elektor

Pattern Lock

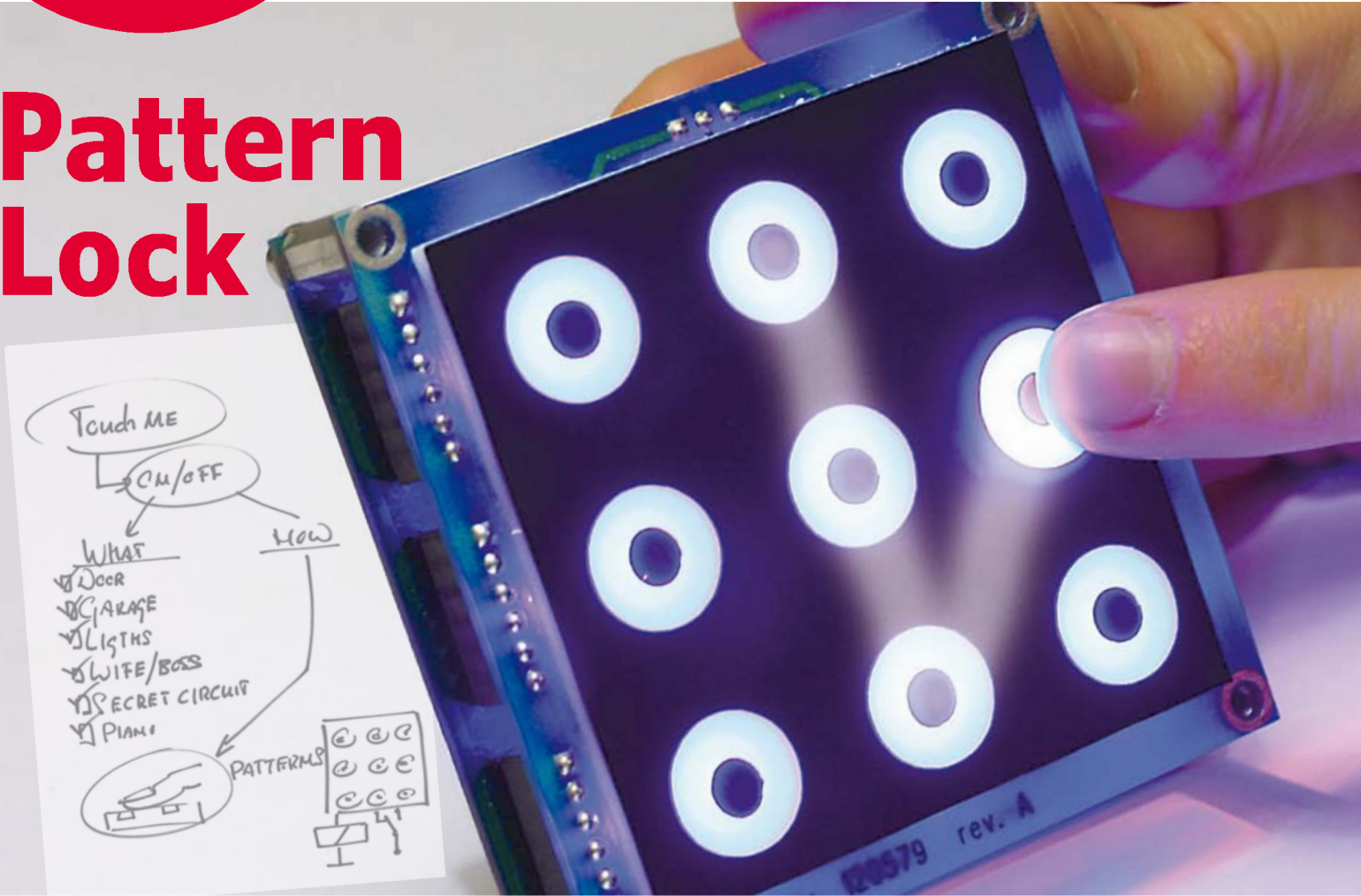Touch ME
ON/OFF
WHAT
HOW
Door
GARAGE
LIGTHS
WIFE/BOSS
SECRET CIRCUIT
PIANI
PATTERNS

Shape recognition for access control

● **Linux Board Extension** | Power LED Driver | Thermo Book |
Handlebar Heating | **Arduino'd Cookie Guard** | Gadgeteer | DIP'ed
Accelerometer ● **XTAL Killer** | LDO in THT ● **The Curse**
of the Collector ● Virtual Money ● **Elektor World**

Contents

RH readout can also be toggled manually (literally) by clapping your hands. The sensor used is a Sensirion SHT15.

● Industry

● Labs

● Magazine

●Community

From PRware to working project

Pattern Lock, this month's front cover project
and leading article, proves that technology
rolled out big time by leading manufacturers
can take a long time before it gets adopted by
design engineers to the extent of DIY pro-
jects for home assembly using low cost parts
and free software libraries. In this case, our
contributing author David Ardouin based his
design on Atmel's Qtouch suite of hardware
and software, but with an original touch to it.
I remember Qtouch being launched by Atmel
PR and sales staff (and a few hired young
ladies) at Embedded Systems Conference
(ESC) Boston in 2008. The launch was underscored with development kits, leaflets,
a pet dinosaur you could stroke, Norwegian giants, inflatable hammers, and a few
slick presentations on huge screens. Getting a dev kit wasn't easy though, I even-
tually received one from one of the very few engineers around on the booth, after
asking some basic questions about intrinsic capacitances, static buildup and EMC
issues — and getting really useful answers once the PR people were out of hearing
range. Later that day I powered up the Qtouch dev kit in my hotel room and was
impressed by the quick response and electrical robustness of capacitive sensing in
combination with microcontroller technology. Some of my fellow journalists prefer-
ring NASDAQ over technology were amazed to see the blue PCB actually working.
Three years on, I am grateful to David for his effort in bringing Qtouch to the
dynamic stage called Elektor magazine, and for covering the basics of capacitive
sensing, rather than presenting another black box circuit with an ATtiny88 in it
doing 'something' between I and O. His project and in-depth article successfully tell
the story of capacitive touch evolving from techno concept right up to hands-on —
from PRspeak right up to understanding and soldering. Hopefully our publication
triggers some thought that should go into replacing keypads for access systems
with capacitive sensors allowing a personal signature to be 'drawn' — it's much
cooler than banging in PIN codes like 1-2-3-4 or 6-5-0-2 to open the lab doors.
Enjoy reading this edition,

Jan Buiting, Managing Editor

The Team

Managing Editor:	Jan Buiting (editor@elektor.com)
International Editorial Staff:	Harry Baggen, Thijs Beckers, Eduardo Corral, Wisse Hettinga, Denis Meyer, Jens Nickel, Clemens Valens
Design staff:	Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser
Membership Manager:	Raoul Morreau
Graphic Design & Prepress:	Giel Dols, Mart Schroijen
Online Manager:	Daniëlle Mertens
Managing Director:	Don Akkermans

Edward T. Dell, Jr.: In Memoriam

February 12, 1923 – February 25, 2013
Edward Dell, founder and former publisher of Audio Amateur Inc., and former publisher of *Elektor USA* magazine (1990-92), died Monday, February 25, 2013, at the age of 90.

Dell, a legendary audio guru, developed his taste for publishing and audio as a teenager. He became a veteran builder of audio hi-fi speakers and was a long-time full member of the Audio Engineering Society and the Boston chapter of the Acoustical Society of America. He published magazines and books on all areas of audio for more than 35 years. After launching the company's flagship publication, *The Audio Amateur*, in 1970, Dell published a number of specialized titles covering a wealth of subjects in audio. Proud to lead a force of some 700 authors worldwide — experts and enthusiasts in audio technology — he started *audioXpress* 32 years later, in 2002. Dell propelled *audioXpress* to its position as the audio technology authority and, in 2011, he sold his company's assets — which included *audioXpress, Voice Coil*, and the annual directories *Loudspeaker Industry Sourcebook* and *World Tube Directory*, as well as several books and CDs — to the Elektor group, which publishes *Circuit Cellar* and the North American version of *Elektor*.

Dell corresponded with many long-time readers, authors and editors after his retirement, maintaining a connection to the audio world he loved. His literary legacy continues to inspire a new generation of audiophiles as they design and build their own "dream" systems.

Supporting Companies

Not a supporting company yet?

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

Elektor World

Compiled by
Wisse Hettinga

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

Good marks & certified too

Doing a training with Elektor pays off. Quite recently Elektor was awarded the CEDEO certification. More than 80% of participants of Elektor Workshops in The Netherlands rated the overall quality of the training and workshop sessions as 'good' to 'very good'. CEDEO is Holland's independent institute for rating and monitoring business training institutes. With this certification in the pocket Elektor is on the list of 800 officially recognized training institutes for The Netherlands. Good job!

Now eat this

When Cederique, a trainee at Elektor Labs, opened his lunch box, most of us were a bit surprised. Normally you'd be greeted by Dutch sliced bread with some cheese (and you drink milk with it) but now something completely different emerged. Wrapped in old paper this nice shiny tube hailed us from the olden days... our best guess is, it 's a Philips photo multiplier tube. The type number we reckon is PM 1910, oddly typed as "I9IO" on the primitive label. The first thought is always — what can we do with it? We tried looking up the specs but this piece of engineering is from the pre-google period — no results showed up. Anyone of you worked with this component? Let us know.
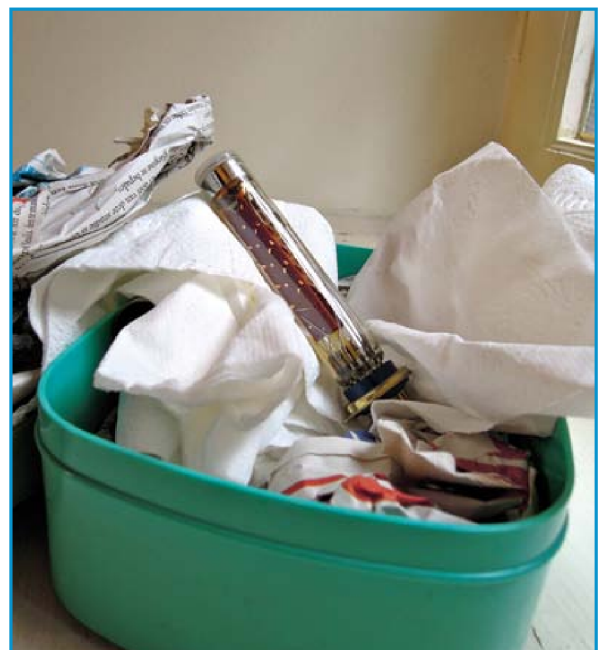
Starting out with Elektor

One of Elektor's' best kept secrets is its vast network of correspondents. They are all over the world and talk with professors, students, designers and technicians. Pictured here is Professor David Banjerdpongchai, Head of Electrical Engineering Department of the Chulalongkorn University in Bangkok, Thailand, with his first edition of *Elektor* magazine. Welcome to the network, David!

Inside the minds of our engineers

E. HARTFORD, CT, USA — The *Circuit Cellar* team recently surveyed a random sample of more than 1,000 engineers and academics on their technical interests and preferences. How often do you solder? How many milliamps have you felt? Do you know more than three programming languages? Do you use FPGAs? Which companies make the best embedded products? And more!

The results are fascinating! To check them out, go to

http://bit.ly/11Qzflp.

The electrical engineers, academics, and students who read *Circuit Cellar* hail from a wide range locations across the globe, such as the US, Brazil, India, The Netherlands, Germany, the UK, and Japan. Despite having different languages and cultures, the survey results prove they share a common dedication to, and passion for, electrical engineering.

