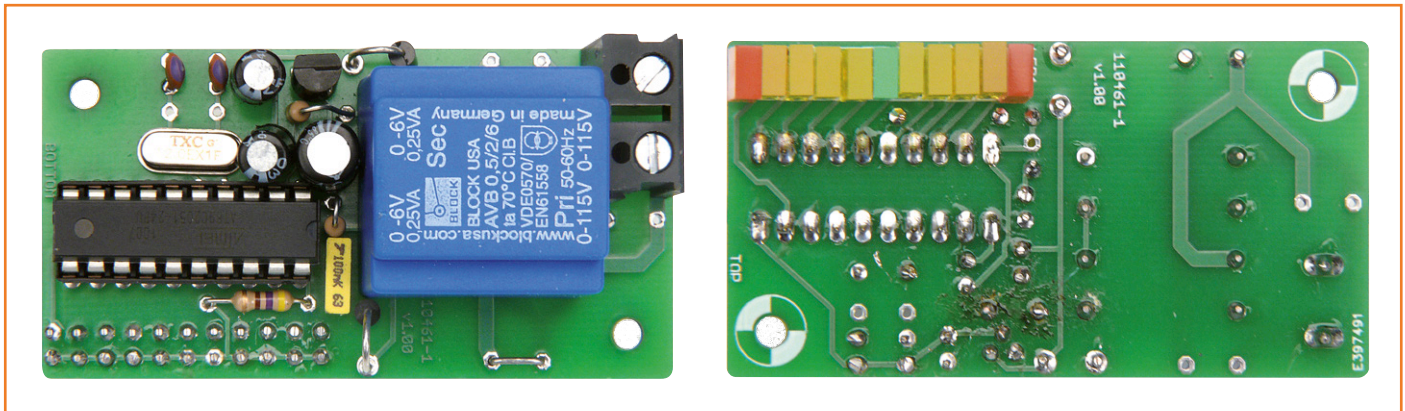
this frequency within very tight bounds. In the author's country, even in exceptional circumstances the frequency error is normally only at most 0.2 Hz, although in one incident in northern Germany, on the evening of 4 November 2006, an extreme excess of demand over supply led to the frequency dropping almost to 49 Hz for several minutes. The grid frequency can thus be used as a way to measure the balance between energy supply and demand and as a real-time proxy for the health of the grid as a whole.

Features

- Monitoring of grid frequency
- Logging over long time periods
- Logs stored on USB memory stick
- Frequency resolution: 2.5 mHz
- Absolute frequency accuracy: 25 mHz
- Display of instantaneous grid frequency using LEDs
- Graphical display of frequency against time
- Suitable for use on 230 V or 115 V grids
- Suitable for nominal frequencies of 50 Hz or 60 Hz
- Full source code available

Measure + record = log

The excellent 'Grid Frequency Monitor' project published in Elektor in January 2012 [1] provides a convenient means to indicate the instantaneous grid frequency. The circuit measures the frequency and displays the result using 11 LEDs to give a visual indication of deviations of up to ± 0.2 Hz, including two red LEDs that light when extremes of deviation occur. A printed circuit



board was designed for the project: as **Figure 1** shows, the unit, including its on-board transformer that can be configured to run on 115 V or 230 V for worldwide use, forms a very compact module. So this device solves the problem of how to make the measurements that we need.

What struck the author after reading that Elektor article was that it would be desirable to have a means of viewing the changes in grid frequency over time, rather than just seeing its instantaneous value. This involves periodically extracting the frequency readings from the device and storing them somewhere. Since the microcontroller used in the frequency monitor project had a pin to spare and some unused space left in its program memory, the obvious approach was to equip the monitor with a serial interface over which it could be made to transmit readings. Then the only problem remaining is to store these readings.

To receive and process the serial data the simplest solution would be to use a PC, which almost everyone would have available anyway. A terminal program, configured to match the communication settings of the modified frequency monitor, could be used to show the results directly. A more elegant approach would be to use the author's specially-written program, which can display the frequency readings graphically (see **Figure 2**). And even more elegant would be to avoid the use of a PC altogether and create a stand-alone solution. Again in this case we have a ready-made option, in the form of the 'USB Data Logger' project described immediately before the grid frequency monitor in the December 2011 issue of Elektor [2]. This design uses a microcontroller that simply records all incoming data on a USB

memory stick. **Figure 3** shows the tiny printed circuit board of the data logger.

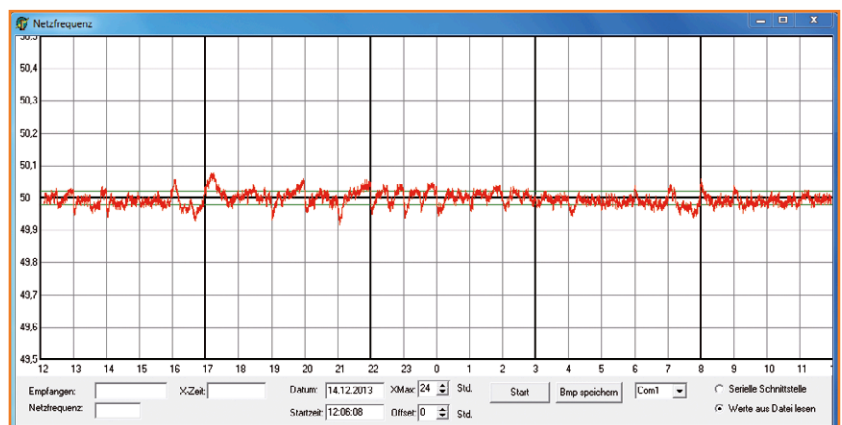
In addition to combining these two projects and modifying the firmware of the frequency monitor, the author also wrote a PC-based program to read back data recorded by the data logger on a USB memory stick and display them graphically. Bringing these elements together provides a complete solution for long-term monitoring of grid frequency data.

Hardware changes

As alluded to above, pin P3.1 of the microcontroller is not used in the grid frequency monitor project: conveniently for us, this can be configured as the output of the hardware UART peripheral. To connect the monitor to the data logger (or to a PC) all we need to do is bring this pin out: literally, in this case, as we simply solder short wires to pin 3 of IC1 and to a ground point on the board to connect to the additional hardware interface. **Figure 4** shows the original circuit of the frequency monitor, plus (highlighted)

Figure 1. Prototype of the grid frequency monitor published in Elektor in January 2012. The compact module indicates the grid frequency using a row of LEDs.

Figure 2. The program 'Netzfrequenz_P' can be used to visualize the logged frequency data.



the extra components that comprise the two new interfaces. To connect to the USB data logger we simply use a ten-pin two-row header on a 0.1 inch pitch: it is best to use a box header, mounted on a small extra piece of perforated board.

It is important to note that the power supply for the frequency monitor is not adequate to provide the additional power required by the USB data logger and its USB memory stick. This means that pin 1 (+5 V) and pin 2 (ground) on K3 should not be connected to the corresponding points on the frequency monitor board. An additional 5 V power supply is required, connected to these pins using a suitable socket. K3 can then be connected to the logger using a length of ten-way ribbon cable fitted with insulation displacement connectors,

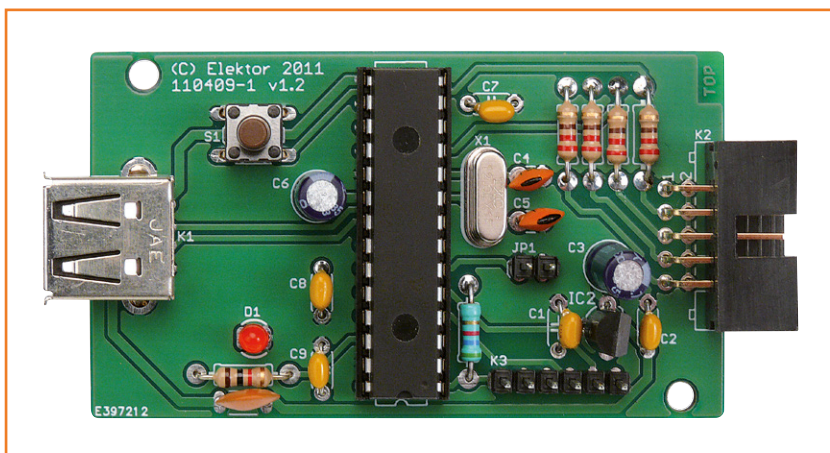


Figure 3. Prototype of the USB data logger published in Elektor in December 2011. The microcontroller is responsible for piping data received over its serial port onto a USB memory stick.

carrying both the data to be logged and power. A power adapter capable of delivering 0.5 A and 5 V should be adequate for the demands of any USB memory stick.

If you wish to keep the option of connecting the frequency monitor directly to the serial port on a PC, a suitable nine-way sub-D connector (K2) should also be mounted on the perforated board. It is also necessary to invert the polarity of the signal for the PC to be able to decode it correctly. A simple MOSFET (T1) and pull-up resistor (R4) make a suitable inverter and driver for the serial port on most PCs when run at a low data rate: in our case, the data rate is 4800 baud. If, due for example to the length of the interface cable, the circuit does not operate reliably, this interface circuit can easily be replaced by a 'genuine' RS-232 driver IC such as a MAX232.

Software changes

The hardware changes required are relatively straightforward, but we still need to consider the extensions needed to the firmware in the microcontroller in the grid frequency monitor to allow it to output data over its serial port. The author has stayed faithful to the original software by Dieter Laues [1] as far as possible: in particular, the LEDs are still correctly driven to allow direct monitoring of the instantaneous grid frequency. All the source code files and a hex file suitable for programming directly into the microcontroller are, of course, available for free download via the Elektor web page corresponding to this article [3]. If you wish to make further modifications to the software, you will need the Wickenhäuser C compiler [4]: the free demonstration version is adequate for this project.

The firmware has the following functions: it first measures the time taken for fifty cycles of the line input (sixty cycles in the case where the grid runs at 60 Hz) and converts the result to an output value. This means that a new value is obtained every fifty (or sixty) cycles, so about once every second. Hence if we know when a series of measurements was begun, then we can use this fact to determine, without the need for timestamps, the time corresponding to each measurement in the series. Since the long-term accuracy of the grid frequency is very high, we get good agreement between time estimates using this method and the actual time, even after many days of operation.

Every second a reading is also transmitted over the serial interface, either directly to a PC or to the data logger. The analysis software developed by the author turns a series of readings into a graph of grid frequency against time. The firmware configures the serial interface parameters to 4800 baud, no parity and one stop bit. To ensure that the USB data logger is configured in the same way it must, when it is powered up, find a text file on the USB memory stick containing the line "COM_BAUDRATE: 4800". This can be created using a simple text editor under Windows, or the example 'config.txt' file in the download [3] can be used instead. No other configuration or modification to the data logger is required.

PC program

As mentioned above, the PC-based program developed by the author is capable not only of

receiving data from the frequency monitor directly over the serial port but also of processing and displaying data stored in a file, for example by the USB data logger. The program can be switched between two operating modes using the control at the bottom right of its window (see Figure 2). 'XMax' can be used to adjust the displayed time range and 'Offset' the zero point (measured in hours) of the x-axis relative to the time of the first reading. 'Save bitmap' writes the currently-displayed frequency graph to disk as an image file.

The software does not need to be formally installed: it is simply necessary to ensure that the application 'Netzfrequenz_P.exe' and the file 'Netzfrequenz.ini' are stored in the same directory. The '.ini' file records all the settings of the

program, so that when it is opened parameters such as the window width and height are preserved from when it was last closed. If you decide to make changes to the program and then recompile it, bear in mind first that the program is written using Delphi, and secondly that before the compiler can be run the package file 'CommRec.dpk' for the serial interface must be installed. This file is included in the archive 'ASYNC32.ZIP' which is inside the directory of that name in the download [3].

File read mode

The USB data logger always writes its output in a file called 'LOGGING.TXT' on the USB memory stick. It is a good idea to rename the file to something more informative about its con-

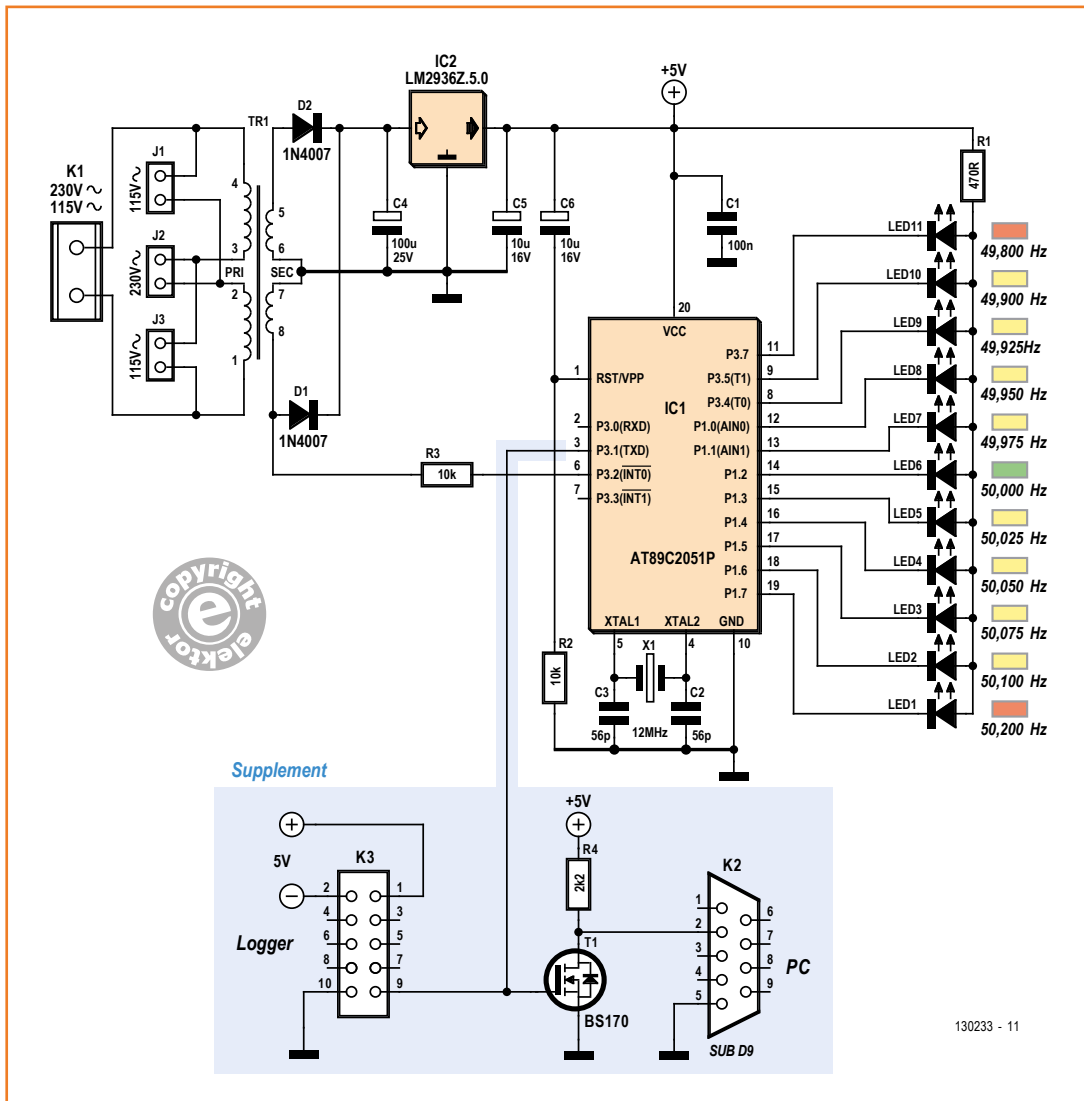


Figure 4. Expanding the original grid frequency monitor circuit to provide two serial interfaces requires just a couple of components.

The log file

It is possible for the first entry in the log file to be erroneous, and so it is best to delete it before proceeding: only one second's worth of data will be lost. The log file format is plain ASCII text and so can be edited using any ordinary text editor.