Pattern Lock
Shape recognition for access control
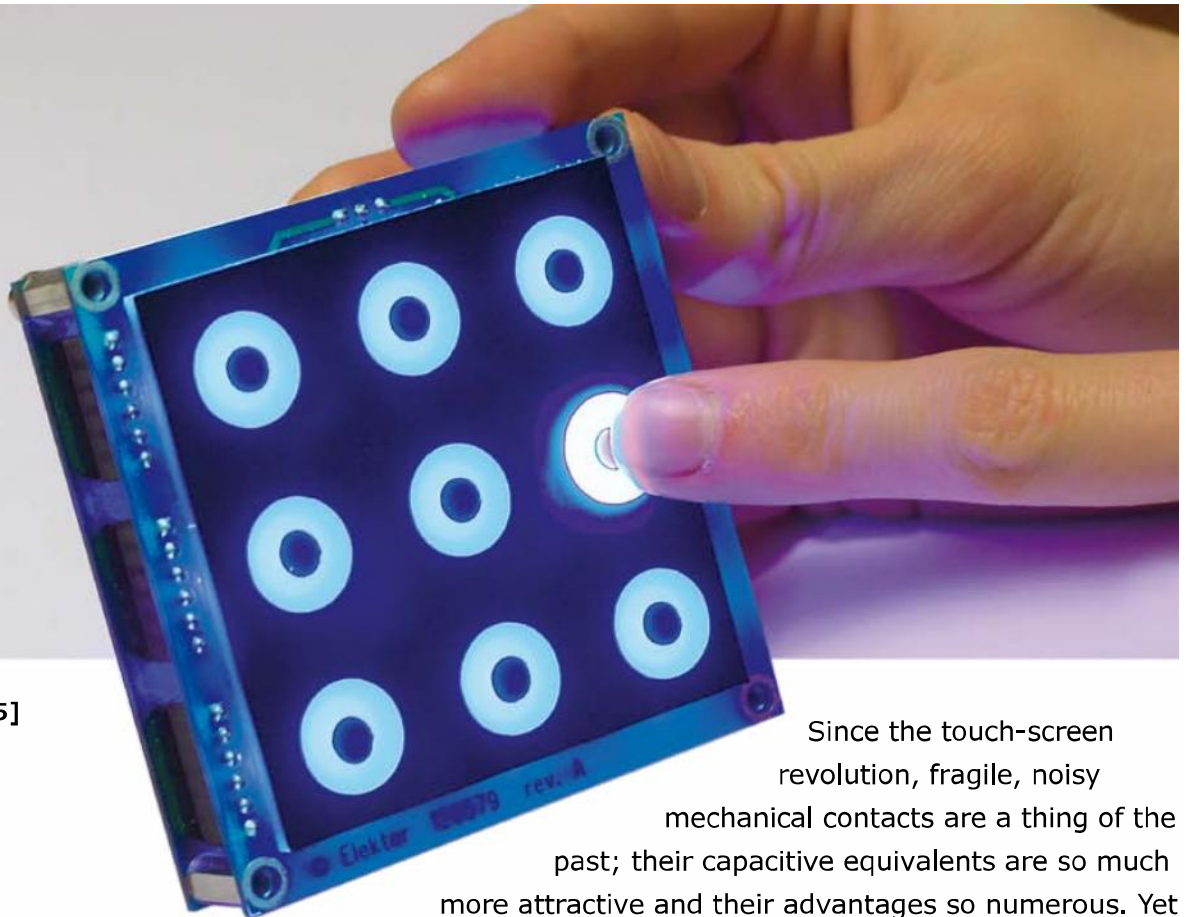
By **David Ardouin [5]**
(France)

Since the touch-screen revolution, fragile, noisy mechanical contacts are a thing of the past; their capacitive equivalents are so much more attractive and their advantages so numerous. Yet judging by the slim presence of this technology in the circuits published by Elektor, a lot of people still need convincing it's worthwhile. This is why I'm proposing here not only an entertaining approach, in order to demystify the subject and get a good grasp of it, but also, to encourage you in turn to use touch-screen interfaces, a serious, well-designed application, using Atmel's QTouch detection method.

After developing a touch-screen interface in a professional context, I wanted to share my experience in the form of a capacitive technology demo board. With the craze for smartphones using the Android OS with their famous pattern lock, I thought it would be interesting to modernize the well-known digicode system: instead of entering a code on a 9-key pad, you trace out a geometric pattern with your finger. My Pattern Lock is an elegant, back-lit touch-sensor lock capable of separately memorizing two patterns, each comprising four to nine sensors. Recognition of one of these codes makes it possible to activate any circuit you like via a relay.

Before getting down to the bones of the subject, let's take a moment to look at the principle of touch interfaces. Veteran readers will recall that

Elektor has published a whole host of capacitive contacts of all sorts, at a time when only analogue techniques were available. It was a fascinating adventure, but the results were often disappointing. Thanks to the use of microcontrollers, the effectiveness of today's techniques, like the one described here, far exceeds those earlier attempts.

Limits and advantages

A direct question: "What use is a capacitive sensor?" Depending on their function, touch interfaces are distinguished in particular according to their degree of freedom, a term mechanical engineers are very keen on. The simplest detectors are pushbuttons and switches, for which your freedom is reduced to almost nothing; these controls, referred to as "zero dimension", meaning they have only two states: active or inactive, closed or open, on or off, 0 or 1. These are capacitive sensors (**Figure 1a**), they operate a single command in an all-or-nothing manner.

The touch interface offers the possibility of a finer capacitive command, e.g. a linear command, proportional to the movement of a finger sliding across a surface from one extreme to the other. We speak of a slider, but this one-dimensional travel, which thus allows us to vary only a single variable, doesn't necessarily have to be in a straight line; it can be circular, as on one very famous mp3 player — and we speak then of a rotor (**Figure 1b**).

If the touch zone is an area, where the finger is free to move in all directions, as on a phone screen, it becomes possible to modulate two output variables at the same time (**Figure 1c**). This technique, although very close to the other two, is beyond the scope of this article.
The big advantage of capacitive keys or sliders
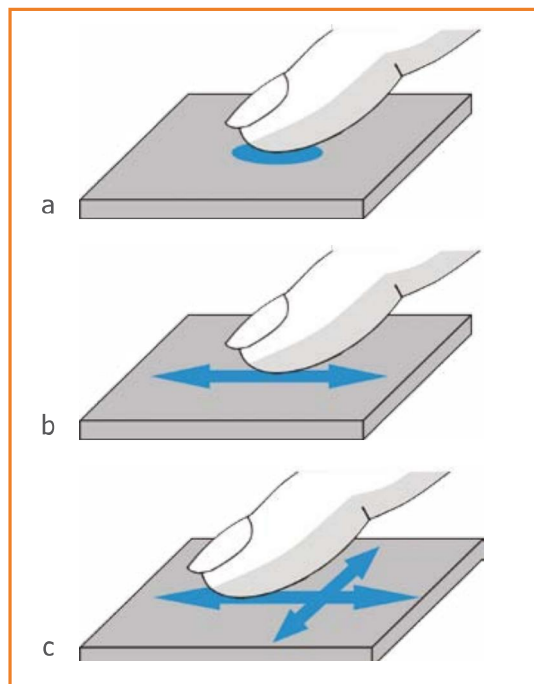
Figure 1.
A simple key, with 0 dimensions (a),
or a one-dimensional touch surface or slider (b), or a two-dimensional touch surface (c).

lies in the total freedom in laying out the command elements, labels, and pictograms used. Not to overlook the hi-tech aspect, often regarded as more attractive than old-fashioned round or square buttons. A touch sensor also has the advantage of being completely sealed; it doesn't wear out and is easy to illuminate from inside. If you are thinking about fitting touch keys to a project already controlled by an Atmel microcontroller, and you have a few pins and resources left, then including capacitive keys will only cost a few passive components and a little bit of space on the PCB.

Some reminders of physics

A capacitor consists of a pair of conductive electrodes separated by an insulating layer called the dielectric. The dimension of the electrodes, their spacing, and the dielectric material are adapted

The author

When I was little, I dreamt of becoming "a handyman just like Dad". This childhood dream never left me, thanks to the design and technology lessons at secondary school and a great many projects built from magazines. Then I turned to electronics: a High-School diploma in Sciences and then 5 years' higher education at the *ESEO* engineering college in Angers enabled me to make it my career without diminishing my enthusiasm. Between my roles as husband and father, I manage to find enough free time to perfect a few of the ideas teeming around inside my head, as well as to do a bit of model aircraft making.

Figure 2.
Single (a) and coupled (b)
electrodes.
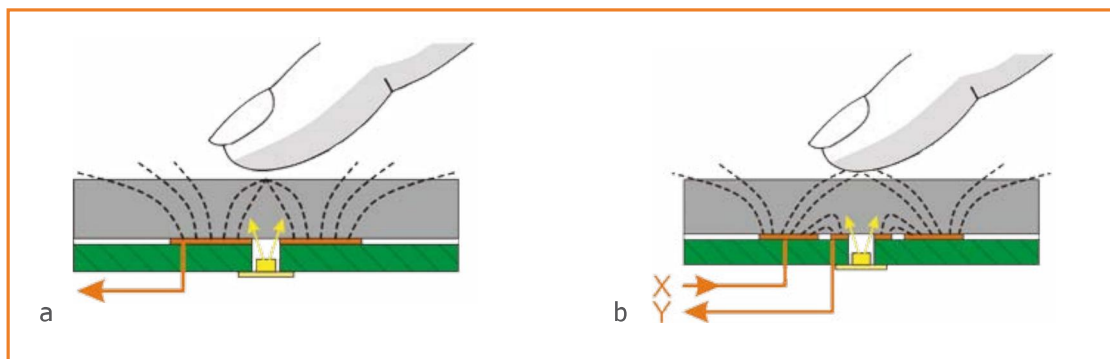
according to the degree of capacitance required. To ensure good sensitivity, we look for the greatest possible difference in capacitance between the absence and the presence of the finger. This variation is of the order of only a few tens of picofarads. So for our detectors to be reliable, we need to obey a few rules.

Figure 3.
How the touch circuit's coupled electrodes are arranged.

Figure 4.
Details of one electrode.

Twins

For capacitive detection, two types of electrodes are commonly used, each with its own advantages and drawbacks. The first is the simplest (**Figure 2a**); the key consists of a conductive pad of around 10 mm (⅜") in diameter, a resistor and a capacitor — and not forgetting the insulating material, to form the sensitive surface. This sort of functional block can be duplicated to create as many buttons as you wish, as long as the microcontroller handling the control has the required number of pins, i.e. $2n + 1$ pins, where n is the number of keys. This electrode forms one of the sides of a capacitor: it is at one and the same time a transmitter and receiver of electrical charges. The finger forms the second electrode: when it comes close, the capacitance of the capacitor increases considerably, and this change can be detected. Contrary to popular belief, there's no need whatever to be grounded or connected to the system ground for the detection to take place. Atmel calls this first detection method *QTouch* or *Self-Capacitance Keys*.

When the number of keys required exceeds ten or so, a second technique is to be preferred, referred to as mutual capacitance or QMatrix, with coupled electrodes (**Figure 2b**). Each sensor includes a driver electrode (X) and a receiving electrode (Y). In the absence of detection, part of the electrical charge emitted by electrode X is received by electrode Y, but as soon as a finger approaches, it upsets this balance by capturing part of the charge, thereby reducing the capacitance measured. Arranging these electrodes in a matrix makes it possible to create up to 64 keys with only 8 Xs and 8 Ys.

By combining three sensors appropriately laid out, one or other of these methods makes it

possible to easily produce sliders or rotors. However, designing interlaced electrodes like this does require particular care. These special considerations are very well detailed in the document [1], where you'll find all the rules to be respected. Right from the design stage, it is crucial to obtain the lowest possible capacitance in the absence of detection; to do this, we must take care to keep everything that can be well away from the sensors, especially the ground plane. For matrix electrodes, we need to take particular care with the routing of the Y lines. The X lines are less sensitive to interference.

Basis of operation

My touch surface comprises nine capacitive keys, arranged in an XY matrix. The square of 3×3 keys is ideal (**Figure 3**). Splitting it into two PCBs serves a dual purpose: it reduces the space required, while isolating the sensitive tracks as far as possible from stray coupling to the other components. The design of the electrodes has been optimised for insulating material that is 3 mm thick; the shape of the interlacing provides maximum coupling between the sensor electrodes, but there's nothing to stop you adapting it to your own requirements (**Figure 4**). The dimension 'e' represents the thickness of the insulating material placed over the keys. LEDs D1–D9, in 1206 format, are fitted backwards, on the back of this first PCB, to allow each zone to be backlit through a hole. This means each zone can be illuminated individually. Capacitors C1–C9, fitted in parallel, are essential. Without them, the sensitivity would be reduced. Lastly, five connectors J1–J5 make the connection between these two PCBs, carrying the X and Y signals, along with the two terminals for each LED.

The simplicity of the electronics (**Figure 5a**) required to make the sensors work is made possible thanks to the inclusion of the QTouch library in the ATtiny88 microcontroller from Atmel's AVR series. During the key acquisition phase, the driver electrode (X) is excited by a train of high-frequency square-wave pulses. The charges emitted by this copper plane are partly picked up by the adjacent Y electrode and are accumulated in C19–C21 via R14–R16. At the end of the pulse train, a timer is triggered in the microcontroller to measure the time taken for the capacitors to discharge through R11–R13. When you bring your finger close during an acquisition cycle, it absorbs part of the charge normally stored in the capacitor. So at the end of the pulse train, the voltage across the capacitor is lower and hence the time taken to discharge is likewise reduced, which the timer can then measure.

This classic and proven measurement cycle is then repeated by the microcontroller for each X-Y combination.