If you would prefer not to enter the date and time manually into the PC program whenever it is run, you can add an extra line at the start of the file (in other words, replacing the possibly erroneous reading) to indicate the date and time when the log started. The frequency values themselves are stored in the file as periods expressed as integers in units of one microsecond. A 50 Hz grid frequency is therefore represented by a stored value of 20000. Fresh from the USB memory stick a log file might start as follows:

```
9998
20001
20000
19998
...
```

and after replacing the first line with the date and time it might read:

```
14.12.2013 12:06:08
20001
20000
19998
...
```

A sample log file is included in the software download [3].

tents. One idea is to use a convention such as 'NetzLog_20131214_120608' where the date is encoded in 'YYYYMMDD' format and the start time as 'HHMMSS'. The program gives you the opportunity to select a file when 'Start' is clicked. The date and time can then be entered by hand in the appropriate fields in the program's main window. Alternatively the file itself can be edited before importing it into the program to include date and time information (see the 'The log file' text box).

Serial interface mode

In this mode the program reads data from the selected COM port (from 1 to 9) and simultaneously makes a copy in a file called 'NetzLog_yyyyymmdd_hhmmss.txt' on the PC for subsequent processing and inspection. Logging begins when the 'Start' button is clicked.

Conclusion

By combining two previously-published Elektor projects and adding a little new software the author managed to create a useful device while avoiding the temptation to reinvent the wheel. The combination is more than the sum of its parts, providing not just real-time autonomous recording of the grid frequency, but also a complete solution for monitoring and analyzing the quality of the grid supply over longer periods. The software is available in 50 Hz and 60 Hz ver-

sions, and, since the source code is published, it is easy to make changes to it yourself. One possibility would be to build in an electronic real-time clock so that timestamps could automatically be added to the logs: perhaps the circuit described in the article 'Time Transporter' in the July/August 2011 issue of Elektor [5] will provide some inspiration.

(130233)

Web Links

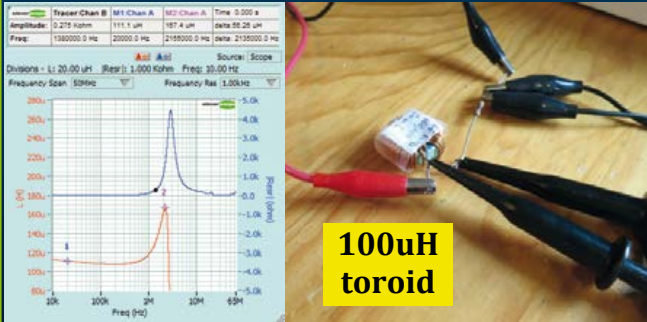
- [1] 'Grid Frequency Monitor', Elektor January 2012: www.elektor-magazine.com/110461
- [2] 'USB Data Logger', Elektor December 2011: www.elektor-magazine.com/110409
- [3] Software and source code: www.elektor-magazine.com/130233
- [4] C compiler (demonstration version): www.wickenhaeuser.de
- [5] 'Time Transporter', Elektor July/August 2011: www.elektor-magazine.com/110285

LCR + Stability Measurement

Use the Cleverscope FRA panel to easily auto plot Gain/Phase, Impedance, Capacitance or Inductance vs Frequency. Display the Gain and Phase Margin. Check for instability.

Easy As, with Cleverscope.

See our FRA tutorial video to show you how to verify your operating power supply or amplifier design. Check the impedance of your DC buses. Verify magnetics you have wound. 80 dB dynamic range! 0 - 65 MHz isolated Sig Gen.



100uH toroid

Streaming 100 G samples to disk ♦ **Frequency Response Analysis** ♦ Protocol Analysis • Symbolic Math • Matlab Interface • 80 dB dynamic range • 100 MHz Bandwidth • Tracking Zoom • 0-65MHz isolated sig gen • Video Tutorials



CS328A-FRA
14 Bit MSO



www.cleverscope.com

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



FREE Software



ONLY \$90.24 with custom logo engraving

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

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

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

FRONT PANEL EXPRESS

FrontPanelExpress.com
1(800)FPE-9060

Create Complex Electronic Systems in Minutes Using Flowcode 6

FLOWCODE6

Design → Simulate → Download



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



New in Version 6:

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

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

Elektor Radiation Meter using PIC

By **Reinier Ott**
(Netherlands)



Another solution for an existing design

The *Improved Radiation Meter* from the November 2011 issue of *Elektor* is a handy and affordable measuring instrument for the measurement of various types of radioactive radiation. The original design was built around an ATmega88. That it can also be implemented with another controller and at the same time acquire a few more features, is demonstrated by this project.

This radiation meter is based on the 'Improved Radiation Meter' by Burkhard Kainka, which was published in the November 2011 issue of *Elektor*. I purchased the Elektor kit for this project and experimented extensively with it. Based on this experience, I designed a new version, which I decided would be based around a PIC, instead of the ATmega88 in the original. I copied the sensor board in its entirety, with the only difference that I doubled-up on the BPW34. While writing the software for the PIC, using Flowcode, I also added a number of new functions. A lot of useful information on the original project may be found on the Elektor Forum in the Test & Measurement section.

Schematic

As already mentioned, the preamplifier section is completely unchanged and we also continue to use the existing circuit board number 110538-2

for this. This part of the circuit is therefore also not drawn in the schematic for this PIC radiation meter, which is shown in **Figure 1**.

The entire hardware is built around a PIC16F88, which is provided with a crystal of 19.6608 MHz, from which a run-time clock, with a time accurate in seconds, is derived via the internal prescaler of the timer.

The output signal from the preamplifier is connected to connector J1. This then goes via resistor R2 to pin 17 (RA0), which is configured as an analog input. A zener diode of 5.1 V (D1) has been added to protect the analog input. Although the output voltage of the preamplifier can never exceed 5 V (even though it is powered from a 9-V battery), it was nevertheless decided to include some input protection because the design is general-purpose and it is possible to connect a completely different type of sensor to it.

Just as with the original radiation meter, a 2x16 characters LCD is used for the read-out, which is controlled from a few pins of port B (RB0 through RB5) of the PIC. P2 is used to adjust the contrast of the LCD.

In addition to the on/off-switch SW1, there is a mode- and a reset-pushbutton (BT1 and BT2). The purpose of connector J2 is for in-circuit programming of the microcontroller. This programming however does necessitate the removal of jumpers JP1 and JP2. The removal of JP2 is necessary to prevent the output of the 78L05 from being loaded incorrectly by the external power supply of the debug-board. JP1 ensures, during normal operation, that the circuit can be reset with BT1, but while in external debug-mode this has to be removed, otherwise the required pro-

gramming signal on MCLR will be attenuated by R5 and R6, which will prevent in-circuit flash programming of the PIC.

New, when compared to the original circuit, is the audio part comprising an LM386N output amplifier and a small speaker, which makes the detected radiation audible. Visual indication is provided by LED LD1. Pin 1 (RA2) takes care of driving the LED and the speaker.

The backlight of the display can be switched on using jumper JP3. This, however, draws so much current from the 9-V battery that the circuit can oscillate because of instability in the 78L05 power supply regulator that is used here.

Without the backlight the entire circuit draws about 20 mA, which means that a regular 9-V battery will last about 8 hours.

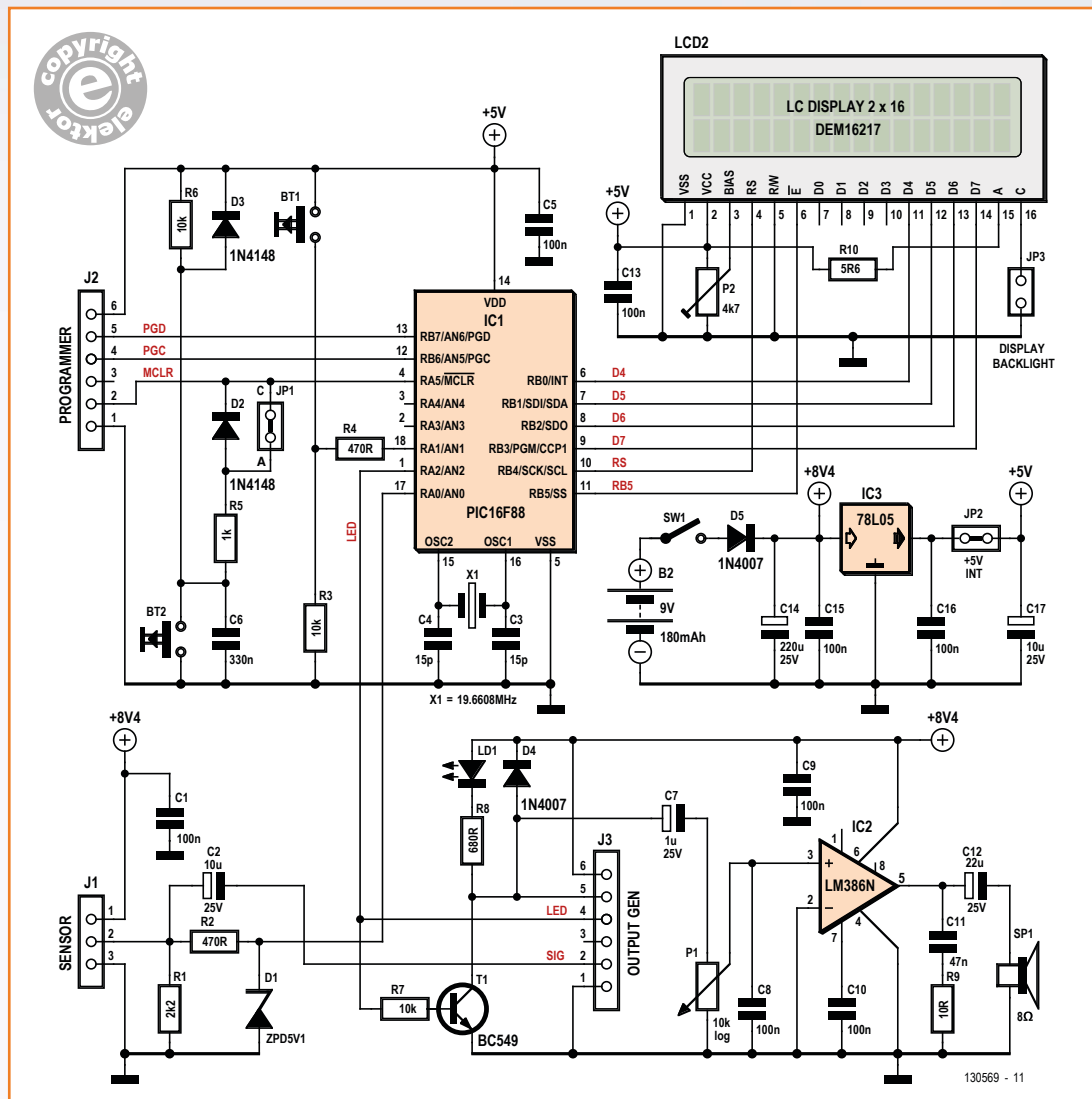


Figure 1. The schematic of the PIC radiation meter with, naturally, a PIC16F88 at its center.

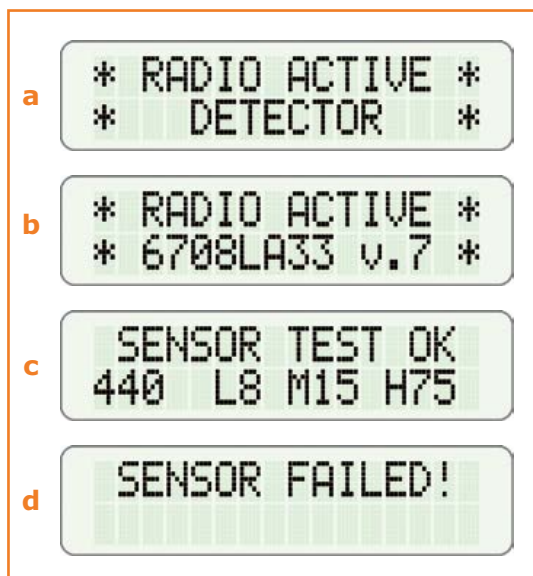


Figure 2. The various screens when the radiation meter is started.

Operation

After the power supply voltage is switched on (the PIC is reset at the same time) a short introduction is shown on the display (**Figure 2a**), followed by some information about the author and the firmware version (**2b**).

During this time the circuit determines the noise level (comparable to the original design) and the final introduction screen (**Figure 2c**) displays a few values:

- noise level in bits, in this case 440 (10-bit ADC -> 0 through 1023)
- The value 'Low threshold' L, in this case the value 8, which is added to the noise level in order to obtain the most sensitive measuring signal.



Figure 3. a) Measurement display, b) intensity mode, c) manual setting of the threshold levels.

- The 'Moderate threshold' M, in this case the value 15 as an average sensitivity of the instrument.
- Finally the 'High threshold' H, for measuring the most energetic radiation.

(Note that the values L, M and H can be changed at any time through pushbutton combinations and these values, just as in the original design, are stored in the EEPROM memory.)

When the sensor is faulty or is not connected, the error message shown in **Figure 2d** is displayed. In this case the red LED will also be turned on continuously.

After the introduction, which takes a total of 5 seconds, the instrument begins to make measurements.

The instrument will now indicate a few values on the display (**Figure 3a**). The top line shows successively radiation level L (low), M (moderate) and H (high) as measured pulses (indicated in a manner comparable to a normal GM counter tube). At the same time each pulse will briefly light up the red LED and there is an audible 'click' from the speaker (the sound level of which can be adjusted with P1).

The bottom line shows the elapsed time (since the intro) and the number of measured low-pulses per minute (a new value appears after each minute). The measured pulses and the time are updated every second.

Operation Pushbuttons

The PIC can be reset with pushbutton BT2. The intro-procedure starts immediately after releasing this pushbutton. If, during the first three seconds of the intro, the other pushbutton (mode, BT1) is also pushed, the program switches to a so-called pulse-speed mode (pulses per second), suitable for measuring radiation intensity (**Figure 3b**). The top line then shows the number of measured pulses per second of the most sensitive radiation range (L). The bottom line shows the same, but this time in the form of an 'intensity bar'.

Programming radiation level

The programming of the 3 radiation levels is possible by first pressing (and then continue to hold) BT1 (mode) and subsequently pushing BT2 (reset), then release BT1 approximately 3 s into the intro. Now all the levels (L, M and H) can

be programmed one after the other (pressing BT1 increases the value by 1 every second, the value is 'frozen' after the button is released), see **Figure 3c**.

The program will similarly work through the other values and finally will show that the new values have been stored. This can also be seen on the display after a reset since these values are shown during the intro (**Figure 2c**).

Software

The software has been developed entirely in Flow-code 4, where the various tasks of the software have been implemented in separate macros (sub-routines) as much as is possible to ensure optimal readability.

The basic program structure comprises 3 parts (modes), where it starts with the intro-part (Initialization). This intro is responsible for determining the average noise level (the method used is identical to the original design), however here a 10-bit ADC is used, which results in 1024 signal levels, to obtain sufficient resolution for the signal to noise ratio. In practice the noise signal will have a value of about 440. This is also used to decide whether the sensor is 'valid'.

The threshold value for this is 100. If this condition is not satisfied (= no or faulty sensor) then this is shown as an error on the display and the program will not enter the measuring loop. The intro-part is also responsible for the definition of the 10-bit measurement threshold values L, M and H. These values are calculated from the average noise level (also a 10-bit value), increased with a fixed signal threshold value, which for each measurement level is stored as a unique 8-bit value in the EEPROM of the PIC. With this, the smallest threshold value will give the highest sensitivity of the sensor.

From the intro, using the mode- and reset-buttons, a selection can be made for the desired operating mode:

- Normal pulse counting mode for the 3 radiation levels;

- Radiation intensity (pulses per second) with intensity bar;
- Programming mode for the 3 separate sensitivities L, M and H.

After the intro has been completed, the main program according to the desired selection is immediately invoked.

It turned out that it was very important to keep the polling-loop for reading the signal as short as possible to maintain sufficient speed in the measurements. This is important so that pulses that arrive in relatively quick succession can be detected.

From testing it proved possible, when using

a sufficiently strong sample (radium paint from an old alarm clock), to reach a measurement value of more than 6000 pulses per minute. Here we have to make the comment that generally the number of measurements that can be detected reliably is about 0.5 x the polling-loop frequency, in order to ensure sufficient margin so that no pulses will be missed. This therefore means that the upper limit of reliably detecting the pulses is around 5000 pulses per minute. (Not including the loss of speed of

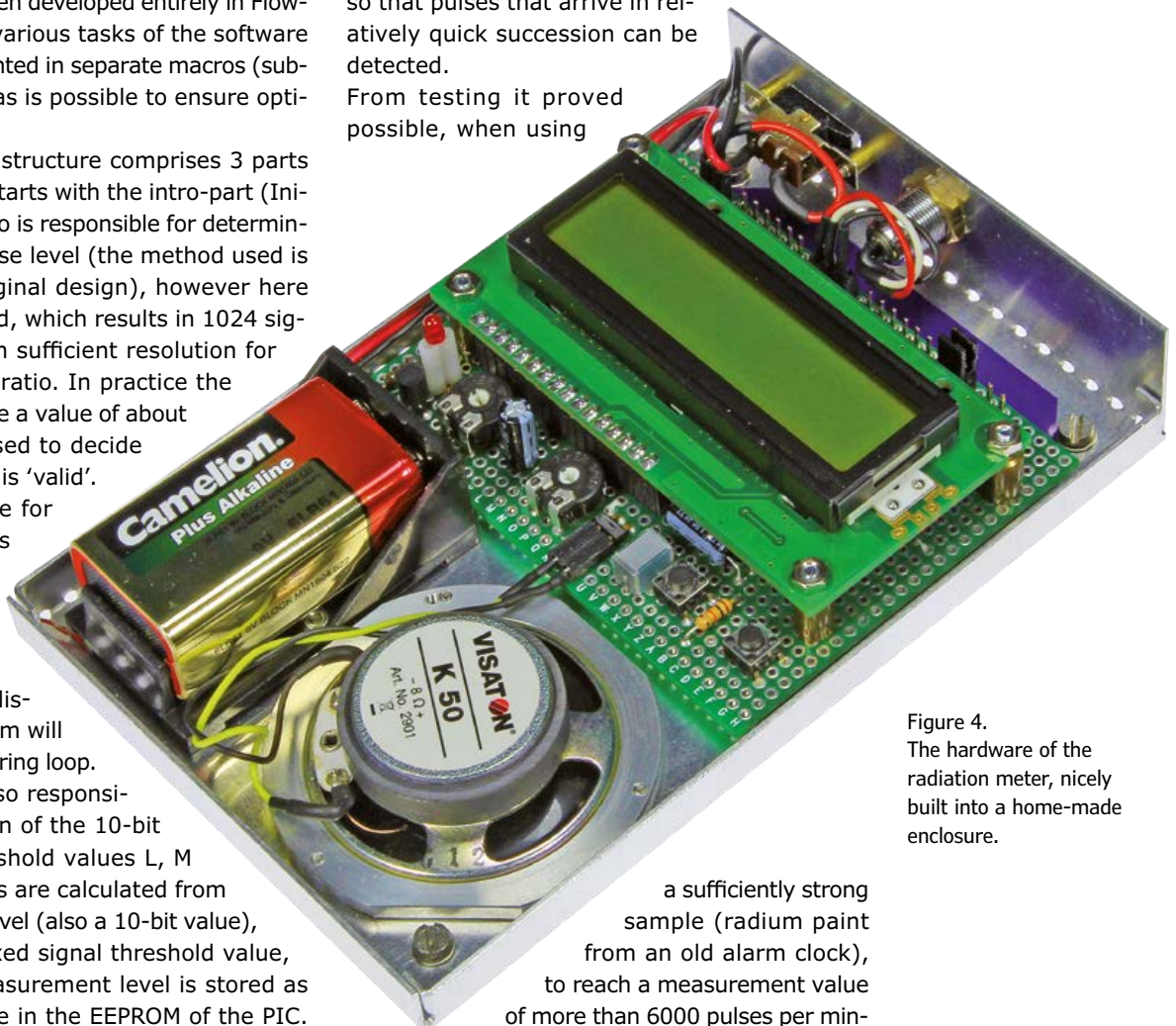


Figure 4. The hardware of the radiation meter, nicely built into a home-made enclosure.



Figure 5. The (also home-made) enclosure for the probe.

the PIC for displaying the values on the display, the speed of the polling-loop in the software was measured at around 9 to 10 kHz.)

The measured analog signal is presented to PA0, is converted to a 10-bit value (INT definition in Flowcode). This value is first compared to the sum of the noise level and the L value (low signal threshold level), as determined during the initialization phase. Only when a signal has been detected which meets this condition, will the M and H levels be examined to see whether the signal is powerful enough to exceed these threshold levels too.

The order adopted here is important to ensure that the PIC runs through as few lines of code as possible during the actual measuring loop. When a sensor signal has been detected the corresponding variable is increased by 1 (incremented), the value of which is updated every second on the display. With this method you have to take into account that an H-signal will also be counted as

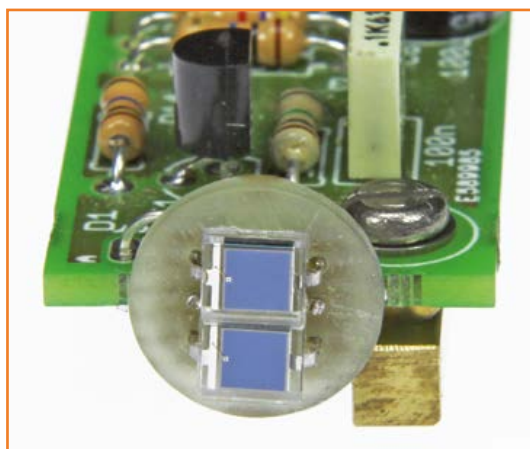


Figure 6. The double sensor, the front surfaces of which are nicely aligned.

an L- and M-signal, and an M-signal will also be counted as an L-signal. This means that when an L-signal has been detected it is possible that it was strong enough that it could have been considered an M- or even an H-signal. In this way the high values are also included in the lower values. The entire program runs on the basis of a timer-interrupt, which is set to a fixed frequency of 75 Hz. This is then directly used to calculate the displayed time and the 'refresh-rate' of the display is fixed at 1 s so that the measuring speed of the instrument is affected as little as possible by the display indication routines. At the same time the necessary (intermediate) calculations are carried out during this phase. A small piece of C-code, implemented directly into the Flowcode display macro, provides a nicely formatted time display in the form of 'hh:mm:ss'.

As already described, the program also has an operating mode for displaying the pulse speed. The measured value is simultaneously visualized as an intensity bar (gauge bar). The length of the bar indicates the measured radiation intensity. It was decided to implement the intensity bar with something approximating a logarithmic scale. In this way it is possible to give a clear indication despite the limited resolution of the 16-character display. Since by default there are very few mathematical functions available for the PIC, a very simple piece of C-code has been included in the implementation part of Flowcode ('supplementary code' under 'project options'):

```
short GaugeValue(int p)
// Position of Gauge bar
{
    short y;
    if (p>1000) y=16; else
        if (p>500) y=15; else
            if (p>250) y=13; else
                if (p>120) y=12; else
                {
                    y=p/10;
                }
    return (y);
}
```

Finally we still need to mention that the intro part of the software also contains some 'built-in data'. Here you can store the specific details of the owner of the instrument. Each time the instrument is switched on or is reset, the post

code and house number, for example, of the owner can be shown on the display. In the version of software that can be downloaded from [2], the final block of the Flowcode-Intro-macro contains '1234AB56 v0'. Here you can enter your own information and version number.

Construction

The sensor, together with the sensor circuit board, are housed in an hermetically, light-proof enclosure (**Figure 5**). The window, a disc of aluminum foil, is attached to the front in such a way that it is easily swapped (using a threaded ring). This has the advantage that it is very easy to experiment with different foils (even ordinary household aluminum foil is already available in varying thicknesses).

Note. When mounting both of the BPW34 sensors it is important that their packages push slightly against the aluminum foil window, to prevent (as was also mentioned in the original article) unwanted microphone behavior.

This 'microphone effect' can be quite persistent. The reasonably elastic mechanical mounting of the measuring PCB with respect to the aluminum foil window does of course not help much when trying to suppress these artifacts. With the construction method described here, an improvement was found by putting a small amount of silicon grease between the surface of both BPWs and the aluminum foil. The grease provides some mechanical damping. However, the layer of silicon has to be as thin as possible since it negatively affects the radiation transmission. It is also recommended to rigidly attach the circuit board on all its sides inside the enclosure (even more so when the measuring probe is implemented as a separate unit).

As can be seen in **Figure 6**, the BPW is implemented using two devices, i.e. they are connected in parallel, in order to obtain an increased sensor surface area. In the original article this idea was already discussed and there was the caution for an increased parasitic capacitance at the input when multiple sensors are connected in parallel. However, experiments showed that this negative effect was outweighed by the increase in sensitivity of the instrument (all the more because generally only relatively weak radioactive sources are being measured). In addition, the sample-frequency is already somewhat limited (see the description of the hardware) and software, because the data processing takes place inside a microcontroller.

(130569)

Weblinks

[1] www.elektor-magazine.com/110538

[2] www.elektor-magazine.com/130569

Flowcode is available through Elektor Store, www.elektor.com

Pololu

Robotics & Electronics



Optical Encoder Pair Kit for Micro Metal Gearmotors

ITEM #2590

\$8⁹⁵

Extends only
5 mm past
motor

Compatible Gearmotors

\$16⁹⁵



Add quadrature encoding to your micro metal gearmotors!

- 3- and 5-tooth wheels included
- 3.3 V and 5 V versions available
- Works with our growing assortment of gearmotors with extended back shaft (gearmotors sold separately)

Premium Jumper Wires

STARTING AT

\$2⁹⁹/pack



Make prototyping connections quickly and easily with these high-quality jumper wires, available with male or female terminations in a variety of lengths and colors.

38 kHz IR Proximity Sensor

ITEM #2460

\$5⁹⁵



- Typical sensing range up to 24" (60 cm)
- Fixed-gain modulated IR detector
- Small size (0.4" × 0.6")

Step-Up/Step-Down Voltage Regulator S18V20ALV

ITEM #2572

\$15⁹⁵

80% - 90%
typical efficiency



- 3 V to 30 V input
- 2 A typical max output current
- Adjustable 4 V to 12 V output can be above or below input voltage

Pololu Basic 2-Channel SPDT Relay Carrier with 12 VDC Relays

ITEM #2487

\$8²⁵



Control two SPDT relays easily with logic level signals. Available with 5 V relays or without relays. Single carrier also available.

Simple Motor Controller 18v15

ITEM #1377

\$39⁹⁵



Highly configurable DC motor controller that supports four interface modes: USB, TTL serial, analog voltage, and hobby radio control (RC).

Dagu Wild Thumper 6WD Chassis (With 75:1 Steel Gearboxes)

ITEM #1563

\$249⁹⁵



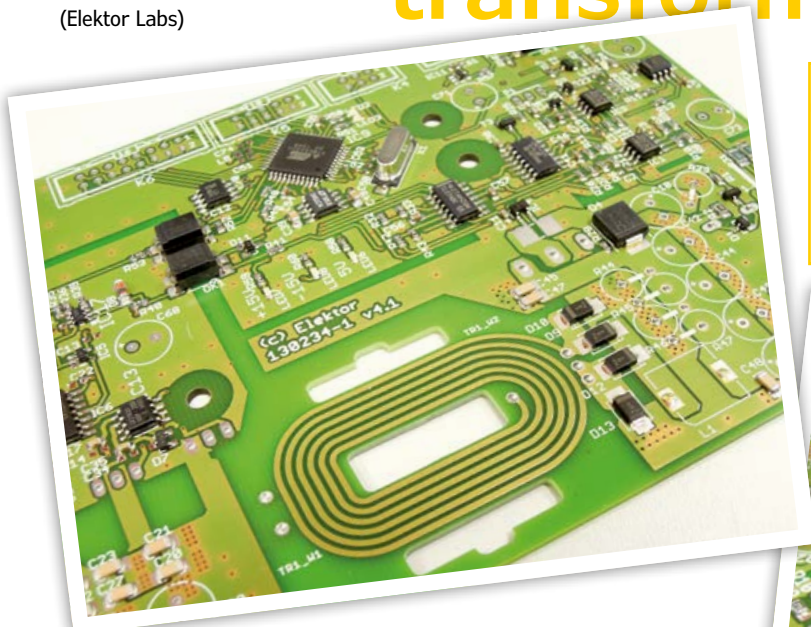
- Rugged aluminum body
- All-terrain suspension
- Available in 4WD and other gear ratios and colors

Take your design from idea to reality.

Find out more at: www.pololu.com

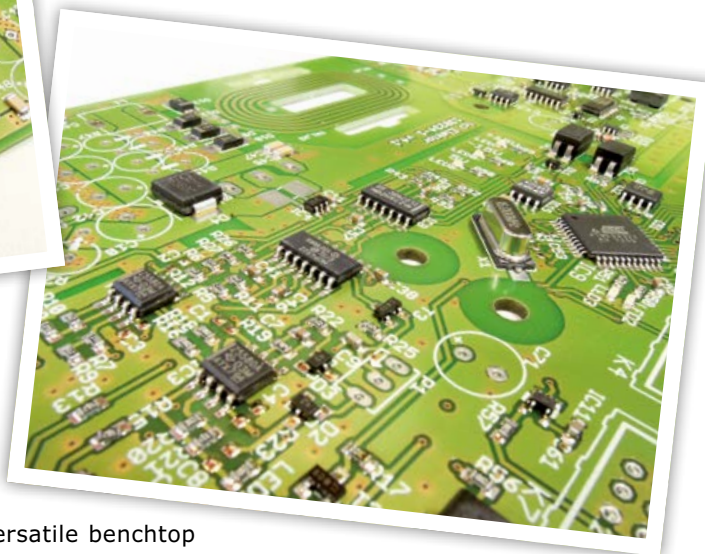
PCB real estate 'transformed'

By **Thijs Beckers**
(Elektor Labs)



Preliminary, Tentative Specifications:

- PC PSU supplies raw 12-V input
- Output voltage adjustable from 0 to 30 V
- Power output 30 watts max.



A power supply was born at Elektor Labs. Arne Hinz has spent his time as an Elektor Labs trainee well, designing a versatile benchtop power supply with the output electrically isolated from the AC power outlet based on a design by Martin Christoph at ISEA, RWTH Aachen, Germany.



His design incorporates an interesting implementation where the essential step-up transformer is designed 'in-board' as proposed by Christoph. The input voltage is stepped up using a switch-mode regulator and this 'PCB transformer' to supply a voltage for the output circuit.

Two 7-segment displays show voltage and current, which are set using two encoders. We cannot reveal all the details for the circuit at this point, but we do have some photographs to show you and whet your appetite.

(130363)



Join the Elektor Community

Take Out a GOLD Membership Now!



Your GOLD Membership Contains:

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



ALSO AVAILABLE:

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



Take Out Your Membership Now at www.elektor.com/member

Connect with us!



www.facebook.com/elektorim



www.twitter.com/elektor

What's up with this cap? (2)

By Thijs Beckers
(Elektor Labs)



Last month I reported on an oddball capacitor imprint. While checking the semi-kit for our ADAU1701 Universal Audio DSP Board [1] I stumbled upon this very odd thing going on with the 0.1- μ F capacitors in the kit. As you can see in the photo, they have a “105” marking on one side and “104” on the other side. Every 100-nF capacitor in every sample kit I checked showed the same 104 / 105 print shown here, so it was not just a misprint on one of them.