Since there is no requirement for accuracy, the device is clocked at 8 MHz by its own internal clock.

The assignment of the microcontroller pins to the matrix electrodes is governed by several constraints, which we won't go into here. Fortunately, Atmel provides the AVR QTouch Studio software to guide you in this step. The X electrodes are driven via R8–R10, which reduce electromagnetic emissions. Their value is non-critical.

Microcontroller outputs PD5 and PB0 driver the reed relays RL1 and RL2 via two cheap MOSFETs, not forgetting the essential flyback diodes D10 and D11. These are sometimes found built in to the relay, but I preferred keeping them separate so as to be able to adapt them for other devices. Two of the microcontroller inputs PB7 and PD7 will allow the system to learn the patterns to be recognized. For obvious security reasons, these will have to be hidden. Codes are

Bonus: Pattern Piano...

My daughters (18 months and 3 years) love playing with the prototype of the Pattern Lock, but it doesn't always prove to be very friendly and shows its discontent by lighting its central LED. Faced with my apprentice electronics technicians' frustration, I converted the ornery electronic lock by simple replacing relay RL1 by a little 8 Ω loudspeaker in series with a 22 Ω resistor (**Figure 9**). A few lines of code later, and the Pattern Piano was born, to their delight and that of your ears! The fundamental frequencies produced correspond to the notes between the C above middle C (525 Hz) and D of the octave above (1180 Hz) and will even let you play little tunes. Try it, it's fun!
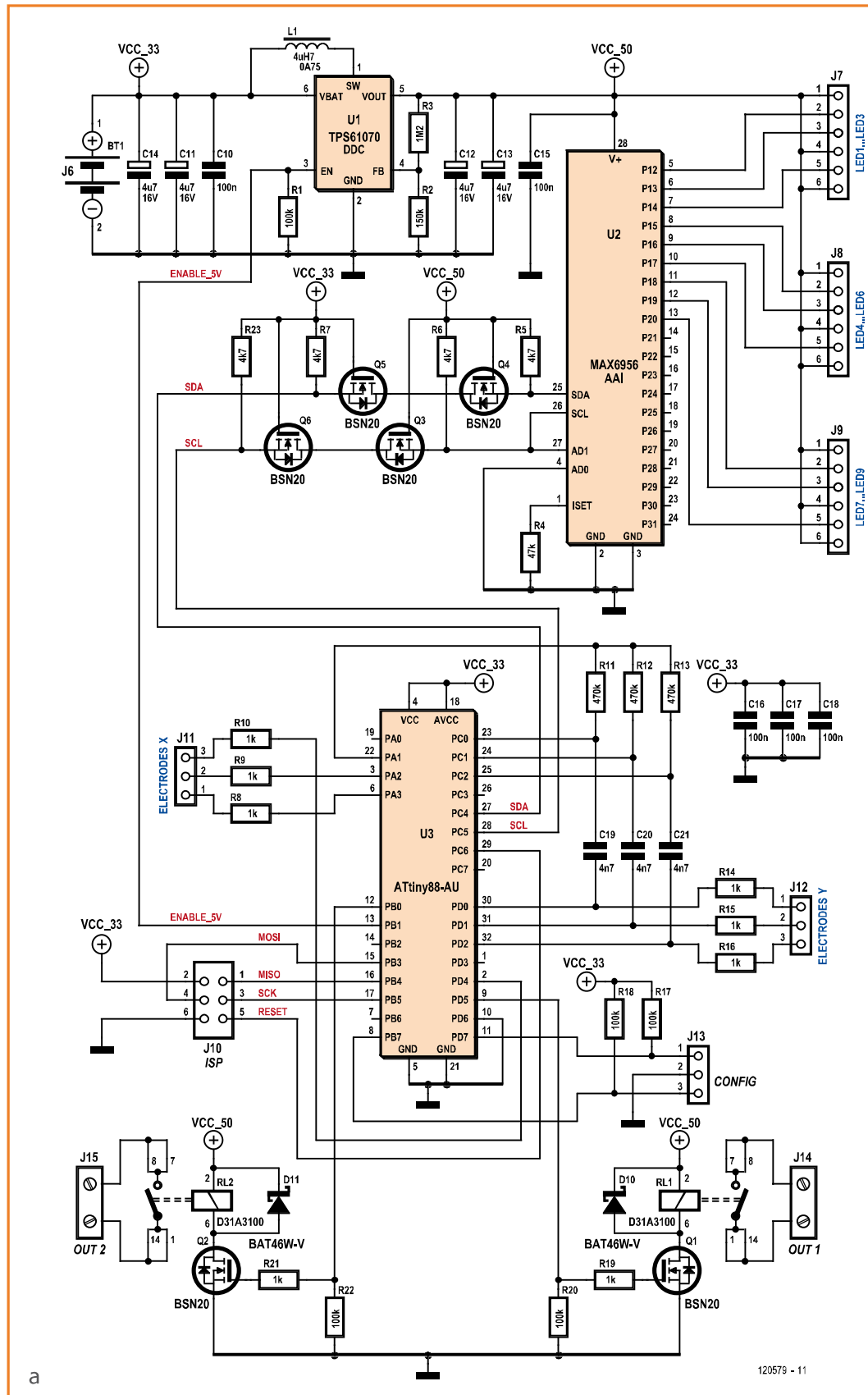
Figure 5.
Circuit diagram of the
pattern-recognition touch
pad.

a

120579 - 11

b
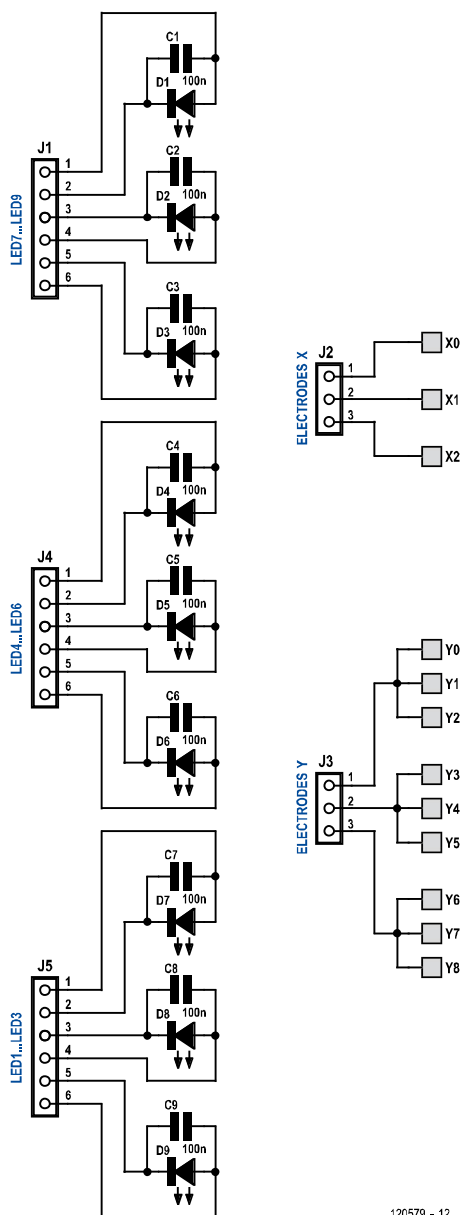
120579 - 12

programmed by fitting a jumper across two pins of J13 before applying power to the system.

The care taken in optimising power consumption means we can make do with just two AAA cells. The ATtiny88 microcontroller, which spends most of its time asleep, is powered directly from the batteries. Its built-in watchdog wakes it up every 1200 ms so it can execute a key acquisition cycle. If a presence is detected, it goes into active mode and turns on the 5 V rail supplied by boost converter U1 and its associated components. This rail powers the relays and the blue LEDs, whose direct voltage requirement of around 3.3 V is too high to be powered direct from the batteries. Unlike other boost converters, the TPS61070 has the advantage of disconnecting its output completely in stand-by mode, thus drawing only 0.5 µA!

For a touch-pad like this to be attractive, it not only needs to be illuminated, but this illumination also needs to be dynamic and follow the movements of the finger. The LEDs are driven by the microcontroller (**Figure 5b**), but it wouldn't be able to supply the LED current directly. This task is entrusted to a MAX6956, which makes it possible to drive up to 20 outputs with a constant current, individually configurable via the I²C bus. The maximum current from U2 is limited by R4 to around 20 mA per output. As the MAX6956 doesn't have a low-consumption stand-by mode, the power to it is cut off when the circuit is quiescent.

Q3–Q6 form two bidirectional level-shifters that allow I²C communication between the two ICs U2 and U3, powered from different voltages. In addition, they avoid U2's being powered from the SDA and SCL pins when the power to it is cut off. The BSN20 MOSFETs are particularly suitable for this low-voltage environment thanks to their low gate/source switching voltage (V_{gs}).

J10 is the ISP-compatible connector allowing *in situ* microcontroller programming and diagnostics.

Soldering and assembly

All the components should be available from local and online suppliers; they are almost all SMD, but I've opted for the larger 1206 packages to make soldering easier for the less experienced. Start by separating the two PCBs (**Figure 6**) using a circular disc cutter, a small saw, or a pair of cutters. Start by fitting the components on the upper board carrying the keys, the nine LEDs and

COMPONENT LIST

Main circuit
Resistors (SMD 1206)
R1,R17,R18,R20,R22 = 100kΩ
R2 = 150kΩ
R3 = 1.2MΩ
R4 = 47kΩ
R5,R6,R7,R23 = 4.7kΩ
R8,R9,R10,R14,R15,R16,R19,R21 = 1kΩ
R11,R12,R13 = 470kΩ

Capacitors (SMD 1206)
C10,C15–C18 = 100nF 50 V
C11-C14 = 4.7μF 16 V

C19,C20,C21 = 4.7nF 50V

Inductor
L1 = 4.7 μH 0.75 A, e.g. Bourns type SRR3011-
4R7YL; Farnell 1828144

Semiconductors
U1 = TPS61070DDC (Texas Instruments) Newark /
Farnell # 1461062
U2 = MAX6956AAI (Maxim IC) DigiKey #
MAX6956AAI+T-ND
U3 = ATtiny88-AU (Atmel), programmed, Newark /
Farnell # 1704570
D10,D11 = BAT46W-V
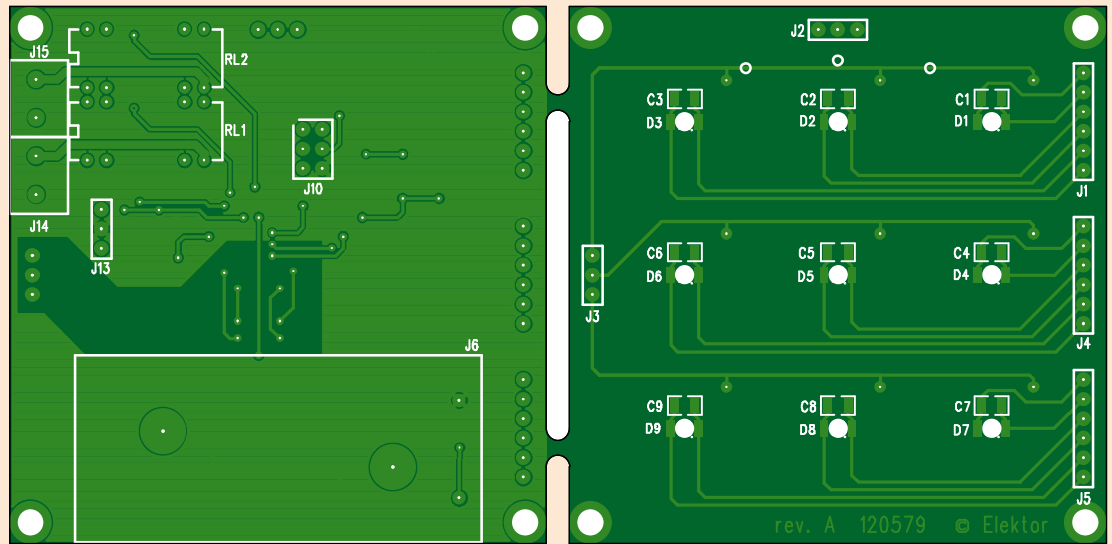
Figure 6.
PCBs.

nine capacitors. To make sure they are correctly aligned, before soldering them, plug together the male and female parts of connectors J1–J5, J7–J9, and J10-J12 on the stacked boards. Soldering U2 and U3 is tricky; I use the brute force method which involves first of all soldering all the pins together to the PCB using a large solder bit, without worrying about shorts; I then use desoldering braid to suck up the excess solder, leaving just the strict minimum required to hold the pins to the copper.