The relatively simple answer to this conundrum came from our supplier. As can be seen from the marking illustration taken from the datasheet [2], the front shows the capacitance code (104: “10-with-4-zeroes” = 100 000 = 100 000 pF = 100 nF = 0.1 μ F) and the back shows the 3-digit date code (105: “2011/5” is our educated guess). In this case the date code and capacitance code wreaked havoc here at Labs. Without (1) a way to determine the front or back side of the device and (2) without knowing the other codes AVX employ, or (3) being aware AVX is a capacitor manufacturer in the first place, it is hard to declare the

actual capacitance with dead certainty and not being dragged over to the LCR Meter by fellow lab workers.

Calling out to our well-informed readers and professionals last month resulted in some response on the topic. Most answers correctly surmised an issue with the date printing (but still not knowing what’s what); some went as far as suggesting a secret auto-destruct time, a new notation like 105^{-1} or a new base value adopted for the famous 1-0-x system to indicate the value or even an April 1 spoof from EIM (Elektor International Media).

No harm befell on any of the ADAU1701 DSP semi-kits shipped or due for shipping and you can safely solder in those yellowish 100-nF Cs. Thank you all for your helpful and amusing answers and your involvement in our quandary.

(130523)

Web Links

- [1] www.elektor-magazine.com/130232
- [2] www.avx.com/docs/Catalogs/skycap-sr.pdf

Professional Quality
Trusted Service
Secure Ordering



Elektor PCB Service at a glance:

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

Delivery
from 2
working
days

Post and Win

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

For the year 2014 we decided to offer a small present in return for a project posted on Elektor.Labs. Every month we choose a gift and all you have to do is post a project to receive it!* Check the banner on the Elektor.Labs homepage to find out what this month's gift is.

Live Q & A



Back in the nineteen eighties Elektor organized weekly one-hour telephone sessions allowing readers to ask questions and sometimes put themselves to shame in regard to published projects. One poor soul at the wrong end of the line and flanked by piles of magazines and TI books would try to

answer these questions live on the phone. It was rumored that some callers got connected through to a 600-ohm resistor for advice. This cool service was abandoned due to several factors including overheating ears, loss of voice, and painful elbows (i.e. phone related RSI symptoms overall) but it was never forgotten. And now it is back! Not exactly as a phone service, but online in the guise of free webinars. We plan to organize a Q & A session once a month when you can ask your tech questions. Isn't Internet wonderful? Announcements will be made on Elektor.Labs and in our dot-POST newsletters.
www.elektor-labs.com/qa1



Copy-Paste Don't Work

Often Elektor.Labs users prefer typing their project text in a text editor like Word or Notepad prior to copying it on to the .Labs project page. The advantage of doing so is that you will not lose your text when bits flip or drop on the Internet or when you forget to save your project before surfing off. We actually recommend working

this way. However, every once in a while we do receive a complaint from a user saying that it is not possible to paste text in a project's text field. Clearly something must be wrong with the Elektor.Labs website!

I would not venture to say that our website is perfect and that there are no problems at all, but this issue is in fact browser related. A security feature to be more precise to prevent malware programs from copying-pasting themselves or other unwanted content all over the Internet. In case you didn't know: when enabled, small JavaScript programs embedded in webpages can

have access to your computer's clipboard and use it to fool around, which is why some browsers disable this option by default. If you run into this problem, change your browser's security settings. Depending on your browser, when you try to paste something, it may also ask you to allow clipboard access for the current active page.

(130524)

* 1 (say, one) gift per user/poster per month - www.elektor-labs.com/



We're celebrating a new look — and the next 25 years.



MCU-BASED COLOR DATA ACQUISITION

Our robot's MCU captures and processes camera images of cube facets



+ robot control + search algorithm execution = solution

IN THIS ISSUE

- Weekly Code Challenge | Q&A: Pro Electronics Entrepreneur
- Rubik's Cube-Solving Robot | R-Pi I/O Board | Robotic Neural Networks
- Battery Basics for Engineers | LED Characterization | Cooling Servers
- | Tips for Using BOM Files | ■ Electronics Beyond Silicon

25% OFF CIRCUIT CELLAR

Whether it's **programming advice** or **design applications**, you can rely on *Circuit Cellar* for solutions to all your electronics challenges. **Raspberry Pi**, **embedded Linux**, **low-power design**, **memory footprint reduction** and more! Become a member, and see how the hottest new technologies are put to the test.

JOIN TODAY!

www.circuitcellar.com/sepN13



DesignSpark Tips & Tricks

Day #10: Custom Outline Models

By **Neil Gruending**
(Canada)

Let's import our PCB back into our 3D model and have a little fun.

Over the last couple of days (that mysteriously lengthened to months in Elektor) we designed a circuit board to fit in a Hammond 1551N enclosure using DesignSpark PCB and DesignSpark Mechanical. Today we'll import our board back into DesignSpark Mechanical and have a little fun along the way.

Import the PCB into DesignSpark Mechanical

We had finished the board outline in the last installment so now it's time to export an IDF file from DesignSpark PCB by using the "Output->DesignSpark Mechanical (IDF)" menu. This will open the DesignSpark Mechanical IDF export window where you can specify a name for the file and the board thickness. Note that the export will use a default component height of 1mm unless the components specify a different height in the library. We'll see an example of that later.

because we just need to line up the new board outline to our original model. The first step is to move the move anchor point to a board corner using the Anchor tool. Then select the Up To tool and select the same corner on the original PCB outline. Both of these tools are available in move mode in the left side of the drawing window.

The updated model should now look like **Figure 2**. Everything looks fine except that our mounting holes have squares over them. That's because our mounting holes are components on the PCB and DesignSpark PCB applied the default 1mm component height rule to them which gives us the square boxes over the holes. A quick edit to remove the square will actually update both of the holes automatically because DesignSpark Mechanical has also imported the component structure from the PCB. I expanded the model structure in **Figure 3** to show the mounting hole components. The figure also shows what the imported PCB looks like after being cleaned up. Now let's add some components to the board to get a complete rendering of our design.

Adding PCB components to our model

Now let's add some components to our board and see what happens. I chose to add a few SOT23 transistors and a couple 0603 resistors like in **Figure 4** and then imported the board into DesignSpark Mechanical which looks like **Figure 5**. I had problems getting the PCB components to import correctly into DesignSpark Mechanical while I was playing with the playing with the component heights though. The solution ended up being to delete all of the files in the IDF export directory.

Figure 5 also shows an example of what importing a PCB with different component heights looks like in DesignSpark Mechanical. The mounting holes have the small box drawn around them like before and the 0603 and SOT23 patterns are 0.5 mm and 1.12 mm tall respectively. The trick is that you have to specify the component heights in DesignSpark PCB by adding a value named "Height" to the component properties. I

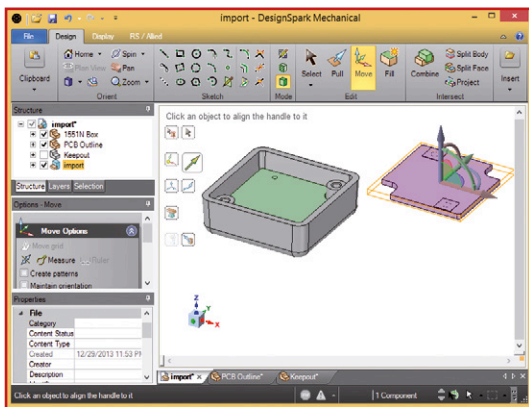


Figure 1. Imported board outline.

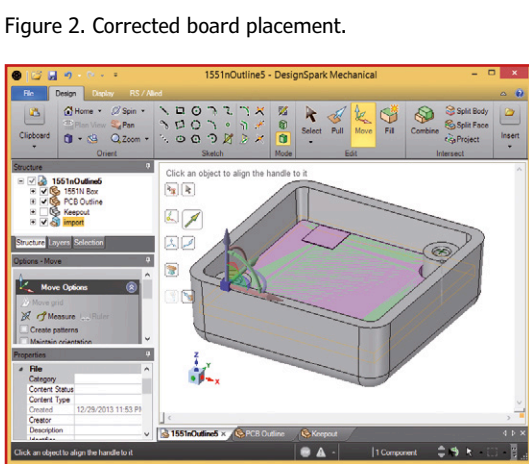


Figure 2. Corrected board placement.

Now let's import the IDF into our DesignSpark Mechanical model using the file import tool (Insert tab->File) and you should get something like **Figure 1**. DesignSpark has already selected the board outline for us so now we just need to move it to the right position in the enclosure. This is really easy in our case

recommend doing that in the component libraries so that you don't forget later. My board was already set to metric units so I set the height to the desired value without any units, i.e. I entered 0.5 mm as 0.5.

But what if you wanted to make our imported 3D PCB look more realistic? The first thing we'll need are the 3D models for our components. It's possible to draw them yourself in DesignSpark Mechanical but in this case I will use STEP models downloaded from 3D Content Central [1]. DesignSpark also has a lot of 3D models available as part of Modelsource [2]. The only real requirement for the models is that they use the same co-ordinate origin and orientation as the PCB components so that the new 3D models will line up properly. For example, I like to use the center my PCB footprints as the component origin so I used 3D models that also used their bottom center point as the origin. Note that DesignSpark Mechanical won't let you edit STEP models so sometimes you might have to try several different models before finding one that will work.

Now we're ready to update the 3D PCB model. Open the imported PCB with the "Open Component" command which will open the PCB in its own viewing window. Now you can select the components to change in the Structure window. Right click on it and select "Source->Replace Component" and then choose the 3D model file you want to use. DesignSpark Mechanical will then exchange the model for the new one and it will also rotate it as necessary. This is why it was important to use models that correspond to the PCB footprint. The final result will look like **Figure 6** after a little bit of editing. Make sure you double check the component 3D model position if its placement is critical.

Conclusion

Today we used DesignSpark Mechanical to make a 3D model of our finished PCB. Next time we will focus on using DesignSpark PCB for more complex designs.

(130575)

Web Links

- [1] www.3dcontentcentral.com
- [2] [www.tracepartsonline.net/\(S\(rhrqx-sieinz4vp45g4eluh55\)\)/content.aspx?fwid=DESIGNSPARK&Lang=&P=](http://www.tracepartsonline.net/(S(rhrqx-sieinz4vp45g4eluh55))/content.aspx?fwid=DESIGNSPARK&Lang=&P=)

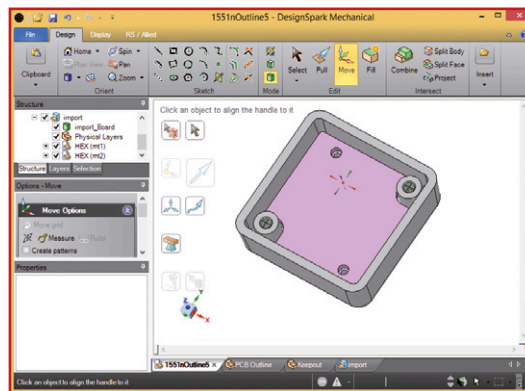


Figure 3. Finished board import.

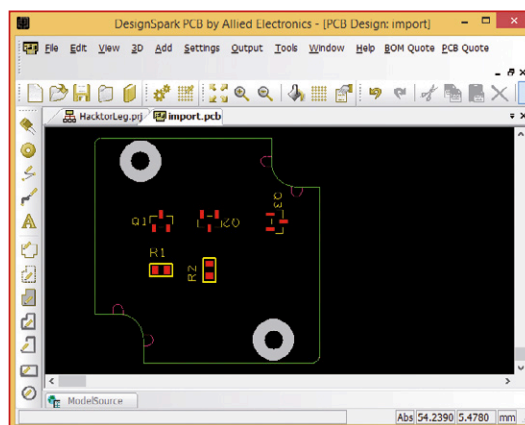


Figure 4. Components added to PCB.

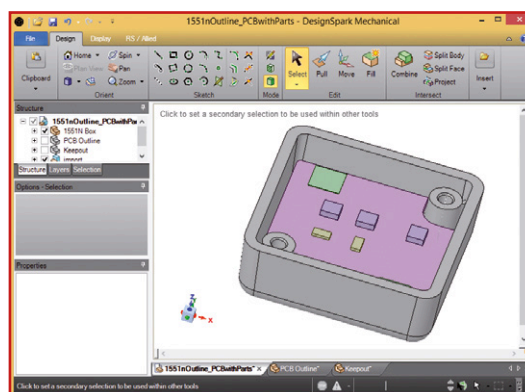


Figure 5. 3D PCB model.

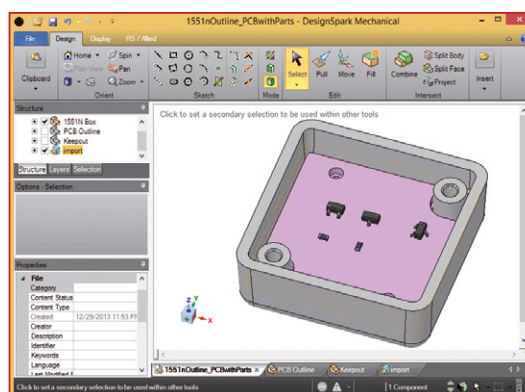
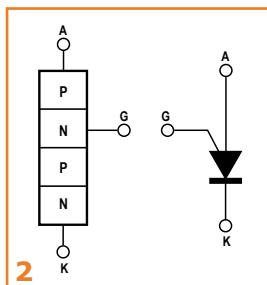
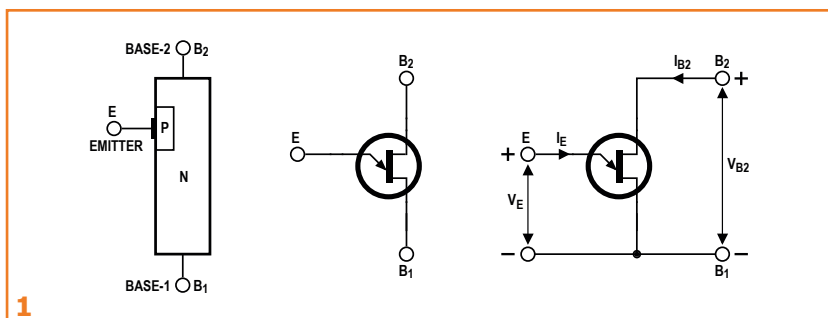


Figure 6. Final PCB model.

Unijunction Transistors Weird Component #5

By **Neil Gruending**
(Canada)

Mention the word transistor and I bet most people immediately think of the usual bipolar variety. Unijunction transistors (UJTs) aren't common anymore but a few decades ago they achieved widespread use in low frequency oscillators and silicon controlled rectifier (SCR) firing circuits. Let's take a look at how these devices work—and at a modern replacement.



A traditional UJT is a 3-pin device with a single (!) PN junction inside. Its construction, circuit symbol and basic circuit arrangement are shown in **Figure 1**. Two of the pins are used for the base connection and are labelled B1 and B2. They connect to either side of a bar of N-type silicon with a well of P-type silicon in it for the third connection called the emitter (E). When the UJT is off there's a resistance between the base pins, and the emitter acts as a diode. The base construction acts as a voltage divider for the diode so that no current will flow into the emitter until the voltage exceeds the internal base voltage. Once the emitter voltage increases enough to start conducting, the UJT will switch on and create a low resistance path between the emitter and B1. This switching point is called the peak voltage

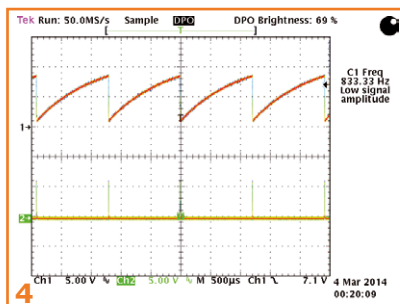
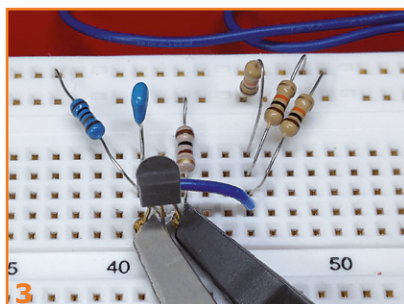
and the UJT will continue conducting until the emitter voltage drops below the valley voltage threshold. The valley voltage is always less than the peak voltage, which gives UJTs their negative-resistance characteristic and makes them great for triggering from short pulses.

A modern replacement for UJTs are programmable unijunction transistors (PUTs). They operate in a similar manner as a UJT but internally they are an SCR with a 4-layer P-N structure. This construction means that PUTs have an anode, cathode and gate connections (**Figure 2**) instead of the usual UJT connections. Just like a UJT, a PUT has a negative-resistance characteristic when the anode voltage exceeds the gate voltage which is programmed with a resistor voltage divider.

I couldn't find UJTs to play with but I did find some 2N6027s which are a PUT so I put together a simple relaxation oscillator (**Figure 3**) and measured its output with an oscilloscope (**Figure 4**). Channel 1 in the oscilloscope trace shows the anode voltage and Channel 2 is the cathode voltage. The anode is charged by a RC circuit and when the threshold voltage is reached the SCR kicks in and discharges the capacitor very quickly as shown by the cathode voltage pulses. This circuit isn't terribly useful by itself but if you used it to trigger an SCR then you could start controlling much larger loads.

UJTs may be almost obsolete but the fact that they are being replaced by PUTs shows just how useful their functionality can be.

(130477)



Elektor and National Instruments Present:

Exclusively for Students!

Professional Hard & Software at a Special Discounted Price!

As the new sales partner of National Instruments, Elektor offers products from the NI platform for the education of students and enrichment of educational institutions. This educational platform brings together hardware, software and teaching materials to allow students an attractive and inspiring learning environment.

Do Engineering – Real-World Solutions for Hands-On Learning

LabVIEW

With the LabVIEW system design software, students can gain hands-on learning through projects and systems in a single environment and thus acquire skills and procedures that are invaluable to any future career they might wish to pursue.



Circuit Design Suite

The NI Circuit Design Suite combines NI Multisim and Ultiboard software for a complete circuit design, simulation, validation, and layout platform. The Suite has features that are specifically tailored to the needs of students, which are useful in the development of electronic concepts.



myDAQ

myDAQ is a low-cost data acquisition device that allows for measurement and analysis of physical signals in any place, at any time. myDAQ is compact and portable, guaranteeing students practical experience outside of the laboratory and use of industry standard tools and methods.



Further Information and Ordering at www.elektor.com/ni-platform

Connect with us!



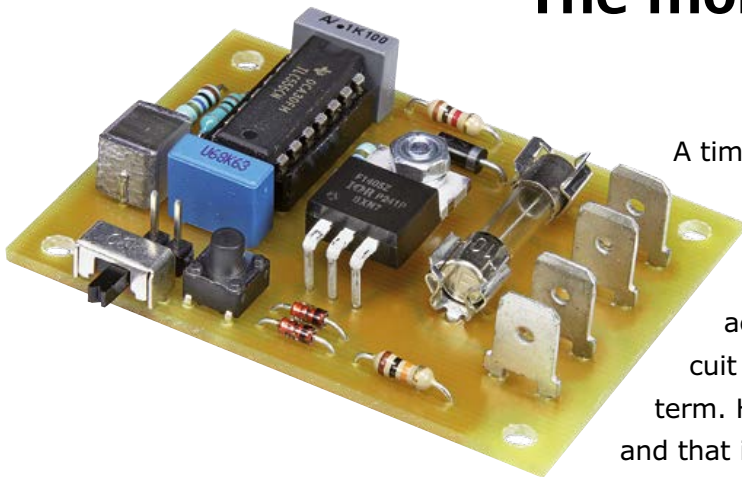
www.facebook.com/elektorim



www.twitter.com/elektor

Zero-Electrolytics 555 Timer

The more resistance the better



A timer circuit that turns on a pump for a short time every five minutes can be easily made using the famous 555 IC. Unfortunately, it's almost impossible to avoid using electrolytic capacitors to achieve such a long period, which makes the circuit less accurate and reliable, especially in the longer term. However, there is one solution to this problem, and that is to use a special 1 giga-ohm resistor!

By **Albert van Dalen**
(Netherlands)

The author designed a timer for his vacuum pump that turned it on for a short period roughly every five minutes. This was just enough to avoid the loss of vacuum due to small leaks. Such a timer can be easily made using the well-known 555 IC (in this case the dual version, 556). The circuit described here stands out by the fact that no electrolytic capacitors have been used for the components that determine the delay. Instead, ordinary film capacitors were used, which have a much lower leakage current than electrolytics. Since the capacitance of this type of capacitor is much lower than that of electrolytics with a similar size, the load resistor has to be a very high ohmic type. In this circuit a resistor of 1 G Ω (giga-ohm) is used! Most electronics hobbyists probably don't even realize these exist. This component appears to be available cheaply, and the author thought it a nice challenge to use it in the design for the timer.

Circuit without electrolytics

The result can be seen in **Figure 1** and requires little explanation. There are two timers, which are connected in series. The first timer (IC1A) is configured as an astable multivibrator, with a period of about 270 s. The components R1, R2 and C1 are used to set this time. By using the previously mentioned 1 G Ω resistor for R1, it is sufficient to use a capacitor of only 0.39 μ F to achieve this period. The second timer (IC1B) has been implemented as a monostable multivibrator.

With the component values given for R3 and C2 it outputs a positive pulse of about 7 s when it gets a falling edge from the output of IC1A. The MOSFET (T1) is turned hard on during this time, powering the motor (max. 12 V, 10 A). The timer section can be switched on or off using switch S1, or it can be permanently activated via JP1. The pushbutton (S2) offers the facility to turn on the motor manually, and to keep it running for as long as the push button is held down. Finally, the power input is protected by a combination of a resistor (R4) and a 12 V zener diode (D1). These ensure that the timer circuit won't be damaged by voltage spikes on the power input, or when the input voltage is a bit higher than 12 V.

Sturdy board

A small, single-sided printed circuit board without wire links has been designed for this circuit (**Figure 2**), measuring 58.2 x 43.6 mm. It is very easy to build the circuit, since only through-hole mounted components have been used. A large area of the PCB has been taken up by the fuse and the 'Faston' spade terminals. The tracks on the PCB have been designed to cope with currents of up to 10 A. The specified fuse holder clips (see parts list) can even cope with 15 A. When switching currents up to 10 A it is not necessary to mount MOSFET T1 on a heatsink, but we would recommend that you leave a gap of a few millimeters between the MOSFET and the PCB. This will prevent the specified $R_{th(j-a)}$ from increas-

Cut the Cord — Power Over Ethernet Touch Panel PC

Habey USA's new PPC-6612POE Power-over-Ethernet Panel PC is powered by an Intel Atom N2800. This touchscreen computer, featuring an 11.6 inch 4-wire resistive touchscreen, is a slim, fanless system is perfect for automation, video camera control, digital signage, and more! Offering a 1.3-MP front facing camera, and two bezel mounted front facing stereo speakers the PPC-6612POE is also perfect as a point of sales device.

With the Power-over-Ethernet port, all you need is a compatible Ethernet cable for both power and data for easy network installation. Additionally, this system is Wi-Fi ready with a traditional 12-V power adapter.



www.habeyusa.com (130458-II)

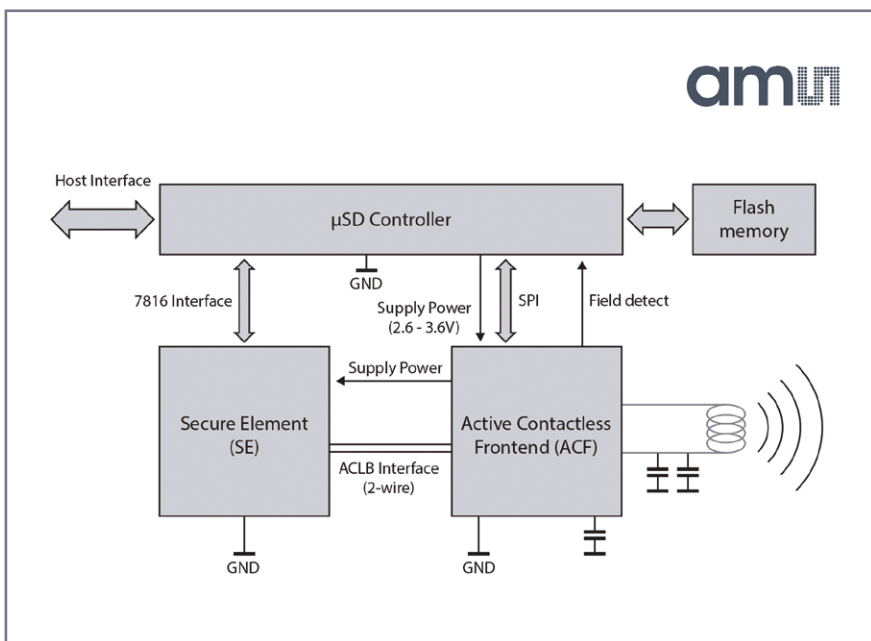
World's First Stand-Alone NFC MicroSD Card Certified for Visa and MasterCard Mobile Payments

ams AG announced that US-based DeviceFidelity, Inc. is using unique ams RF technology in its latest CredenSE 2.10 Near Field Communication (NFC) microSD card to enable secure, certified NFC transmissions between any mobile phone and contactless payment terminals from Visa and MasterCard.

DeviceFidelity CredenSE 2.10 is the world's first NFC microSD card to successfully achieve global payment certifications from both Visa and MasterCard without requiring external booster antenna or device specific attachments. Using the AS3922 chip from ams, an integrated NFC front end with Active Boost technology, CredenSE achieves a typical read range of 4 cm in a mobile phone's microSD slot.

The mobile phone is a notoriously difficult environment for RF and variations between phone models make it difficult to consistently achieve good performance. The stringent requirements for read range compatibility with payment terminals for payment applications cannot be met in small form factors such as SIM or microSD cards with a traditional passive NFC card emulation front end and simple planar antenna.

The DeviceFidelity CredenSE 2.10 is the first commercially produced NFC microSD card that meets EMV standards using only an ultra-small antenna embedded in the card, making distribution and compatibility with hundreds of phone models possible with one easy-to-deploy microSD card.



Active Boost allows for robust tag-to-reader communication at a coupling factor 100 times higher than is possible with conventional passive tag designs. The AS3922 also offers unique Antenna Auto Tuning and Q factor adjustment, which are critical to microSD, SIM and μSIM applications. The IC includes an ACLB interface for communication with the contactless interface of any Dual Interface Secure Element, and DCLB and NFC-WI interfaces for digital communication.

Use of the AS3922 with a 3D antenna also provides for smooth operation with any payment terminal by eliminating the need for the user to hold the phone in any orientation.

In addition to successful performance with Visa and MasterCard payment terminals, DeviceFidelity's CredenSE microSD also provides an option for service providers to deploy mass transit and physical access applications.

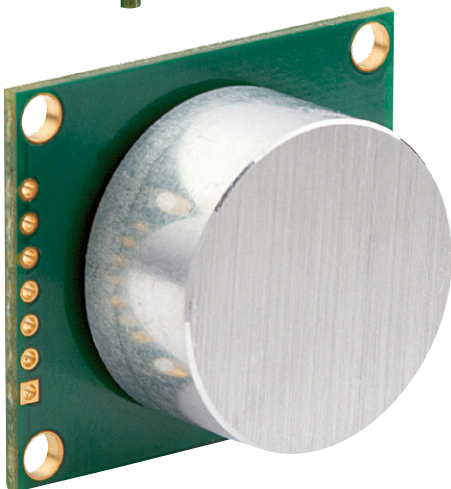
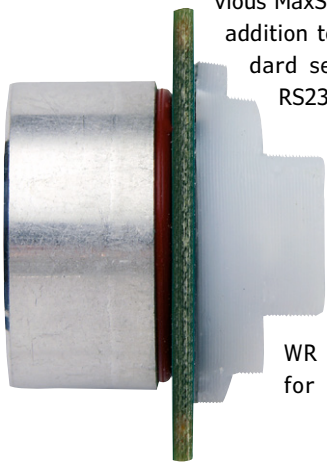
www.ams.com/NFC/AS3922 (130458-V)

High Performance, Low Cost, Ultra-Compact Ultrasonic Sensors

Our most popular sensor is now available in a new design which is physically shorter than any of our current outdoor sensors, allowing easy integration into users' applications. Our new UCXL-MaxSonar-WR-series sensors are flexible, OEM-customizable products intended to be integrated into a customer's system with our horn, or designed for flush mounting into your existing housing. These rugged, high performance sensors are individually calibrated to provide the quality that you have come to expect from MaxBotix.

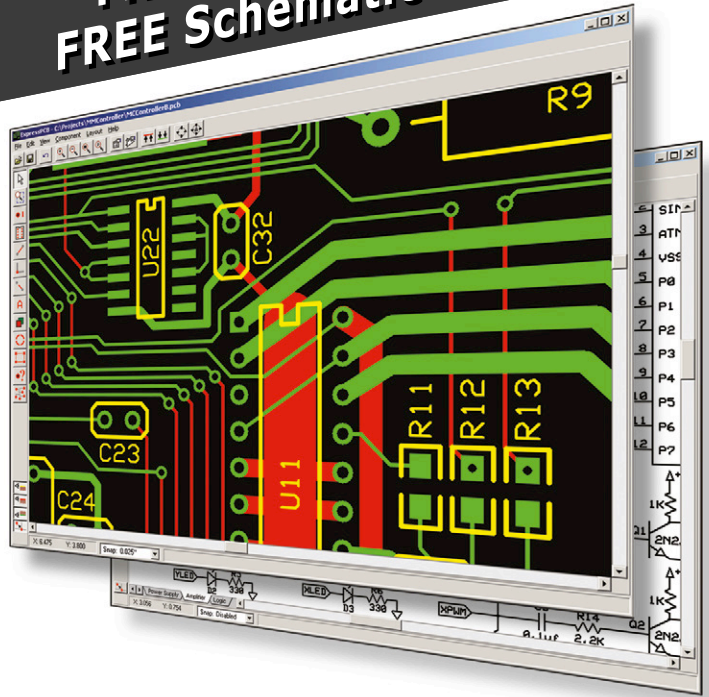
Mounting design recommendations are provided through our 3D CAD models (available in multiple formats) to facilitate your design process. The UCXL sensors are RoHS and IP67 compliant with proper mounting design, and the sensor layout offers four conveniently placed mounting holes for design flexibility.

The UCXL-MaxSonar-WR comes with the easy to use outputs and standard pin configuration of the previous MaxSonar products. In addition to the three standard sensor outputs of RS232 serial (TTL output available upon request), Analog Voltage, and Pulse Width. We are offering four models: MB7260 WR & MB7270 WRA for integration into



\$51^{For 3} PCBs

FREE Layout Software!
FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

expresspcb.com

a customer's housing design which includes our recommended horn, and MB7267 WRC & MB7277 WRCA, flush-mounted designs that do not include a horn.

These sensors feature 1 cm resolution, operational temperature range from -40°C to $+70^{\circ}\text{C}$ (-40°F to $+160^{\circ}\text{F}$), real-time automatic calibration (voltage, humidity, ambient noise), 200,000+ Hours Mean Time Between Failure, an operational voltage range from 3.0-5.5 V, with low 3.4-mA average current requirement. These sensors are also RoHS Compliant and CE Compliant.

Sensors are available for immediate shipment.

www.maxbotix.com (130521-III)



STM32 Nucleo prototyping boards free to users of FTM Board Club website

Broadline technical distributor Future Electronics today announced that it is to make the new range of STM32 Nucleo prototyping boards from STMicroelectronics available free via its FTM Board Club website for design engineers. The FTM Board Club provides a wide range of evaluation, development and prototyping boards free to any OEM engineer based in the EMEA region, provided the project which the board supports has a nominal commercial value. Development boards from many leading franchises are available via the site. Now Future Electronics has extended the FTM Board Club's broad portfolio of products with the STM32 Nucleo base boards, providing users of STM32 ARM®-based microcontrollers with free access to the latest prototyping boards.