The mechanical assembly of the capacitive sensors needs particular care, all the more so because they are backlit. After some patient searching, I finally got my hands on some 3 mm (1/8") Acrylic sheet (also known as Perspex®; Plexiglass) from www.acrylite-shop.com (www. plexiglas-shop.com), which is able to diffuse the light from the LEDs into haloes of around 15 mm (⅝") diameter (see component list).

For the keys to work properly, there must be **no mechanical clearance** between the PCB and the dielectric (i.e. the essential acrylic sheet — no point trying without it, it just won't work!) I stuck these two layers together using transparent double-sided adhesive tape. A thin layer of epoxy adhesive over the whole surface would also do. You can experiment with other materials, transparent or not, as long as they are insulators — like glass, for example, which is the ideal dielectric. My sensor design is optimised for a thickness of 3 mm. If your dielectric sheet is thicker or thinner than that, detection will be less successful.

Connectors J1–J5 must be fitted in such a way that their pins are not flush with the capacitive board. They must be soldered on the side they are fitted. Otherwise, the Acrylic sheet would have to be cut away around the contacts, as they would be sticking out. This would be impractical and not very elegant.

Q1–Q6 = BSN20, Newark / Farnell # 1081309

Miscellaneous
J6 = battery holder for 2 AAA batteries, e.g. Keystone type 2468, Newark / Farnell # 1650679
J7,J8,J9 = 6-pin pinheader, 0.1'' pitch
J10 = 6-pin (2x3) pinheader, 0.1'' pitch
J11,J12,J13 = 3-pin pinheader, 0.1'' pitch
J14,J15 = PCB screw terminal, 5.0mm pitch, e.g. CamdenBoss type CTB5202/2, Newark / Farnell # 1717001
RL1,RL2 = reed relay, e.g. Celduc type D31A3100, Newark / Farnell # 1214549
PCB # 120579-1
Acrylic sheet, dim. 60x60x3 mm, PLEXIGLAS® led

(truLED) White WH72 GT, www.plexiglas-shop.com or www.acrylite-shop.com

Touch circuit
Capacitors (SMD 1206)
C1–C9 = 100nF 50V

Semiconductors
D1–D9 = LED, blue, SMD 1206, e.g. Kingbright type KPTD3216QBC-D; Newark / Farnell # 8530050

Miscellaneous
J1,J4,J5 = 6-pin pinheader, 0.1'' pitch
J2,J3 = 3-pin pinheader, 0.1'' pitch
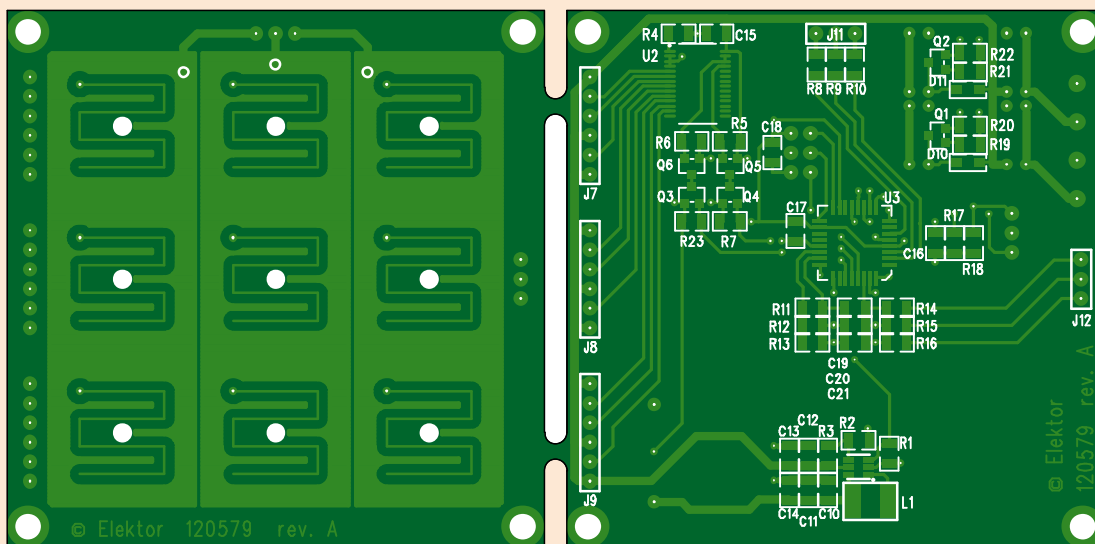
To indicate the position (and shape) of the sensors, you can use cutout adhesive vinyl or spray paint and masks, as you choose. The dimensions of the front panel I built (**Figure 7**) will give you something to start from, before you give free rein to your creativity.

Firmware and QTouch library

The simplicity of the hardware design of the capacitive detectors is made possible mainly through the use of several fairly elaborate algorithms in the QTouch library. This handles, for example, the periodic automatic calibration of the measurements and follows the slow changes in capacitance due to variations in environmental conditions (humidity). It also makes it possible to compare the values from adjacent keys, so as to keep only the highest one, which is very handy when the sensitive areas are small and our big fingers overlap (Adjacent Key Suppression or AKS algorithm). Lastly, to avoid the phenom-
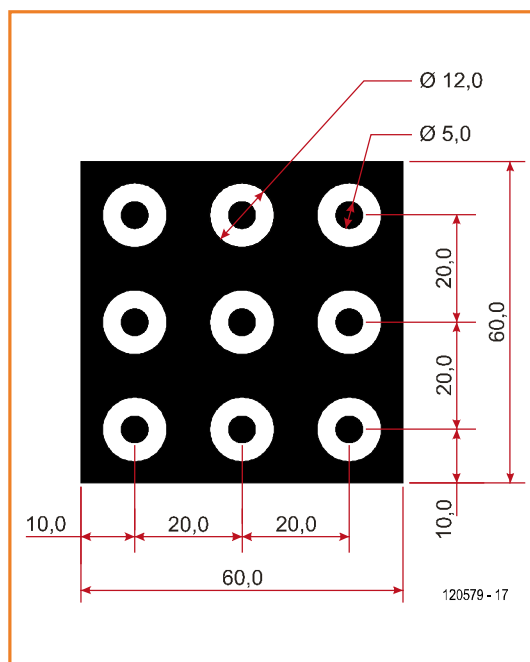
Figure 7.
Suggestion for a front panel design.

Make way for the capacitor!

My capacitive sensor interface has nothing in common with the resistive versions used in the earliest satnavs, which used two overlaid conductive films with a tiny gap between them. The slight pressure from the finger was enough to make a contact that allowed the finger to be detected. The sensitivity of this type of touch panel is mediocre compared to those using capacitive techniques. Nothing mechanical here, make way for the capacitor!

enon of a 'stuck' key (i.e. activated despite the absence of finger contact), a maximum duration can be configured, beyond which a fresh calibration is performed. These various parameters can be adjusted in the touch_config.h file; I kept the default values. Assigning the key numbers, their sensitivity, and the AKS group they belong to are defined in the QT_config_keys function in the QTouch.c file.

The software source code, with detailed comments, is available [3]. The incorporation of the QTouch library follows rules that are dictated by the document [2] and is illustrated by the pseudo-code available in **Figure 8**. At power-up, the system is initialized by calling the functions QT_config_sensors, qt_init_sensing, and QT_set_parameters. After that, a measurement is launched each time qt_measure_sensors is called. If a capacitance variation is detected, the library performs several acquisitions in quick succession in order to confirm or not if one of the keys has been activated. Repeating the acquisitions in this way is vital for measurement reliability, while still maintaining a fast response time. If the button state changes, the variable *u16_flagStatus* is updated, and the macro GET_SENSOR_STATE() is called to indicate the state of each of them. The measurement frequency will depend on your software, and is a compromise between responsiveness, processor busy time, and power consumption. To obtain a response that seems to correctly follows the finger sliding over the sensor surface, I chose an acquisition period of 50 ms, which makes the detection very fast. For a more conventional application where the keys are activated one by one, this delay can be increased to 100 or even 200 ms. Beyond that, the user will find detection slow or even erratic.

Surrounding the sensor detection, the rest of the software is implemented in a state machine that's too bulky to reproduce here but which I recommend you to download from [4] so you can study it.

Implementing the QTouch library occupies between 2 and 4 KB of program memory, depending on the configurations, and takes around 250 bytes of RAM. It also makes occasional use of Timer 1, so it's best to avoid using this during the measuring phases. The complete code for the Pattern Lock, compiled using IAR Embedded Workbench, occupies just under 6 KB of program memory and 500 bytes of RAM, and so fits perfectly in the ATtiny88.

Use and fine-tuning

The security offered by this system is a compromise between being tamper-proof and unlocking speed. It would be foolhardy to entrust it with the security of your front door. But securing use of the home computer would be a more suitable role for it. It was with this sort of application in mind that I opted for 100% self-contained powering, so as to leave as many options open as possible. Powering from batteries like this does impose a few compromises in order to achieve adequate battery life. The relays, major power consumers, are only activated for half a second once a pattern is recognized. This way of operating is suitable for replacing a computer on/off button or driving an electric door release. If you want to power the circuit all the time using a stabilized supply, all you need do is modify a few line of code so as to make the LEDs light and the relays operate permanently.

As soon as it is powered up, the system goes into standby with all lights out, but carries out periodic measurements. If it detects the presence of a finger on any zone for more than 1.2 s, it wakes up and lights the LEDs dimly. The user can then enter the unlocking code. If the geometric pattern is correct, an **O** is displayed briefly and the corresponding relay is energized. If the pattern is incorrect, an **X** will be displayed. The user has three tries to enter a code correctly, after which the lock is blocked and the central LED lights for 10 s, then it goes back into 'sleep' mode.

```c
#include "QTouch.h"
#include "touch_api.h"

void main(void){
  // Configure the Sensors as keys or Keys With Rotor/Sliders in this function
  QT_config_sensors();
  // initialise touch sensing
  qt_init_sensing();
  //  Set the parameters like recalibration threshold, Max_On_Duration etc
  QT_set_parameters();

  do{
    // Autonomous loop which performs capacitive sensing every 50ms
    if (g_u8_flagMeasure){
      g_u8_flagMeasure = 0;
      do{
        // one time measure touch sensors
        u16_flagStatus = qt_measure_sensors(g_u16_current_time_ms_touch);
        //Set flag if multiple measurements are required by library
        u16_flagBurst = u16_flagStatus & QTLIB_BURST_AGAIN;
      }
      while(u16_flagBurst);

      // One touch is in detect
      if (u16_flagStatus & QTLIB_IN_DETECT){
        u16_touchState = 0;
        //Find the touched key number
        for (i=0; i<9; i++){
          //Check if this key is in touch
          if (GET_SENSOR_STATE(i))
            u16_touchState |= (1<<i);
        }
      }
      else
        u16_touchState = 0;
    }

    //Process keys
    if (u16_touchState){
      //Look for which key is touched
      for (i=0; i<9; i++){
        if (u16_touchState & (1<<i)){
          u8_ledCurrent[i] = LED_CURRENT_HIGH;  //Update corresponding led
          MI_PlayTone(i);                       //Play Tone
        }
        else
          u8_ledCurrent[i] = LED_CURRENT_LOW;
      }
    }
    else
    {
      for (i=0; i<9; i++){
        u8_ledCurrent[i] = LED_CURRENT_LOW;
      }
      MI_StopTone();
    }
    LED_Update(u16_ledValue, u8_ledCurrent);          //Update led current
  }
  while (1);
}

//Timer 0 interrupt function, called every 50ms
__interrupt void MI_Timer0Interrupt(void){
  //Here every 50ms
  g_u16_current_time_ms_touch += 50;
  g_u8_flagMeasure = 1;

  return;
}
```

Figure 8.
Calling and using the QTouch library. The extract here comes from the PatternPiano software (Figure 9), a variant with a similar structure, but slimmed-down, and hence easier to understand.
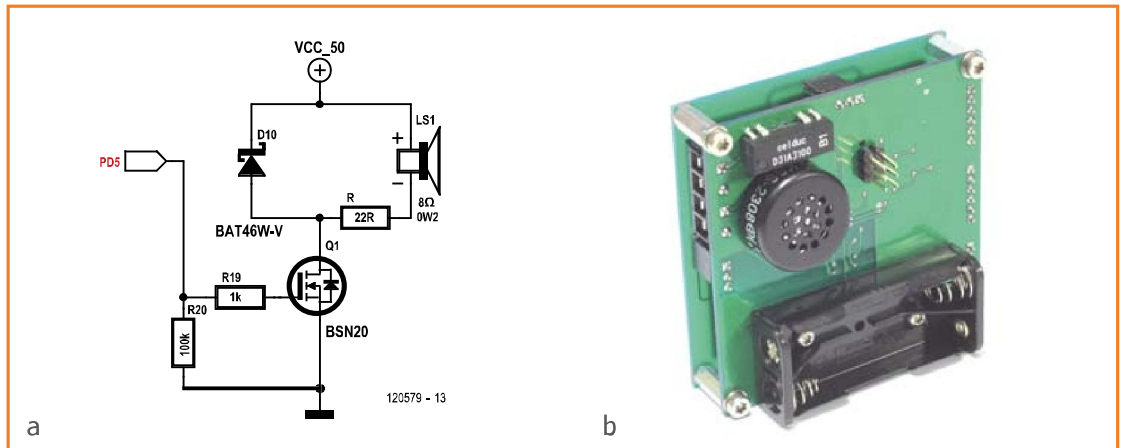
Figure 9.
Converting the Pattern Lock into a Pattern Piano.

a

b

To program a new pattern, when powering up for the first time or if you forget, disconnect the power then fit the jumper on J1: across pins 1 and 2 for energizing the first output, or across pins 2 and 3 for the second output. Power the system up (with the jumper still in place), wake it up by placing your finger on any button, then enter your new pattern, comprising four to nine points. Once the code has been memorized, the interface goes into 'sleep' mode. You can remove the jumper and use the lock.