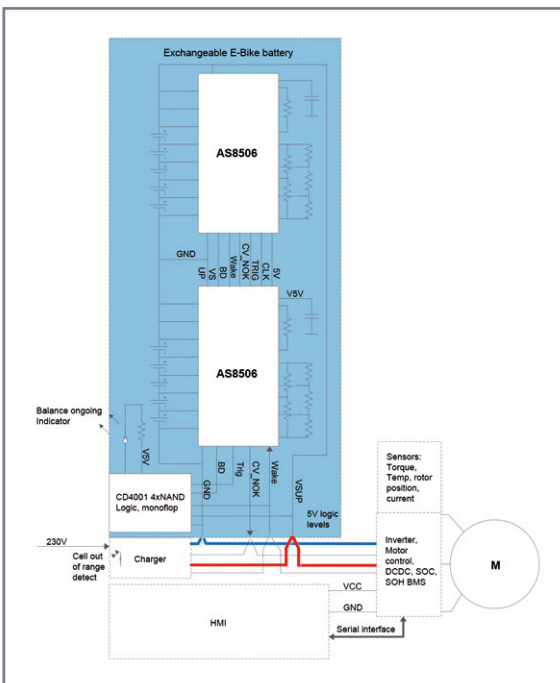
The new STM32 Nucleo boards are compatible with ARM's mbed application development platform. They also include ST Morpho extension headers to allow access to all of the microcontroller's on-chip peripherals, and Arduino headers which accept shields from the extensive Arduino ecosystem, allowing developers to add specialised functionality quickly and easily.

In addition, ST will offer its own dedicated shields supporting functions such as Bluetooth® Low Energy and Wi-Fi® connectivity, GPS satellite positioning, audio recording, proximity sensing and wireless control. Any shield can be re-used with any STM32 Nucleo board and across various projects.

The STM32 Nucleo-F030R8, STM32 Nucleo-F103RB, STM32 Nucleo-F401RE and STM32 Nucleo-L152RE boards are available immediately on the FTM Board Club, free to qualified OEM engineers.

www.my-boardclub.com (130521-I)

Cost-saving System Management Reference Design for Lithium Pedelec/E-bike Batteries



Ams's example design for lithium pedelec/e-bike batteries implements accurate cell monitoring and balancing without the need for a microcontroller in the Battery Management System (BMS). Battery pack and pedelec manufacturers which use the design will benefit from valuable bill-of-materials (BoM) cost savings and a simpler circuit design, compared to batteries in production today.

The ams design is for a 48-V pedelec battery consisting of up to 14 lithium-ion cells. It uses two AS8506 smart cell monitoring ICs, with few supporting components, to monitor the temperature and voltage of up to seven cells each and to implement passive balancing of the cells when charging.

By contrast, conventional BMS designs in pedelecs use dumb voltage monitoring ICs to measure the voltage and temperature of cells, reporting the values to a dedicated battery management microcontroller via a serial communications link. The MCU is required to control safety and protection functions (over- and under-voltage and over-temperature shut-down) and cell balancing.

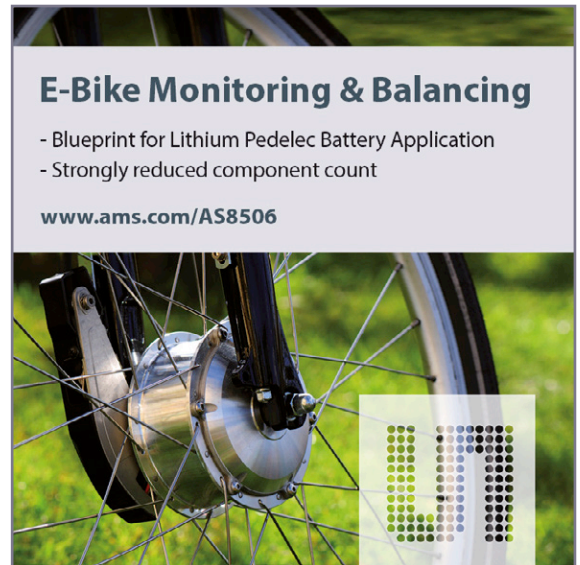
The AS8506 from ams, however, includes built-in logic functions for controlling cell safety, protection and balancing. These functions can easily be configured by the user, with the settings saved in an on-board OTP memory. The device also features integrated MOSFETs for use in passive cell balancing operations. During charging, each cell's voltage is compared to a user-programmable reference voltage threshold. Up to

100 mA may be discharged through the MOSFET from any cell exceeding the threshold, until all cells have reached the threshold and the battery module is fully charged.

This architecture, in which battery monitoring and cell balancing operations are implemented inside the AS8506 voltage-monitoring device, dispenses with the dedicated MCU required in conventional pedelec battery designs. When an AS8506 detects an over- or under-voltage or over-temperature condition, an interrupt signal is transmitted to the pedelec's motor controller IC to complete the required safety shut-down operations.

The reference design can be used in any pedelec or e-bike battery containing up to 14 lithium-ion cells. It can also be extended to supervise more than 14 cells by daisy-chaining additional AS8506 ICs as required. The reference design files are available on request from ams.

<http://www.ams.com/eng/battery-stack-monitor/AS8506-demokit> (130521-II)



E-Bike Monitoring & Balancing

- Blueprint for Lithium Pedelec Battery Application
- Strongly reduced component count

www.ams.com/AS8506

Complete Ecosystem for HMI Development on Intelligent Displays

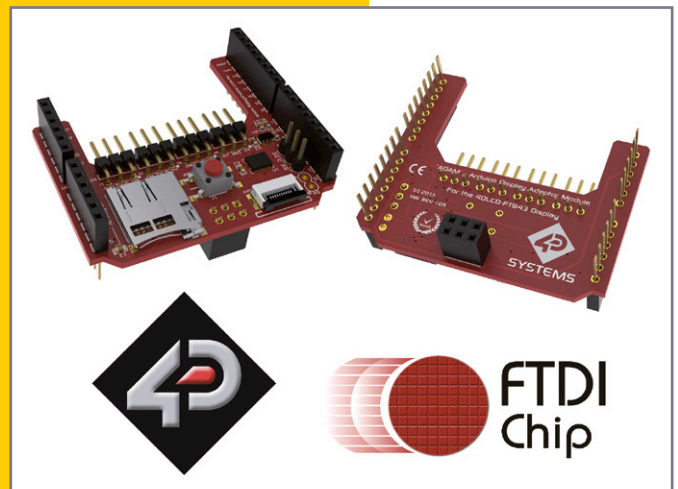
4D Systems and FTDI Chip have now added to the recently announced 4DLCD-FT843 intelligent display solution—which incorporates the award-winning FT800 Embedded Video Engine (EVE), where display, audio and touch functionality are integrated onto a single chip - with the subsequent introduction of two further products. Profiting from the novel object-oriented approach employed by EVE, this combined product offering presents design engineers with a foundation on which to construct compelling new human machine interfaces (HMIs) in a quick and trouble-free manner.

The first of these new products is a compact Arduino-compatible shield named ADAM (Arduino Display Adaptor Module), which has been developed specifically to interface with the 4DLCD-FT843—permitting communication between it and the Arduino via the SPI interface. With dimensions of just 47.5mm x 53.4mm, the shield is suitable for use with Arduino Uno, Due, Duemilanove, Leonardo, Mega 1280/2560 and Pro 5V, as well as variety of popular Arduino clones. It has a micro-SD

card that provides the Arduino-based display system with capacious data storage. Through this the 4DLCD-FT843 can retrieve objects (such as images, sounds, fonts, etc.). Drawing power from the Arduino's 5V bus, ADAM regulates the 4DLCD-FT843's supply to 3.3 V. The FT800 EVE controller can deal with many of the graphics functions that would otherwise need to be undertaken by the Arduino.

ADAM is complemented by the 4DLCD-FT843-Breakout. With a footprint of 26.5mm x 12mm, this is a simple breakout module that allows the 4DLCD-FT843 to be attached to a general host or breadboard for prototyping purposes. It features a 10-way FPC connection for attachment with the 4DLCD-FT843, along with a 10-way, 2.54mm pitch male pin header for connection directly to the host board. An operational temperature range of -10°C to +70°C is supported by both these new products.

The EVE-driven 4DLCD-FT843, which was released last month, has a 4.3" TFT QWVGA display with a 4-wire resistive touch screen. It features a 64 voice polyphonic sound synthesizer, a mono PWM audio output, a programmable interrupt controller, a PWM dimming controller for the display's backlight, plus a convenient flexible ribbon connector.



www.4dsustems.com.au www.ftdichip.com (130521-IV)

Speedy 1200+ Acoustic Modem

Telecom authorities under pressure

By **Gerd Kowalewski**
(Germany)

Some of you may have heard of modems to enable a PC to “go online” to access BBSs in the dim past, but always with cable connections to the telephone network. Here we reminisce on the precursor of the wired modem: the acoustic coupler. Now a thing of the past, it blew the dust out of some telecom authority offices in the early 1980s.



First, let's do the ACK Handshake so anyone under 40 is not lost straight away: BBS stands for bulletin board system; modem is an acronym for modulator/demodulator, and a PC in our realms is a personal computer.

Acoustic couplers in the early 1980s were used for data communications via POTS (plain old telephone service — no kidding, *Ed.*) telephone handsets, and even on the German “C-Netz” carphone network (“Autotelefon”) to reach a remote line modem hooked up somewhere else on the public switched dial-up telephone network (PSTN). In the US such a device was called an acoustic coupler, or MUFF, where the microphone and earpiece parts of a telephone handset got pushed into a pair of cushioned seatings acting as acoustic seals to prevent external noise from interfering with the data carried over the voice channel once communications with the remote system had been established.

Historically, technically

Such a device technically was in fact a simple FSK (frequency shift keying) modulator and demodulator connected via an asynchronous serial connection (US: RS232C; EC: CCITT V.24/V.28) to one of these “newfangled” desktop computers, frequently built by the modem user or “hacker” himself.

Surely unforgettable: that scene from John Badham's 1983 movie “War Games”, with the text characters moving at a snail's pace (300 bps) across the green CRT screen, when a teenager gamer was trying to play the fascinating new computer game called Thermo Nuclear War against a US military super computer's brain, after hacking a secret line modem access port. Using two voice channel compatible pairs of carrier frequencies in an FSK modulation scheme, according to US Stds. Bell 103 for a whopping 300 bps FDX (full duplex), or Bell 202 for “high-speed” 1200 bps HDX (half duplex) or comparable European CCITT Stds. V.21 FDX or V.23 HDX, such devices established one major, or even two, in-band data communication channels within the telephone network's extremely variable (!) and noise infested narrow voiceband channel of about 300 Hz to 3.4 kHz (realistically, 2.5 kHz), depending on telephone lines used on the individual connection. FDX implies bidirectional communication at the same time; HDX means one-way communications at a time, with the direction of transmission changing under the control of a data comms protocol running on the host computers (anyone remember Kermit, Xmodem...?).

There was even a special mixed mode within the CCITT V.23 standard, the so-called split-mode,

ESTD 2004

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

with a “fast, high-throughput” RX channel at 1200 bps and a slow 75 bps TX channel. This technique was used for Germany’s Datex-P based “Bildschirmtext” (screen text) system of Deutsche Bundespost, or “Minitel” as I think it was called in France, to mention some of the first public data networks.

This was the intended market for EXAR’s world famous XR2206 (October 1976!) and XR2211 chips, plus at least a dozen others, like AMD’s AM7910 WorldChip, and Silicon System’s 73K22x series. EXAR/TI, Rockwell, AT&T and US Robotics soon became the major players in specialized DSP solutions for “data-pump” modems.

Clever use of CCITT V.22

The Speedy 1200+ acoustic coupler modem was one of a few types to have been tested successfully over the German C-Netz, the third shoot of Germany’s early mobile radio phone networks, which was used by lots of business and high up sales people on their travels and tours across the country. More on the early A-, B- and C-Netz networks in a future *Retronics* article.

This Speedy 1200+ for the first time used a DPSK modulation scheme for a 1200-bps FDX link according to CCITT V.22, making better use of the changing communication voice channel bandwidth. Uniquely at the time, the unit employed an inductive coupling method (selectable by a slide switch) instead of the traditional microphone to pick up the signal created by the magnetic stray field of the dynamic transducer in the earpiece. This avoids acoustic noise pick up on the low-signal level side of the device. This selection switch was needed for added compatibility with piezoelectric speakers used in earpieces, which do not create any magnetic stray field.

The system also supported the former 300 bps FDX CCITT V.21 standard, switch selectable for Originate or Answer mode.

In terms of hardware, on opening the Speedy 1200+ we find a Silicon Systems Inc. (later, TDK Semi. Corp.) 73K222 DSP, an 80C39 microcontroller, an 8-KB EPROM, a few V.28 interface drivers, opamps, LEDs and switches.

Many countries, many standards

I received the Speedy 1200+ pictured here from my brother who was clearing out his stuff, and found it still in the box! The unit was kind of dear to me because around 1986 I worked for a few years as a design engineer for its manufacturer,

CPV mbH in Hamburg.

Those were “wild west” times in Germany, with a lot of turmoil due to the changes afoot in the whole area of telecommunications. The German telecom authority, Deutsche Bundespost (DBP), was forced by European regulations to open up its former national network structures and to relax some of the very strict technical regulations, in order to allow nationwide competition.

Those exciting days saw strong confrontation between the industry and the Ministry for Post and Telecommunications, on how to break up the crusted structures of the former monopolist DBP. Especially the grueling lab tests done at the ZZF approval labs (Zentralamt für Zulassungen im Fernmeldewesen), located at Saarbrücken, southern Germany, scored highest in prevent-



Inventor

Robert Weitbrecht (1920-1983), deaf born, physicist and amateur radio enthusiast, made his first long distance call using his invention called "acoustic coupler device", hooked up to a TTY (teletypewriter) in May 1964. His so-called "deaf TTY" (telephone TTY) pioneered telecommunications between deaf people around the globe. Light and portable electronic typewriter machines combined with acoustic couplers soon replaced the earlier electromechanical TTY monsters (like the Silent 700 from TI [1]). This idea was picked up and worked out by the "computer" generation to follow.

ing foreign companies to get type approval for their communication equipment to be marketed in Germany. This was also the time when a ZZF approved 300-bps FDX/1200-bps HDX dial-up modem operating to CCITT V.21 and V.23 carried an immense price tag of DM 3,000. Far Eastern and US manufacturers just capitulated in front of the price quote and red tape for type approval in ole' Germany.

Even in later years, when laptop computers appeared, US standards seemed to dominate in the area of dial-up modems, and just a handful carried ZZF approval, i.e. showing the FTZ label with an old post hornet pictogram and the device's approval number.

Web Link

[1] Deafness.about.com/od/peopleindeafhistory/a/weitbrecht.htm

Further Reading / Browsing

A small gallery of Acoustic Couplers may be found at:

[Commons.wikimedia.org/wiki/Category:Acoustic_couplers?uselang=en](https://commons.wikimedia.org/wiki/Category:Acoustic_couplers?uselang=en)

Including:

Dataphone S21 (Woerltronic, Cadolzburg)

CX-21 (Epson)

AK2000 (EDV-Kontor GmbH)

Datenklo (Chaos Computer Club, 1985)

AM211 (Anderson-Jacobson)

"307" (TRANSDATA)

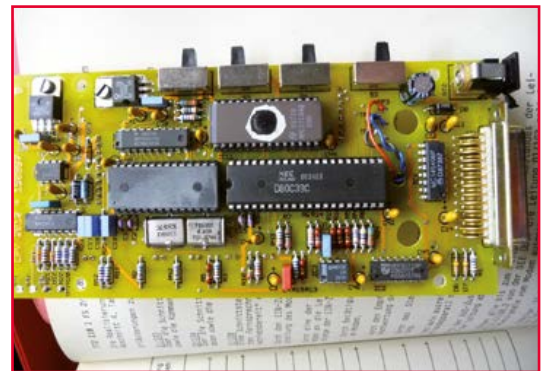
Silent 700 (Texas Instruments— from RCS/RI collection)

Also found there:

Telecoupler II (Road Warrior Intl.) (claims 33.4 kbps)

Carterphone (1959) — historic acoustic coupling device

Exactly at that time, the dwarf company CPV mbH jumped the bandwagon. CPV designed lots of dial-up modems with full German approval for many of the internationally well-known laptop manufacturers, as well as for a few German ones. It should be mentioned here that in those years there was no such thing as *European telecommunications*, it was all *EC* telecoms (European Community). Every European country including Switzerland (not part of the EC) had its own national equipment registration procedure and associated technical approval systems in place. No wonder that this European diversity was — and



still is — driving mass volume producers of data communication equipment ultimately crazy! Meanwhile this situation has changed somewhat. The EC-wide and national standards are much relaxed. And a dial-up modem is about €15 if you ever want to use one again when your broadband Mbits/s or Gbit/s lines are down.