The Y electrodes go via J3/J12. Avoid putting your fingers there or positioning metal objects close to these, so as not to upset the detection. In stand-by, the project consumes around 5 µA, punctuated every 1,200 ms by 8 ms wake-up periods @ 2.4 milliamps In this way, the average

current in stand-by comes out at around 21 µA – so using two AAA cells, battery life is several years! The access codes are saved in EEPROM and thus retained even if power is removed.

(120579)

Links & References

[1] Atmel - QTAN0079 Sensor design guide
    www.atmel.com/Images/doc10752.pdf

[2] Atmel - Atmel QTouch Library 5.0
    www.atmel.com/Images/doc8207.pdf

[3] Atmel – QtouchStudio
    www.atmel.com/tools/QTOUCHSTU-DIO4_3_1.aspx

[4] www.elektor.fr/120579

[5] david.ardouin.projects@gmail.com

Figure 10.
To ease the mounting of the sheet onto the keypad, solder the connectors from the underside so they do not protrude.

Low Cost. High Performance.

PC-based measurements from only £69

Multifunction DAQ

NI USB-6008

12-Bit, 10 kS/s, Multifunction DAQ

- 8 analogue inputs; 2 analogue outputs; 12 digital I/O; 32-bit counter
- Bus-powered for high mobility; built-in signal connectivity
- NI-DAQmx driver software and LabVIEW SignalExpress LE interactive data-logging software
- *£99*

ALSO AVAILABLE:

NI USB-6009 14-Bit, 48 kS/s, Multifunction DAQ - *£179*
NI USB-6210 16-Bit, 250 kS/s, Multifunction DAQ - *£399*
NI USB-9201 12-Bit, 8 Channels, 500 kS/s - *£549*

Sensor-based measurements

NI USB-TC01

Thermocouple Measurement Device - *£79*

ALSO AVAILABLE:

NI USB-9211A 24-Bit, 4 Channels, Thermocouple Input - *£529*
NI USB-9219 4 Channels, Universal Analogue Input - *£1049*

Instruments

NI USB-5132

50 MS/s Bus-Powered Digitiser/Oscilloscope - *£499*

ALSO AVAILABLE:

NI USB-5133 100 MS/s Bus-Powered Digitiser/Oscilloscope - *£699*
NI USB-4065 $6^1/_2$-Digit USB Digital Multimeter - *£1049*

Digital I/O

NI USB-6501

24 Channel, 8.5 mA, USB Digital I/O - *£69*

ALSO AVAILABLE:

NI USB-6525 8 Solid-State Relays, 8 DI, Counter, Channel-to-Channel Isolated - *£249*
NI USB-9421 8 Channels, 11-30V Digital Input - *£339*
NI USB-9472 8 Channels, 6-30V Digital Output - *£339*

Instrument Control

NI GPIB-USB-HS

GPIB Controller for Hi-Speed USB 2.0 - *£349*

ALSO AVAILABLE:

NI USB-232 Single-Port RS232 Interface for USB - *£129*
NI USB-485 Single-Port RS485 Interface for USB - *£129*
NI USB-8451 I2C/SPI Interface - *£339*

At lower cost than you might think, National Instruments provides an intuitive platform that can connect to thousands of sensors and instruments to acquire, analyse and process signals, and present results graphically. Whether programming in C, Basic or LabVIEW on Windows, Mac or Linux, get the latest in PC-based data acquisition, instrumentation and instrument control, starting at **less than £70**.

>> Buy online and explore the range at **uk.ni.com/lowcost**

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

Download '**The Complete Guide to Building a Measurement System**' at:
ni.com/data-acquisition/whatis

JOIN THE CONVERSATION: *Follow us on* *Search* **niukie**

NATIONAL INSTRUMENTS

500 ppm LCR Meter (2)
Part 2: standalone use, functionality, menus, and software
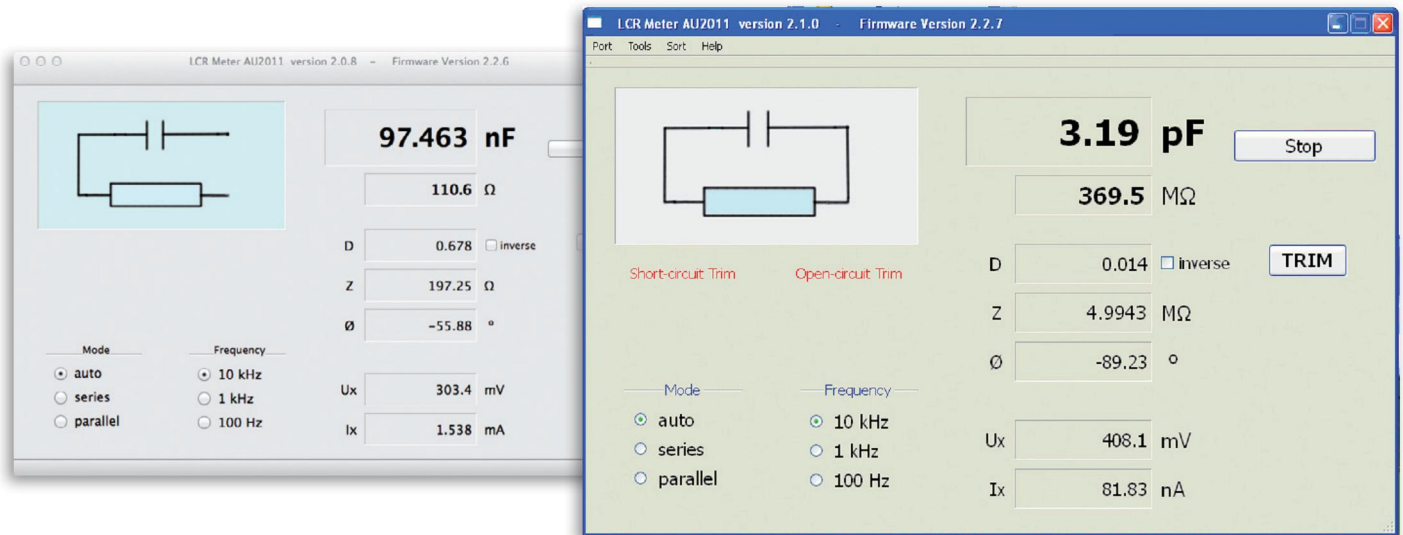
By
Jean-Jacques Aubry
(France)

In this second article, you're going to be discovering:

• the display & keypad extension for using the device without a PC

• the device's functions via its menus and the display

• the software, or at least certain of its more noteworthy aspects, as it's impossible to go into all its intricacies here (the description, source code, and executables are available in full from our website [1]).

To get the most out of this Part 2 article, it's best to have digested Part 1 and have the circuit diagram of the device in front of you.

Due to lack of space in the first article [1], we haven't yet shown the LCR meter's second (optional) board, which turns it into a self-contained device. But it's not very big and we'll soon get through it.

Display & keypad extension

At the outset, this LCR meter only worked when connected to a PC. Opinion is divided about the restrictions that entails. With some in favor of a stand-alone unit and some who aren't put off by being dependent on a computer, to keep everyone happy I added the extension at the request of Elektor Labs. The task was (relatively) easy, as not only did the previous AC powered version of the device already use a 128×64 pixel graphic display, but also the code for managing this display and the associated mini-keypad was 90% reusable.

The reduced number of port lines available (5 instead of 13 in the earlier version) obliged me to abandon parallel drive for the display in favour of serial drive. In addition, I needed to be able to automatically detect the presence of the extension and the type of powering (via the PC's USB port or via an external supply).

The circuit diagram for the extension is modest (**Figure 1**). The 128×64 pixel liquid crystal graphic display U1 (Displaytech) is managed internally by a Sitronix ST7565R controller.

Capacitors C1–C9 are used to convert the LCD drive voltage. The built-in backlight is driven by the main board via transistor Q1; it will be activated only in self-contained mode. The state of pin 6 on J16 (LCD_RES/) indicates the presence (0) or absence (1) of the extension.

LED D5, driven by transistor Q2, is just a clone of D6 on the main PCB: it flashes at the end of each measuring sequence.

The mini keypad comprises the five buttons K1–K5 and the matrixing diodes D1–D4. Three port lines offer eight different codes, of which seven are usable if we exclude the code 111 (= no button pressed). That's not many, but the LCR meter software will differentiate between a short and long press, which will increase the possible combinations.

The connection to the main PCB (via J16) is made using ribbon cable with a female connector on this end, but soldered at the extension end with J1, which is a transition connector soldered into the PCB, for reduced vertical clearance.

The functions of the buttons, which we'll come back to in detail later, are:

	Short press	Long press
K1	S/P/A	Menus
K2	Param.	Q-D
K3	Freq-	Sorting
K4		RT
K5	Freq+	Trim

So that's about it for the description of the hardware. Now it's time to discover how the LCD meter works, whether or not it is associated with a computer. The procedures for setting up and alignment are covered in two additional copiously-illustrated documents that you can download from the Elektor website [3]. We're not going to go here into details of all the adjustments and menus, but just say enough to make readers want to take the plunge and launch into building this unrivalled device.

PC mode

If the measuring head (for which the circuit diagram was published last month in Elektor) is connected to a computer via a USB link, PC mode is selected by default. The interface of the AU2011 program [1] installed on the host PC is in English by default, but supports translation files.
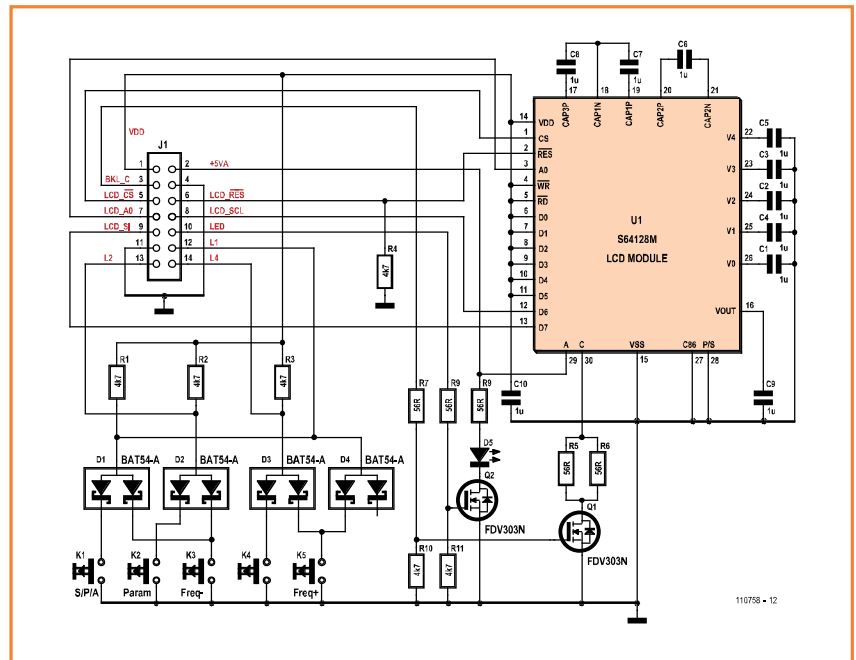
As the device is fitted with the FTDI USB-UART communication IC (FT232R), we need to install the driver for it. This operation is described in the LCR meter | Set-up document where the whole calibration procedure is given in careful detail.

Running the program and menus

Connect the LCR meter to the PC, run the user program on the PC and power up the measuring bridge, then follow the instructions in the set-up guide. Then come the **Preferences, Tools, Sort** menus (**Figure 2**), which we're not going to discuss in detail here. Let's instead take a closer look at the **User Interface** (**Figure 3**). The program's main window displays the measurement results, and a number of buttons let you modify certain parameters or initiate certain actions.