Follow-up versions

For this particular apparatus, the German PTT was forced to create a whole new device class for approval—because nobody ever thought of having V.22 on an acoustic coupler! No worries, it's got its label with the post horn pictogram, allowing it to be used within the German public dial-up telephone network. A later, upgraded version, including the first optical line interface with German approval, previously invented for the space limited slots found in laptop computers, went on market. Another version, this time with an FCC68 approved dial-up line interface, data compression and MNP4 error correction, even made it to COMDEX electronics show—little wonder, it turned out a total flop in the US market.

(130465)

engineering electronics
 DISCUSS **embedded**
 design tips tutorial software
 engage **engineering** tools
 contests system **audio** business
 data **networking** **media**
 COMMUNITY **mobile**
social media talk
 information product news
 projects

Want to talk to us directly?
 Share your interests and opinions!
 Check out our New Social Media
 Outlets for direct engagement!



CIRCUIT CELLAR / AUDIOXPRESS / ELEKTOR

Hexadoku The Original Elektorized Sudoku

Spring weather allowing you should have had opportunities, however short, to take your Hexadoku puzzle outside for an attempt at solving. And here's another one to keep you going. Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of five Elektor book vouchers.

The Hexadoku puzzle employs numbers in the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the

thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

Correct entries received enter a prize draw. All you need to do is send us **the numbers in the gray boxes**.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for five Elektor Book Vouchers worth **\$70.00 (£40.00 / €50.00)** each, which should encourage all Elektor readers to participate.

Participate!

Before June 1, 2014, supply your name, street address and the solution (the numbers in the gray boxes) by email to: **hexadoku@elektor.com**

Prize winners

The solution of the March 2014 Hexadoku is: **0E4D6**.

The €50 / £40 / \$70 book vouchers have been awarded to: Adrie van de Ven (Netherlands), Ola Sandin (Sweden), Jose Carlos Negro (Spain), Saudin Dizdarevic (Bosnia and Herzegovina) en Kiss Tibor (Hungary).

Congratulations everyone!

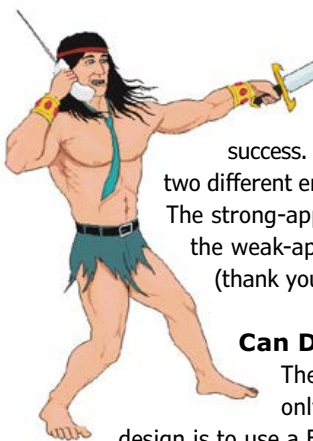
5	7			2	4			F	8				0	E
E	B			C					3				4	8
	2	6											A	9
	F		8	0			5	9			E	B		2
B			F	4		0	C		D		6			1
	C	D			6			0					7	A
6		1		F	A			E	4		0			3
		E	4	7	2			6	A	D	5			
		C	5	A		3			8	7	9	6		
1		0		4		C			5	2		D		7
	6	9			5			B				F	1	
3			A	1		F	D		0		2			5
	E		D	1			7	4			0	3		5
	5	8							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

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

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

Appearing Strong

By Gerard Fonte (USA)



In today's competitive environment the difference between appearing strong and appearing weak can have a significant impact on your success. Let's take a look at several scenarios with two different engineers that have identical capabilities. The strong-appearing engineer is John Bigbooty and the weak-appearing engineer is John Smallberries (thank you Buckaroo Banzai).

Can Do!

The boss has decided (correctly) that the only realistic approach to the new widget design is to use a Field Programmable Gate Array (FPGA). However, neither engineer has any real experience in working with FPGAs. (Note: the cost for the engineer's salary, benefits, support and overhead is about \$2500/week.)

John Bigbooty: "No problem! I'll get on it right away. You can count on me." He goes to his desk and starts researching FPGAs. There's loads of data on the Internet as well as many vendors. He figures that the first thing to do is decide which FPGA type is best. He spends several days working through each vendor's products and eventually chooses the Xilinx Spartan family. He downloads and studies the data-sheets, user guides and application notes. There are several hundred pages of data that takes a few more days to examine. He downloads the development software and takes a few more days to get familiar with it. At the end of two weeks, he is ready to start developing the FPGA for the widget. Cost to company: \$5000 and a two week delay.

John Smallberries: "Gee boss, I've never done anything like that before. Let me check it out." He goes to Pinky Carruthers (*ibid*) who has a lot of experience with them and asks for help. Pinky is involved in another very important project but can spend half an hour explaining FPGAs. John comes away with a basic understanding of what is needed, which FPGA is best suited for the application, and what he needs to learn. He spends an hour on-line researching Xilinx technical support. About two hours after being given the assignment, he goes to his boss, explains the decisions he has made and requests \$1500 for two on-line courses to bring him up to speed. His boss agrees and in two days he is ready to start developing the FPGA for the widget. Cost to company \$2500 and a two day delay.

Say What?

At the weekly meeting a new assignment is presented. A low-speed serial interface has to be added to the widget as per the customer request. The stated reason is to interface to some older customer equipment.

John Bigbooty: It seems certain that either USB or Firewire are good choices. However, USB is more universal and speed isn't an issue. He decides on USB and spends the next week designing a simple and inexpensive, drop-in module for the widget. It is truly an elegant and sophisticated design that he is anxious to present at the next project meeting.

John Smallberries: He goes to his desk and starts working and realizes that he doesn't really know what the customer wants. His boss wasn't specific at the meeting. So he drops by his boss's office to ask him to clarify the assignment. His boss replies that the customer wants an RS-232 interface. They want to transfer some calibration data during start-up and want to record occasional operating parameters. He returns to his desk and provides a workable RS-232 interface at the next project meeting.

Oops!

The widget project is ready to go into production and the engineer realizes that he made a slight specification error in a power supply capacitor. He overlooked that AC voltage is specified as RMS rather than peak (which is 141% more than expected). So the 50% safety margin in capacitor's working voltage is really only 9%.

John Bigbooty: Admitting to an error will make look bad and I'll get a poor performance review. The capacitor is still working within its safety margins. So, it's not really an error. A different engineer may very well specify a 10% safety margin. After all, the lower working voltage means a cheaper capacitor. Anyway, the design is acceptable as it is. Besides, it will take months or years--perhaps never--for anyone to realize that the capacitor is under-rated. Who knows what will happen in that time.

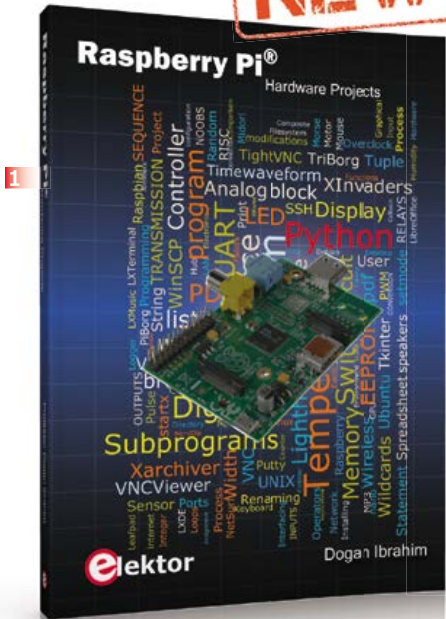
John Smallberries: As soon as he realizes his error he goes to his boss. "Boss, I screwed up. I called out the wrong working voltage for capacitor C6. There's only a 9% safety margin instead of 50%. The capacitor is a critical part of the power supply so reliability is an important consideration. I checked out available parts and we can get a capacitor with 50% margin that will fit the circuit board for only fifteen cents more. We can return our existing stock of old capacitors and replace them with new ones within two days. If we market an unreliable product it will cost us in sales, product returns and customer loyalty."

Role Reversal

It is probably not surprising to realize that employers much prefer the attitude and performance of John Smallberries to John Bigbooty. In this case, being strong and independent is really being selfish and shortsighted. John Smallberries demonstrates better problem solving skills, better interpersonal relationships and a better understanding of business. He also shows loyalty to the company and its customers. In short, being weak is really being strong in character.

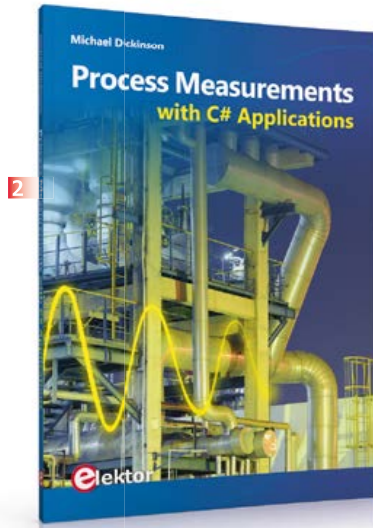
(140008)

NEW



1

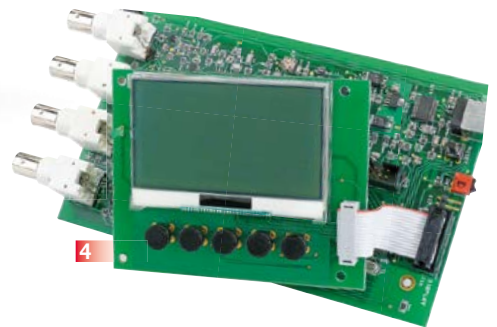
15% DISCOUNT
for GREEN and GOLD Members!
www.elektor.com/may



2



3



4

The RPi in Control Applications

1 Raspberry Pi Hardware Projects

This book starts with an introduction to the Raspberry Pi computer and covers the topics of purchasing all the necessary equipment and installing/using the Linux operating system in command mode. Use of the user-friendly graphical desktop operating environment is explained using example applications. The RPi network interface is explained in simple steps and demonstrates how the computer can be accessed remotely from a desktop or a laptop computer. The remaining parts of the book cover the Python programming language, hardware development tools, hardware interface details, and RPi based hardware projects.

ISBN 978-1-907920-29-5
£34.95 • € 39.95 • US \$54.00

An essential source of reference material

2 Process Measurements with C# Applications

Measurement is vital to the successful control of any process. This book introduces PC based measurement systems and software tools for those needing to understand the underlying principles or apply such techniques. Throughout the book, the C# programming language

is used to give the reader immediate practical desktop involvement. C-Sharp has a wide support base and is a popular choice for engineering solutions. The basics of measurement and data capture systems are presented, followed by examples of software post-processing. Application examples are provided from a range of process industries, with reference to remote monitoring, distributed systems and current industrial practices.

144 pages • ISBN 978-1-907920-24-0
£23.95 • € 27.50 • US \$38.00

Lots of power with low distortion

3 Q-Watt Audio Power Amplifier

Elektor has a long history with audio power amplifiers. We are proud to present yet another fully analog circuit developed entirely in house. Despite the simple design of this audio power amp with just one pair of transistors in the output stage, Q-Watt can deliver over 200 quality (Q!) watts into 4 ohms with exceptionally low distortion thanks to the use of a special audio driver IC. Due to popular demand, this project is now supported by a semi-kit containing everything you need to build a stereo amplifier.

2 PCBs and all electronic parts
Art.#110656-72
£147.95 • € 169.95 • US \$230.00

The luxury of precision within everyone's reach

4 500 ppm LCR Meter

The remarkable precision of this device and its amazing ease of use are the result of careful design. It works so well behind its uncluttered front panel that one could almost forget the subtleties of the measurement techniques employed. A dream opportunity for our readers who are passionate about measurement to enjoy themselves. If, like us, you wonder at the marvels modern techniques bring within our reach, come along and feel the tiny fraction of a volt.

Set: main board and LCD board,
assembled and tested
Art.# 110758-93
See www.elektor.com/lcrometer

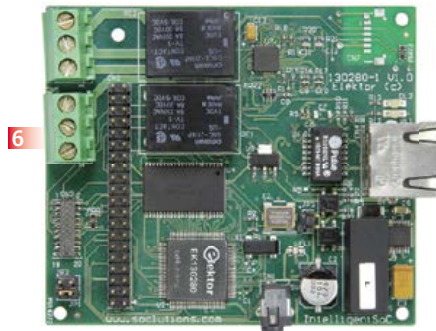
All articles in Elektor Volume 2013

5 DVD Elektor 2013

This DVD-ROM contains all editorial articles published in Volume 2013 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 favorite



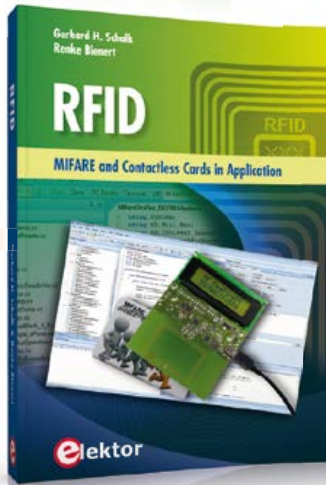
5



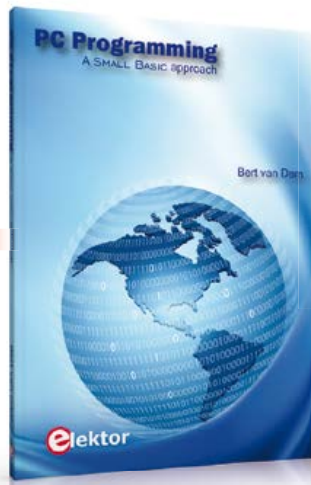
6



7



8



9

graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-277-8
£23.95 • € 27.50 • US \$38.00

The First Elektor Chip

6 E-Lock

The E-Lock chip allows you to connect your control system to the 'Network of Networks' and monitor and control it from anywhere on or around the globe using your computer, tablet or smartphone without having to worry about the security of the connection and in full assurance of protection against intruders.

Ready build module
Art.# 130280-91
£96.95 • € 111.00 • US \$150.00
 See www.elektor.com/e-lock

Programming step-by-step

7 Android Apps

This book is an introduction to programming apps for Android devices. The operation of the Android system is explained in a step by step way, aiming to show how personal applications can be pro-

grammed. A wide variety of applications is presented based on a solid number of hands-on examples, covering anything from simple math programs, reading sensors and GPS data, right up to programming for advanced Internet applications. Besides writing applications in the Java programming language, this book also explains how apps can be programmed using Javascript or PHP scripts. When it comes to personalizing your smartphone you should not feel limited to off the shelf applications because creating your own apps and programming Android devices is easier than you think!

244 pages • ISBN 978-1-907920-15-8
£34.95 • € 39.95 • US \$54.00

MIFARE and Contactless Cards in Application

8 RFID

MIFARE is the most widely used RFID technology, and this book provides a practical and comprehensive introduction to it. Among other things, the initial chapters cover physical fundamentals, relevant standards, RFID antenna design, security considerations and cryptography. The complete design of a reader's hardware and software is described in detail. The reader's firmware and the associated PC software support programming using any .NET language. The specially developed PC program, "Smart

Card Magic.NET", is a simple development environment that supports sending commands to a card at the click of a mouse, as well as the ability to create C# scripts. Alternatively, one may follow all of the examples using Visual Studio 2010 Express Edition. Finally, the major smart card reader API standards are introduced. The focus is on programming contactless smartcards using standard PC/SC readers using C/C++, Java and C#.