Figure 1.
The extension board includes five matrixed buttons and a graphics display.
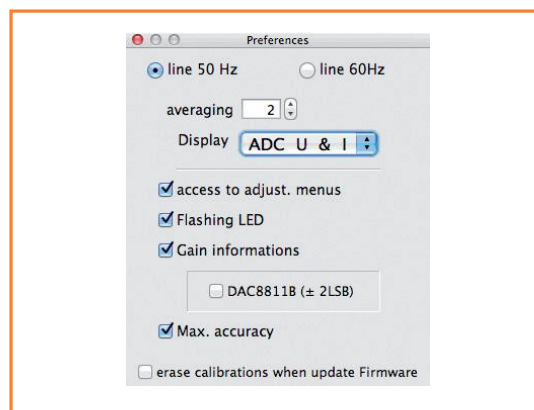
Figure 2a.
You have to start by certain inescapable choices in the Preferences.
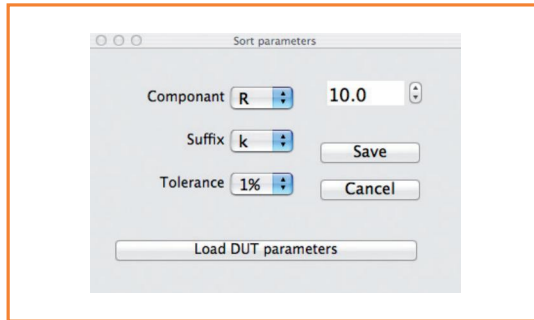
Figure 2b.
The Sort function requires prior configuration.
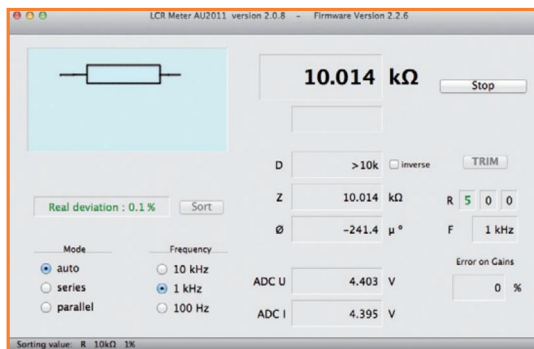
Figure 3.
The main window, here while sorting a resistor (set value displayed at bottom left).
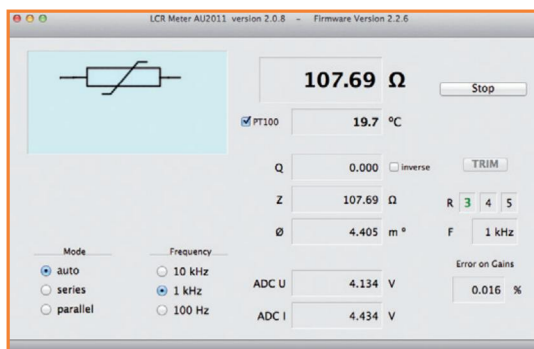
Figure 4.
A platinum probe allows accurate temperature measurement over a range from −80 °C to +600 °C.
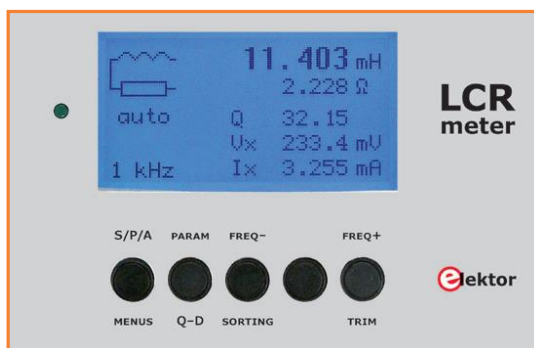
Figure 5.
The buttons and display in standalone mode.

At the top left, a pictogram shows the equivalent circuit of the component under test.

When the mouse pointer passes over a display field, a bubble appears giving information about what is displayed in this field.

At first time use, the **Open the port** button starts the procedure for connecting to the serial port defined (and memorized) in Preferences.

Once the connection has been established, the button is no longer active. From then on, the procedure is automatic and the button is not visible any longer.

As its name indicates, the **Start** or **Stop** button starts or stops the measurements.

TRIM button

Runs the OPEN – SHORT compensation. Compensation, which involves measuring the parameters of the measuring device itself (leads, clips), is carried out for each frequency. In the event of an error, the value is not used. A red symbol indicates when compensation has not yet been performed.

The **Sort** button starts the sort procedure. The reference value appears on the left in the strip across the bottom of the main window. The actual difference is displayed in green if the component tested is within tolerance or in red if it is not.

The **Mode** buttons let you select the mode for representing the component under test:

Auto: selection between series or parallel is performed automatically depending on the impedance of the component under test.

Series: forces series representation.

Parallel: forces parallel representation.

The test frequency is selected using the **Frequency** buttons. The lowest is twice the AC line frequency selected in Preferences.

The first **Gain** field indicates the measuring range:

in green for ranges 3–6 where the error due to the main amplifier is zero.

in magenta for ranges 2–7 (error due to the main amplifier ±0.02 %).

in red for ranges 1 and 8 (error due to the main amplifier ±0.04 %).

The other two fields give the values, between 0 and F, of the steps of the multiplying DAC for the *voltage* and *current* measurements.

If the **Max** check box is visible (here it is not), the gain error value is calculated using either typical values or using the Max value (for the PGA103), depending on whether it is checked or not.

The **Settings** menus let you perform a number of internal settings: offsets, calibrations, etc. The first three don't require you to take the lid off. Please refer to the document LCR meter | Set-up.

Measuring a PT100 platinum probe

There are lots of devices around offering a tem-

What could be more satisfying than building your own tools?

perature display, but only with an accuracy of several degrees! I thought it would be interesting to have a true reference available. Thus a platinum probe lets us measure temperature in a range from −80 °C to +600 °C with an accuracy of ±0.3 °C (class B) or ±0.1 °C (class A). When the value of a resistor lies between 70 Ω and 300 Ω, the frequency is ≤ 1 kHz, and the Q (quality factor) is practically zero, a box marked **PT100** appears to the left of the secondary parameter display field (**Figure 4**).

If you check this, the temperature value is shown for a PT100 platinum probe with this resistance. The symbol changes to a thermistor one. Within the measuring range used (−75 °C to +557 °C), the accuracy of the resistance/temperature conversion is < 0.05 °C.

Standalone mode

Fitted with the display & keypad extension, our LCR meter becomes a standalone unit. No need for a PC (except for performing the preliminary settings, which cannot be done in standalone mode). In this mode, all the texts displayed are in English.

The five buttons (**Figure 5**) let you select the device functions and configure certain parameters. The buttons have different functions depending on whether your press them quickly, or hold them in for more than two seconds (indicated by the green LED). The primary functions are in black (above the buttons on my prototype), while the secondary functions (long press) are in blue. In the menus, the (variable) functions of the buttons are indicated at the bottom of the liquid crystal display.

Primary functions

S/P/A: Choice of model between series or parallel equivalent circuits for calculating the parameters, or the automatic model mode.

Param.: Display options:

modulus of the impedance $|Z|$ and phase angle Φ.

equivalent series resistance R_s and equivalent series reactance X_s.

device's rms voltage V_x and rms current I_x.

Freq− & Freq+: To select measurement frequency.
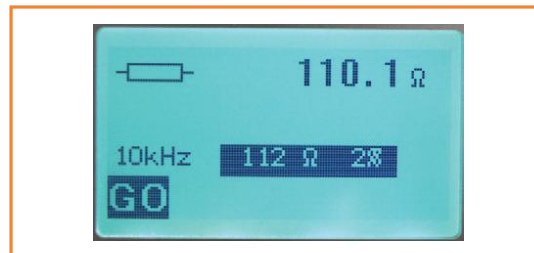
Figure 6.
Sort function.
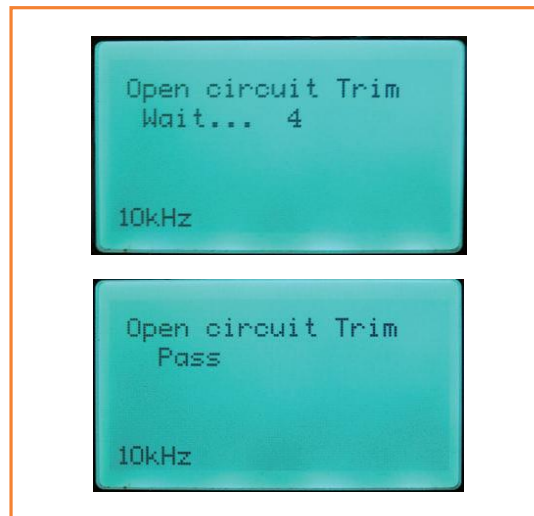
Figure 7.
Trim function.

Secondary functions

Menus: Hold this key pressed in to go into **Menu** mode.

Q_D: Holding this key in toggles the automatic display of **Q**uality → **D**issipation or **D**issipation → **Q**uality.

Sorting: You first have to define the sort parameters via the Sorting Parameters menu.

Holding this key in lets you enable or disable the **Sort** mode.

The value of the component being sorted is displayed with its symbol (**Figure 6**).

The sort parameters are displayed in reverse order.

Pressing the [GO] key starts the comparison, with momentary display of the actual tolerance and the result (OK or BAD!)

Trim: Pressing and holding this key lets you run the compensation (*OPEN – SHORT*) (**Figure 7**). Compensations are performed for each frequency. If $|Z|$ is < 10 Ω, the SHORT calibration is performed.

If $|Z|$ is > 100 kΩ, the OPEN calibration is performed.

If either or both of the calibrations is invalid, a flashing symbol appears in front of the primary parameter value. Re-do the incorrect calibration(s) (**Figure 8**).

Unmarked button: the function of this button is explained in the document *LCR meter | Operating Instructions* [3].
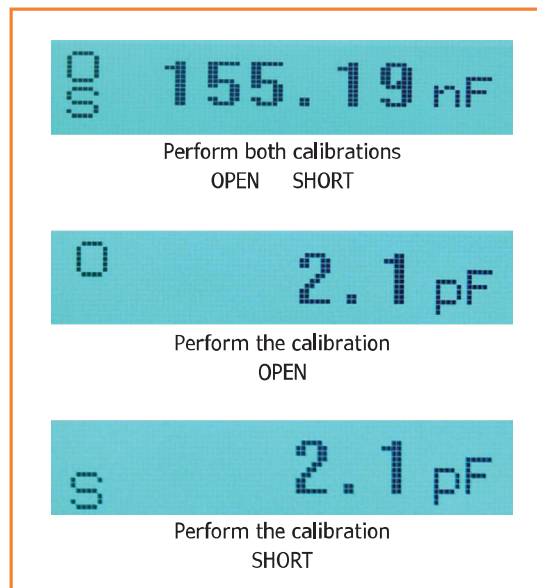
Figure 8.
Failed calibrations

Figure 9.
The choice of menus.
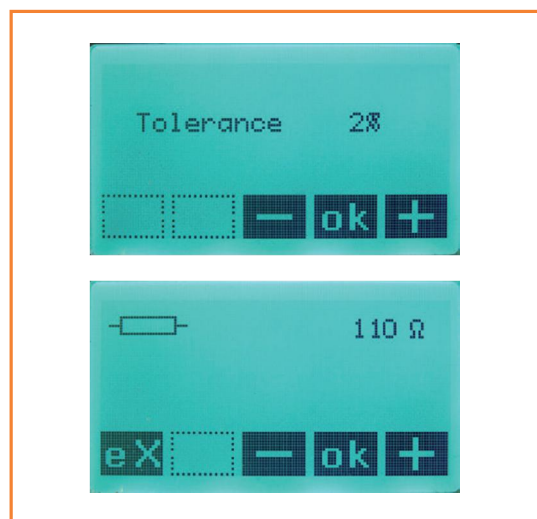
Figure 10.
Sort tolerance and primary sort parameter.

Menus (Figure 9)
Navigation is performed using the ↑ and ↓ keys. The > symbol indicates which menu will be selected by [ok].
Pressing the [eX] (*exit*) key takes you out of *menu* mode.

Sorting Parameters
This menu lets you select the parameters (tolerance and value) used for sorting components. The value proposed will be that of the primary parameter of the reference component connected **before** going into menu mode.
The first step lets you select the tolerance using the [−] and [+] keys, then confirming with [ok]. (**Figure 10**)
The next step lets you adjust the value that will be used as the sort value, starting off from the **primary parameter** of the component connected.
[−] and [+] keys, confirmed with [ok]. (**Figure 10**)

In the online operating instructions, you'll find other functions that we're not going to describe in detail here: *Averaging, Display Range, Adjust Contrast, Back Light, Line freq. 60/50 Hz.* See also the paragraph on **Measuring a PT100 platinum probe** in the PC mode section above — this function is also available in **standalone** mode.