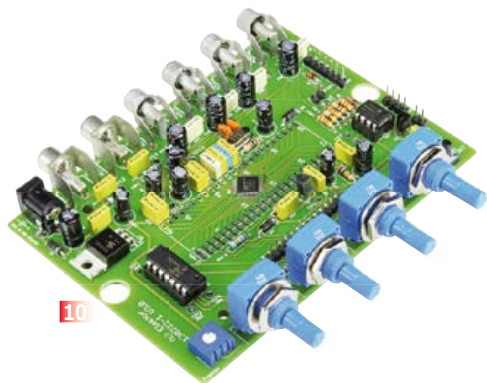
484 pages • ISBN 978-1-907920-14-1
£43.95 • € 49.90 • US \$68.00

A Small Basic approach

9 PC Programming

There are many different PC programming languages available on the market. They all assume that you have, or want to have, a knack for technology and difficult to read commands. In this book we take a practical approach to programming. We assume that you simply want to write a PC program, and write it quickly. Not in a professional environment, not in order to start a new career, but for plain and simple fun... or just to get a task done. Therefore we use Small Basic. You will have an application up and running in a matter of minutes. You will understand exactly how it works and be able to write text programs, graphical user interfaces, and advanced drivers.

194 pages • ISBN 978-1-1-907920-26-4
£30.95 • € 34.50 • US \$47.00

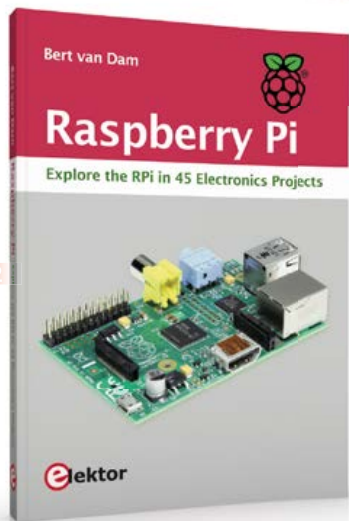


10

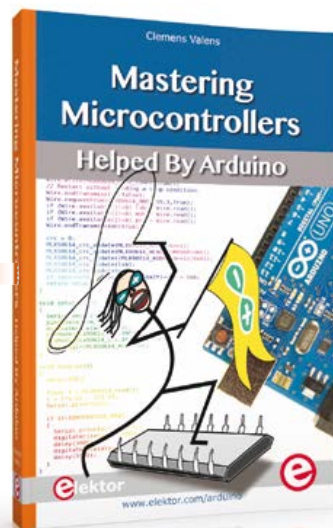
NEW



11



12



13

Drag 'n Drop

10 ADAU1701 Universal Audio DSP Board

Always wanted to start out with DSPs, but afraid of the SMD? Here's the answer: Elektor's all-DIY Universal Audio DSP Board! Based on the Analog Devices ADAU1701 DSP chip and drag-and-drop software, this board will ease your way to becoming a sound craftsman, guaranteed maths-free.

Semi-kit:

all TH components, PCB with DSP preassembled

Art.# 130232-71

£63.95 • € 72.95 • US \$99.00

110 Elektor Editions, Over 2500 Articles

11 DVD Elektor 2000 through 2009

This DVD-ROM contains all circuits and projects published in Elektor magazine's year volumes 2000 through 2009. The 2500+ articles are ordered chronologically by release date (month/year), and arranged in alphabetical order. A global index allows you to search specific content across the whole DVD. Every article is printable using a simple print function. This DVD is packed with ideas, circuits and projects that

are ideal for any electronics enthusiast, student or professional, regardless of whether they are at home or elsewhere.

ISBN 978-1-907920-28-8

£77.95 • € 89.00 • US \$121.00

Explore the RPi in 45 Electronics Projects

12 Raspberry Pi

This book addresses one of the strongest aspects of the Raspberry Pi: the ability to combine hands-on electronics and programming. No fewer than 45 exciting and compelling projects are discussed and elaborated in detail. From a flashing lights to driving an electromotor; from processing and generating analog signals to a lux meter and a temperature control. We also move to more complex projects like a motor speed controller, a webserver with CGI, client-server applications and Xwindows programs. Each project has details of the way it got designed that way. The process of reading, building and programming not only provides insight into the Raspberry Pi, Python, and the electronic parts used, but also enables you to modify or extend the projects any way you like.

288 pages • ISBN 978-1-907920-27-1

£34.95 • € 39.95 • US \$56.40

Helped By Arduino

13 Mastering Microcontrollers

The aim of this book is not only to let you enter the World of Arduino, but also to help you emerge victorious and continue your microcontroller programming learning experience by yourself. In this book theory is put into practice on an Arduino board using the Arduino programming environment.

Having completed this fun and playful course, you will be able to program any microcontroller, tackling and mastering I/O, memory, interrupts, communication (serial, I²C, SPI, 1-wire, SMBus), A/D converter, and more. This book will be your first book about microcontrollers with a happy ending!

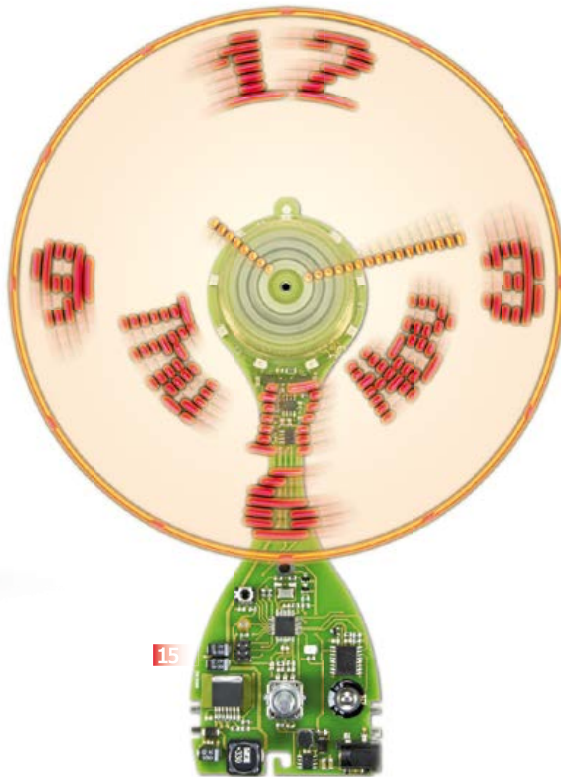
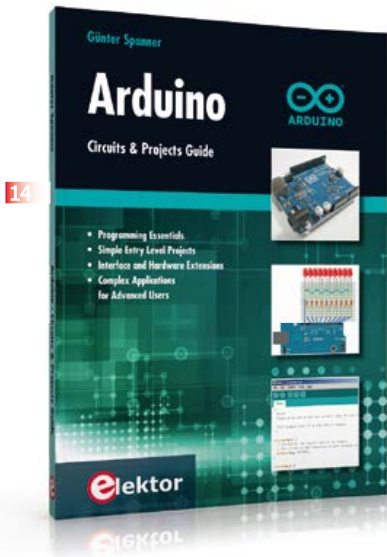
348 pages • ISBN 978-1-907920-23-3

£34.95 • € 39.95 • US \$54.00

Circuits & Projects Guide

14 Arduino

The Arduino user is supported by an array of software libraries. In many cases, detailed descriptions are missing, and poorly described projects tend to confuse rather than elucidate. This book represents a different approach. All projects are presented in a systematical



manner, guiding into various theme areas. In the coverage of must-know theory great attention is given to practical directions users can absorb, including essential programming techniques like A/D conversion, timers and interrupts—all contained in the hands-on projects. In this way readers of the book create running lights, a wakeup light, fully functional voltmeters, precision digital thermometers, clocks of many varieties, reaction speed meters, or mouse controlled robotic arms. While actively working on these projects the reader gets to truly comprehend and master the basics of the underlying controller technology.

260 pages • ISBN 978-1-907920-25-7
£34.95 • € 39.95 • US \$54.00

Time & Date Floating in the Air

15 UltiProp Clock

Electronics is never so fine as when it skillfully combines magic with physics, mechanics with software, imagination with thoroughness and precision, and a taste for beauty with good workmanship. This timepiece was designed to display the time and date in an original way—but we admit we did also design it to

draw cries of amazement from the visitors who find it in our laboratory. We'll bet many of you will want to do the same in your own homes!

Module, ready assembled and tested
Art.# 120732-91
 See www.elektor.com/ultiprop

Ideal reading for students and engineers

16 Practical Digital Signal Processing using Microcontrollers

This book on Digital Signal Processing (DSP) reflects the growing importance of discrete time signals and their use in everyday microcontroller based systems.

The author presents the basic theory of DSP with minimum mathematical treatment and teaches the reader how to design and implement DSP algorithms using popular PIC microcontrollers. The author's approach is practical and the book is backed with many worked examples and tested and working microcontroller programs. The book should be ideal reading for students at all levels and for the practicing engineers who may want to design and develop intelligent DSP based systems. Undergraduate students should find the theory and the practical projects invaluable during their final year projects. Similarly, postgraduate students should be able to develop advanced DSP based projects with the aid of the book.

428 pages • ISBN 978-1-907920-21-9
£43.95 • € 49.90 • US \$68.00

Further Information and Ordering: www.elektor.com/store or contact customer service for your region

UK / ROW

Elektor International Media
 78 York Street
 London - W1H 1DP United Kingdom
 Phone: +44 20 7692 8344
 E-mail: service@elektor.com

USA / CANADA

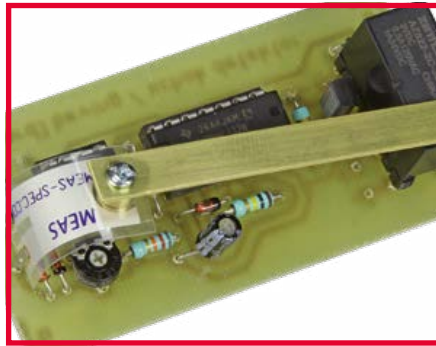
Elektor US
 111 Founders Plaza, Suite 300
 East Hartford, CT 06108 USA
 Phone: 860.289.0800
 E-mail: service@elektor.com

NEXT MONTH IN ELEKTOR MAGAZINE



FT311D Breakout Board

If you're into driving electronics circuits it may be quite useful to employ an Android device to generate control commands for a circuit, or to display data on a screen. To help you on, Elektor Labs have designed a small circuit that can be connected directly to an Android device. The circuit is built around an FT311D from FTDI, which is a so-called USB Android Host IC.



Earthquake Detector

This circuit gives a visual and acoustic warning when it detects an earthquake or a major shock. It is plain simple and consists of a special piezoelectric sensor, a pulse stretcher, an alerting device and a relay. It is also possible to send the alert signal wirelessly to a remote location. A simple transmitter and receiver got designed for the purpose, using low power radio modules. These are also described in the June 2014 issue.



Revolution Counter with OLED Display

Users of milling machines and lathes often need to know the exact speed at which the cutter or the workpiece is revolving. For this is a small shield got designed for putting on an Arduino Micro, creating a nice compact unit. An OLED is used for display function. The speed detection is implemented using an LED-phototransistor device. The display can show both the speed and the total up time.

Article titles and magazine contents subject to change, please check www.elektor-magazine.com for updates.

Elektor June 2014 is processed for mailing to US, UK and ROW Members starting May 20, 2014.

See what's brewing
@ Elektor Labs 24/7

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

A screenshot of the Elektor Labs website. The header features the 'elektor labs' logo and the tagline 'Sharing Electronics Projects'. Below the header is a navigation bar with 'Home', 'Proposals', 'In Progress', and 'Finished' tabs. The main content area is divided into three columns: 'Proposals', 'In Progress', and 'Finished'. Each column shows project thumbnails with titles, view counts, and star ratings. A 'DEAD END' badge is visible on the 'The Cat fountain' project. A 'Q&A on Elektor projects' section is also present. On the right side, there is a 'Create a Project' section with a 'Create a new project or enter a proposal' button, and a 'Not a member?' section with a 'You can! Click here' button.

ORDERING INFORMATION

To order, contact customer service for your region:

USA / CANADA

Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
E-mail: service@elektor.com

Customer service hours:
Monday-Friday 8:30 AM-4:30 PM EST.

UK / ROW

Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344
E-mail: service@elektor.com

Customer service hours:
Monday-Thursday 9:00 AM-5:00 PM CET.

PLEASE NOTE: While we strive to provide the best possible information in this issue, pricing and availability are subject to change without notice. To find out about current pricing and stock, please call or email customer service for your region.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in the magazine. If difficulties in obtaining components are suspected, a source will normally be identified in the article. Please note, however, that the source(s) given is (are) not exclusive.

TERMS OF BUSINESS

Shipping Note:

All orders will be shipped from Europe. Please allow 2-4 weeks for delivery.

Returns

Damaged or miss-shipped goods may be returned for replacement or refund. All returns must have an RA #. Call or email customer service to receive an RA# before returning the merchandise and be sure to put the RA# on the outside of the package. Please save shipping materials for possible carrier inspection. Requests for RA# must be received 30 days from invoice.

Patents

Patent protection may exist with respect to circuits, devices, components, and items described in our books, magazines, online publications and presentations. Elektor accepts no responsibility or liability for failing to identify such patent or other protection.

Copyright

All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, discs, and software carriers published in our books and magazines (other than in third-party advertisements) are copyrighted and may not be reproduced (or stored in any sort of retrieval system) without written permission from Elektor. Notwithstanding, printed circuit boards may be produced for private and educational use without prior permission.

Limitation of liability

Elektor shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connection with, the supply of goods or services by Elektor other than to supply goods as described or, at the option of Elektor, to refund the purchaser any money paid with respect to the goods.

MEMBERSHIPS

Membership renewals and change of address should be sent to the Elektor Membership Department for your region:

USA / CANADA

Elektor USA
P.O. Box 462228
Escondido, CA 92046
Phone: 800-269-6301
E-mail: elektor@pcspublink.com

UK / ROW

Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344
E-mail: service@elektor.com



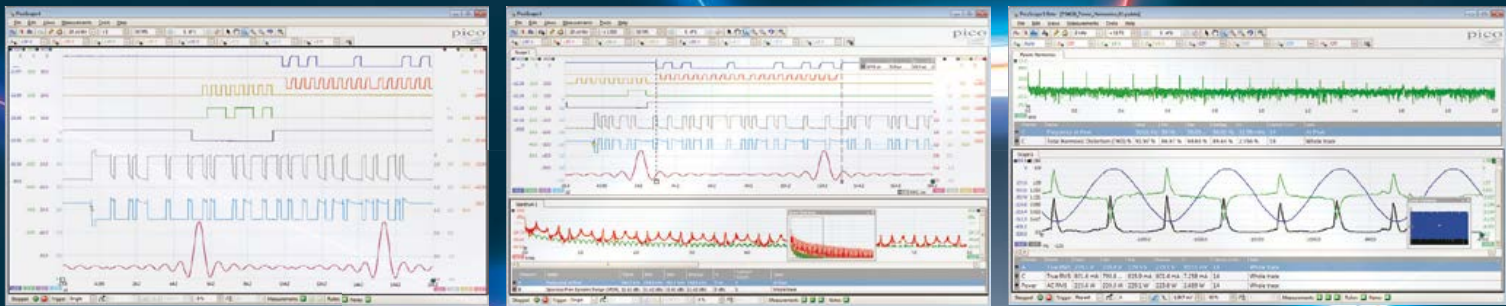
Do you want to become an Elektor GREEN or GOLD Member or does your current Membership expire soon?

Go to www.elektor.com/member.

7 functions of PicoScope:

1. Oscilloscope
2. Spectrum analyzer
3. Function generator
4. AWG
5. Logic analyzer
6. Serial protocol analyzer
7. Automatic waveform test

8 CHANNEL PC OSCILLOSCOPE



For just £1395

- High resolution
- USB powered
- Deep memory

INCLUDES AUTOMATIC MEASUREMENTS, SPECTRUM ANALYZER, SDK, ADVANCED TRIGGERS, COLOR PERSISTENCE, SERIAL DECODING (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASKS, MATH CHANNELS, ALL AS STANDARD, WITH FREE UPDATES

12 bit • 20 MHz • 80 MS/s • 256 MS buffer • 14 bit AWG

www.picotech.com/PS242