Programs
The overall performance of the LCR meter (measurement accuracy, ease of use) is very much dependent on the quality of the three programs used:
the bootloader, which takes command as soon as the measuring bridge is powered up.
the firmware, the most important part of the internal program, which performs all the work of measurement acquisition, calculating the DFT, etc. It also drives the display in standalone mode.
the external AU2011 program running on the connected computer, which in "PC" mode makes it possible to display the results, run commands, etc. This dialogues with the bootloader and then the firmware by sending and receiving messages in the form of predefined character strings.

Bootloader
The bootloader, which runs automatically at power-up, is permanently resident in memory

from the address 0x0000. It's main function is to update the main program (firmware):
if jumper J17 is in place (unconditional update, e.g. in the event of the firmware's having crashed);
during a software reset, when the user requests the update;
if the integrity check on the main program code in EEPROM is negative.
The AU2011 program can only perform this update **if the device is in PC mode**.

The first step is to erase the program memory (all bytes take the value 0xFF), then the file in IntelHEX format is received over the USB-UART link. After checking that the operation has been correctly completed, the new firmware is run with the help of a function pointer to the firmware code start address:

```
((void (code *) (void)) PROG_BEGIN_FLASH_
ADDR) () // -> jump to application code
```

Listing # 1

```c
void UART0_ISR(void) interrupt INTERRUPT_UART0
{
    char SFRPAGE_SAVE = SFRPAGE;
    SFRPAGE = UART0_PAGE;
    if (RI0 == 1)     // if receive flag set, put the byte in  UART_InputBuffer
    {
    RI0 = 0;
    Byte = SBUF0;    // Read a character from UART0
    if ((Byte == '\0') || (Byte == '\n') || (Byte == '\r'))
    {

    RX_Ready = 1;                       // Reception complete
    }
    else
    if (UART_InputBuffer_Size < UART_IN_BUFFERSIZE - 1)    // room needed for string terminal '\0'
    {

    UART_InputBuffer[UART_InputBuffer_Size] = Byte;

    UART_InputBuffer_Size++;          // Update array's size
    }
    }
    if (TI0 == 1)
    // if transmit flag set, send UART_OutputBuffer datas
    {
    TI0 = 0;
    if (UART_OutputBuffer_Size > 0)    // If buffer not empty
    {

    Byte = UART_OutputBuffer[UART_Output_First];

    SBUF0 = Byte;                     // Transmit to UART0

    UART_Output_First++;              // Update counter

    UART_OutputBuffer_Size--;         // Decrease array size
    }
    else
    {

    UART_Output_First = 0;

    TX_Ready = 1;                     // Transmission complete
    }
    }
    SFRPAGE = SFRPAGE_SAVE;     // restore SFRPAGE
}
```

The update part of the firmware comes directly from Silicon Labs application note AN112, but with a modification that is required for the checksum of the memory occupied by the newly-installed program. In point of fact, since a number of 'holes' (with value 0xFF) may exist in the newly-written memory space, this can lead to a difference between the actual size of the program and the calculation performed by end_address — write start_address!

For the other tasks performed by the bootloader, you'll need to refer to the full source code, which you can download from our website [1].

Firmware
This is the heart (and brain!) of the device. It resides in memory from address 0x2000. It carries out the measurements, responds to the user's commands, and either displays the results (in standalone mode) or sends them over the USB link (in PC mode).

Only a few small sections of the code are described here; the full and copiously-annotated source code of the firmware can be found on our website [1].

In PC mode, dialogue with the AU2011 program occurs by exchanging messages. The character strings received or sent are handled by an interrupt routine of the UART0.

One of the most worthwhile features of the micro-controller chosen is direct storage into XRAM (**D**irect **M**emory **A**ccess) of data coming from by ADC0 and ADC1. The DMA interface is programmable for, among other things, writing the data from an address in XRAM and for a certain number of acquisitions of these data. This pro-

Listing # 2

```
void TIMER2_ISR(void) interrupt INTERRUPT_TIMER2
{
     char SFRPAGE_SAVE = SFRPAGE;
     SFRPAGE = TMR2_PAGE;
     TF2 = 0;                          // Immediately reset Interrupt flag
     if (TMR2CF & 0x04)                // if state of the output = 1
     {

        ET2 = 0;                       // Disable Timer2 Interrupt
        SFRPAGE = DMA0_PAGE;           // Switch to DMA0 Page
        DMA0EN = 1;                    // Begin Executing DMA Ops (which will enable ADC0)
     }
     SFRPAGE = SFRPAGE_SAVE;
     // restore SFRPAGE
}
void DMA0_Acquire_Samples (void)
{
     char SFRPAGE_SAVE = SFRPAGE;
     // --- reset TF2 flag just prior enabling Timer2 interrupt
     SFRPAGE = TMR2_PAGE;
     TF2 = 0;
     ET2 = 1;                          // Enable Timer2 Interrupt
     SFRPAGE = DMA0_PAGE;              // Switch to DMA0 Page
     // --- Timer2 interrupt enable DMA-ADC operation
     while (!DMA0INT);                 // wait for DMA operations are complete
     DMA0EN = 0;                       // Stop Executing DMA Ops
     DMA0INT = 0;                      // Clear DMA0INT bit
     SFRPAGE = SFRPAGE_SAVE;           // restore SFRPAGE
}
```

Comparative measurements with a professional measuring bridge

By Ton Giesberts (Elektor Labs)

Measuring inductances is a science in itself. Inductor data sheets often give the current and frequency at which the inductance value is specified. For example, if you measure at 1 kHz instead of 10 kHz as given by the manufacturer and with a different measuring current, you will obtain a noticeably different inductance value. Which is not necessarily a disaster, since chokes have a tolerance of ±20% anyway. In the Elektor labs, I compared the behaviour of our new LCR meter designed by Jean-Jacques Aubry with the one in a very big lab measuring device costing over $2,000 (£1,300): the Hameg Programmable LCR Bridge 8118. The measuring current in this reference device is adjustable; during an initial measurement, its current was ten times that of our little device. Depending on

the material of the inductor, this may lead to considerable differences. But we remained comfortably within the range of ±20 %.

A first measurement made on the *Hameg* for a 100 µH/5 A choke gives 108.7 µH. Reducing the measuring current (and for conscience's sake performing a new calibration), the same device then indicated 97.6 µH. Now you're bound to be wondering what the new Elektor LCR Meter was reading: 96.8 µH. Not bad, eh?

gramming is carried out once and for all by a few instructions in certain of the registers. All that then remains is to send the start command and wait for the end signal: all the work of acquiring data and transferring them into memory is automatic and superbly optimised, making it possible to achieve an acquisition speed of 1 mega-acquisitions per second (16-bit data, i.e. two 8-bit words) using a 24 MHz clock!

This process is organized by the interrupts requested by:

- Timer2 (frequency of the sinewave signal) → starts a sequence of N acquisitions by setting the DMA0ENable flag to 1.
- N = number of samples per cycle of Timer2 × number of cycles of Timer2.
- Timer3 (frequency of Timer2 × number of samples per cycle of Timer2 / 2) → automatically starts an acquisition; the coefficient of 2 is due to the fact that there are two interrupts per cycle of Timer3.

The sequence end is indicated by the DMA0INT flag going to 1. All that remains is to read the XRAM and use the data.

We saw in the first article that it was necessary to compensate for the offsets in opamps U6 (input_offset) and U1 (sine_offset). As we have two 12-bit DACs available, this adjustment can be automated. We just need to unplug the measuring cables so the process is not upset by stray signals (among others, from the power lines).

As for all the device's other adjustments, these can only be performed in **PC mode** via the AU2011 program. They are described in the document LCR meter | Operating Instructions that you can download free and which we strongly recommend you to read [3].

AU2011 program

The AU2011 software runs on a personal computer. It is written in C++ and uses the Qt and qserialdevice/AbstractSerial libraries in order to obtain an executable for PC under Windows, Mac OSX, or Linux, depending on the compilation target. The source code and a number of executables are available from our website [1]. If you want to modify and/or compile the source code, the IDE to use is Qt Creator. Conditional compilation options (#ifdef) in the source code allow you to manage the various targets directly. The User Interface (UI) is very slightly different depending on the compilation target, because of the differences in the character fonts, among other things; thus each target has its own AU2011_mainwindow.ui file.

The core of the user interface is in English, but it is designed to be able to use translation files. The French version (AU2011_fr.qm) already exists. Placed in the same directory as the AU2011 program, together with the *qt_fr.qm* file (a translation file specific to Qt that is located in the *QtSDK/Desktop/Qt/4.8.0/gcc/translations* direc-

A major advantage of the microcontroller chosen is the storage of the data from ADC0 and ADC1 directly into XRAM (Direct Memory Access)

tory, along with those for other languages), this makes it possible to have all the messages and the UI in French **for a French system**.

To obtain the same thing for another **local** language, you need to:

either (in the AU2011_Projet.pro file) modify the directive TRANSLATIONS += AU2011_fr.ts to AU2011_xx.ts where xx is the symbol for the local language (de, da, cs, etc.) and recompile to produce this new file;

or else duplicate the AU2011_fr.ts file and rename the copy AU2011_xx.ts.

Then you need to use the *Linguist* program to translate all the strings, and then lastly run the command *lrelease* to produce the *AU2011_xx.qm* file.

Refer to the Qt *Linguist* documentation for more details [2].

Our readers hopefully will gradually offer new translation files!

Let's remember that the dialogue between the AU2011 software and the LCR meter takes place via the USB link, by emulating a serial link thanks to the driver offered by FTDI (115,200 baud, 8 bits, no parity bit, 1 stop bit, no flow control). The commands sent to the device or received from it are character strings defined in the *AU2011_mainwindow.h* header file [1].

After the communication port has been opened and the bootloader synchronization command has been sent, the software waits to receive the first character string containing the firmware version number; it displays this information in the window title bar. Then will come the string formed from the initialisation parameters:

power line frequency 50 or 60 Hz (**L5** or **L6**)

test frequency (**F1** or **F2** or **F3**) for 100/120 Hz or 1 kHz or 10 kHz

Trim Short performed or not (**S1** or **S0**)

Trim Open performed or not (**O1** or **O0**)

averaging value (**A1** to **A9**)

Then the firmware goes into its event loop and monitors the arrival of a command from the UART (interrupt routine). If a request to perform the measurements has been received, the resulting

parameters are sent to the software in the form of a long string of characters formed using the values of:

"**Rs Xs Freq ranges Vpp Ipp ADC_Vpp ADC_Ipp**"

plus the character 'C' or 'Z' depending on whether the is component is capacitive or not.

When this string is received, read by the **slotRead()** interrupt routine, the **ParseInputString()** function breaks it down and the display functions **Display_xxx()** fill in the relevant fields. One important routine, among others, is *Convert_Value_to_String()*, which converts a floating numerical value into a character string with mantissa and exponent in the form of a standardized suffix, e.g. **12.05 nF** for a capacitor with a value of 1.205×10^{-8} (in farads).

You can almost smell the solder (at last!)

In the next issue, we'll at last be showing you the two boards, which are going to be available as ready-to-use tested modules. For significantly less than \$275 / €200, you'll be able to get your hands on a marvel of accuracy that will never again leave your workbench. There's nothing more satisfying than building your own tools! Between now and then, we invite you to take a look at the online documentation, which will give you both an overall and a very detailed view of our automatic impedance measuring bridge.

(130022)

[1] Downloadable software (bootloader, firmware and main program):
www.elektor-magazine.com/110758

[2] Qt Linguist
http://goo.gl/fIYQh
[or]
http://qt-project.org/doc/qt-4.8/linguist-translators.html

[3] On-line documentation:
LCR meter | Set-up
LCR meter | Operating Instructions
www.elektor-magazine.com/110758

Elektor Linux Board Extension

Display, buttons, real-time clock, buzzer and sixteen GPIOs
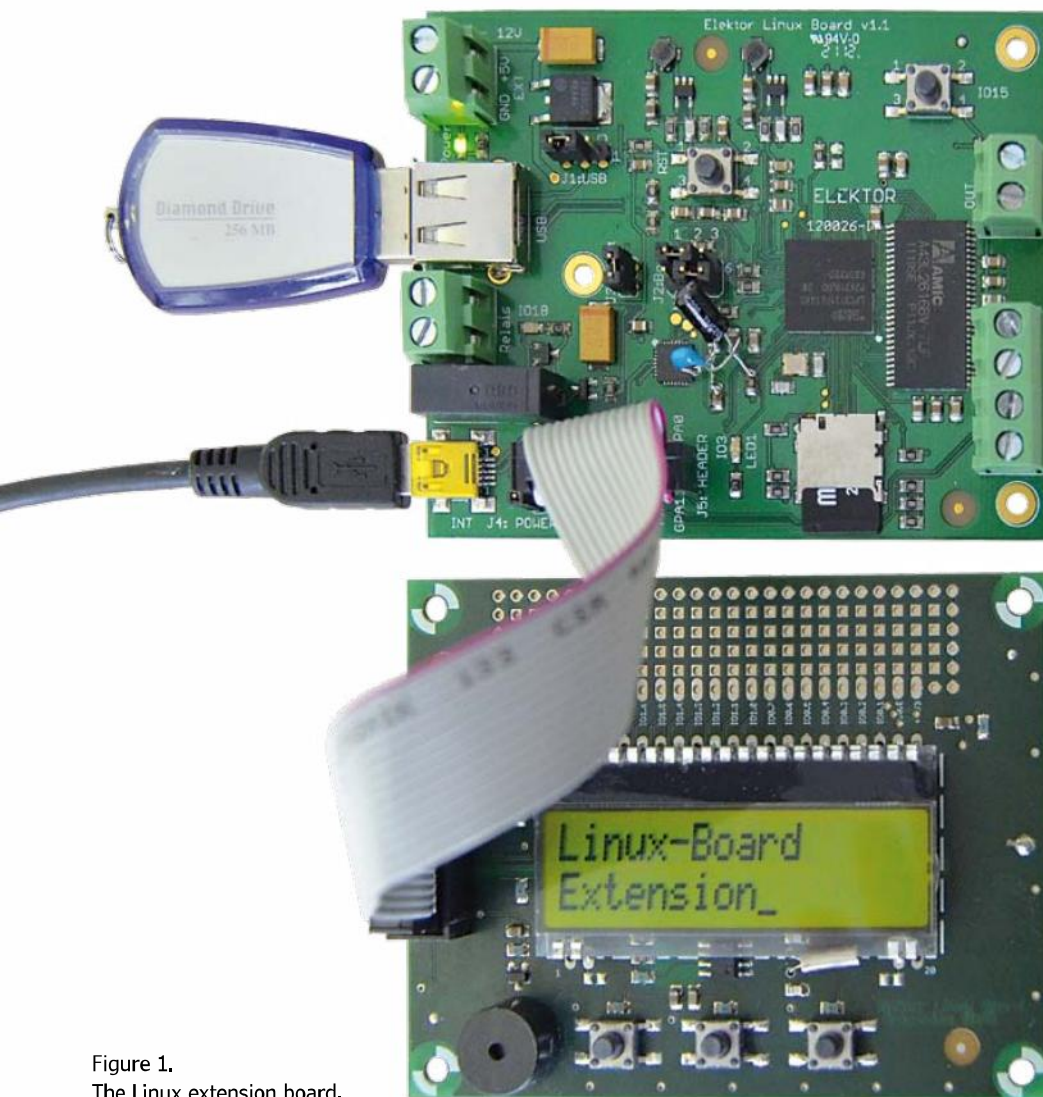
Figure 1.
The Linux extension board.

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

We have already seen how to make a user interface for an Elektor Linux board application using modern web technologies. However, it may happen that we want to be able to control the board but do not have handy access to a PC and browser. In such situations a display and a couple of buttons can be more than adequate for showing and adjusting the most important parameters of the system, and this article provides the means: an extension module for the Elektor Linux board.

The Linux extension board includes everything needed to provide the user interface for a wide variety of projects. The buttons (**Figure 1**) provide inputs and a two-by-sixteen alphanumeric display shows outputs. The buttons are located directly below the display and so it is possible to implement a simple menu system. The board also sports a buzzer for acoustic feedback, a real-time clock (RTC) with battery back-up, and a port expander providing 16 extra digital inputs

Table 1. Signals on the 14-way connector						
Pin	Name	Use	Use	Name	Pin
1	GPA0	Left button		Middle button	GPA1	2
3	GPA3	Right button		Buzzer	PWM	4
5	SCL (I2C)	RTC and port expander		RTC and port expander	SDA (I2C)	6
7	MOSI (SPI)	Display		Display	MISO (SPI)	8
9	System Clock	-		Display	SCK (SPI)	10
11	GPIO14	Display		Display	GPIO11	12
13	3.3V	Power		Power	GND	14

and outputs. There is also a prototyping area measuring about 15 mm by 60 mm to add your own circuitry.

Connector J5 [2] on the Elektor Linux board is a two-by-seven (14-pin) pinheader designed to allow for expansion (see **Figure 2**). The same connector features on the extension board, so the two can be joined using a fourteen-way ribbon cable. The cable carries power at 3.3 V to the extension board as well as data signals.

Table 1 shows the pinout of the header. The extension board can of course be used in a wide range of other projects: for more information see the article 'Embedded Firmware Library' elsewhere in this issue.

The Linux extension board is available from Elektor populated with the SMD components and tested. All that remains is to solder the LCD and a couple of other components [3].

Circuit diagram

The circuit diagram is shown in **Figure 3**. Connection to the Elektor Linux board is via K1. Since only a 3.3 V supply is available, all the components on the board are chosen to be able to run on this voltage. When selecting the display we looked only at those which could be controlled over I2C or SPI (most alphanumeric displays can only be driven in four-bit or eight-bit parallel modes). We settled on the DOGM162L-A [4] from Electronic Assembly, which features an SPI interface.

Three analogue inputs (GPA0, GPA1 and GPA3) are brought out to the expansion connector on the Linux board but are not needed for the main features of the extension board. We have therefore simply connected them to buttons. An alternative would be to connect a rotary switch with voltage divider resistors to allow the position to
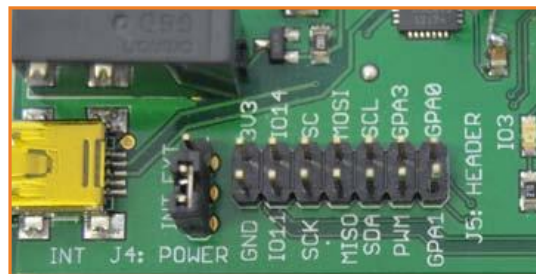
Figure 2.
The Elektor Linux board is connected to the extension board using its fourteen-way connector.

be determined. With the buttons there are just two possibilities: the 'high' state (A/D converter reads maximum voltage) and the 'low' state (A/D converter reads zero), the latter condition corresponding to the button being pressed.

The buzzer is connected to FET T1, which in turn is driven by a PWM output of the LPC3131. This allows simple tones of constant frequency to be generated easily. Since the buzzer is only driven from 3.3 V, the volume is not as loud as it could be: for maximum volume a supply of 8 V to 15 V is required (see the datasheet at [5]).

Also on the board is real-time clock IC2, an MCP79401 [6]. This is equipped with battery back-up in the form of a small coin cell, so accurate time is kept even when the device is off.

Finally we have IC1, a PCA9555 port expander [7], which provides up to sixteen additional digital inputs and outputs controlled over an I2C interface. These inputs and outputs can be used for connecting status LEDs or extra buttons, or even a key matrix.

Display

To make it easy for you to develop your own applications using the display we have written a small command-line program in C. The code can be downloaded as source code or in ready-compiled form and copied onto the SD card. The
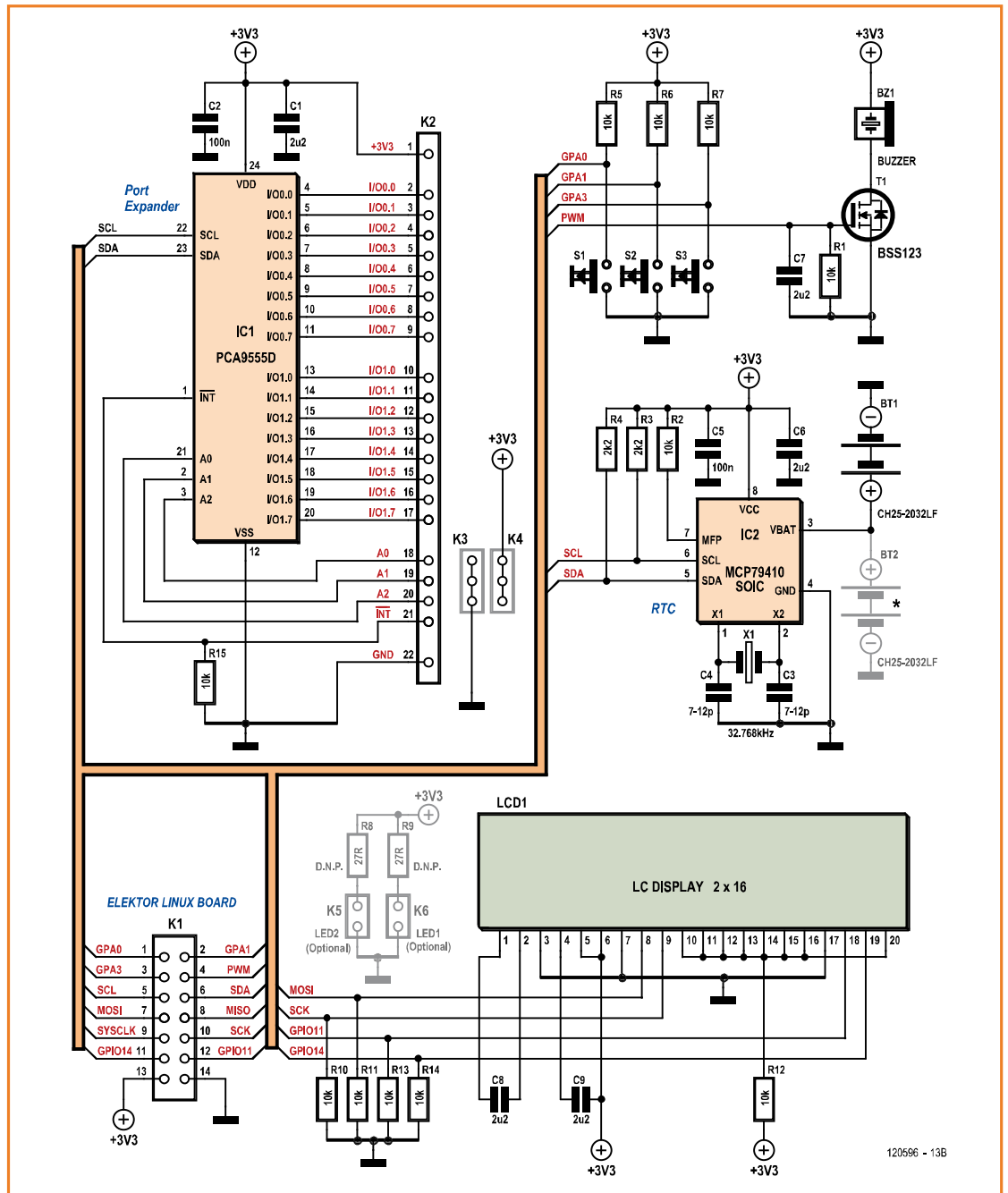
Figure 3.
Circuit diagram of the Linux extension board.

easiest way to do this is with the help of a USB memory stick. First download the file [8] onto the development PC (making sure the URL is entered on one line without any breaks) and then copy it to the USB stick. Then insert the stick into the Linux board (**Figure 4**). The console should display a message when the stick is inserted. Mount the USB stick in the file system:

```
mount /dev/sda1 /mnt/
```

So that you can use the program whenever you want (and not just when the USB stick is present) copy it to the local file system on the SD card:

```
cp /mnt/dog-app /usr/local/bin
```

For a quick test of the program try running it from the console. The command

```
dog-app -h
```

should display help text regarding the program.

Now connect the extension board to the Linux board using the ribbon cable. Make sure the red mark on the cable is at the same end as GPA0 and GPA1. You can write text to the display with a command such as

```
dog-app -n -w "Hello Elektor"
```

using ordinary ('straight') double quotation marks. The text 'Hello Elektor' should appear on the display.

Writing to the second line of the display requires a little understanding of how the display works. Each line of the display consists of a number of segments, and the cursor can be positioned at any desired segment. The first segment in the first row has an 'offset' of 128, and the first segment in the second row an offset of 192. The following command will write text to the second row of the display:

```
dog-app -o "192" -w "Hello 2"
```

Buttons

The buttons are connected to the A/D converter inputs of the LPC3131. We described how to use the A/D converter in an earlier article in the *Elektor* Linux board series.
The following command gives access to the leftmost button:

```
echo 0 > /dev/lpc313x_adc
```

If you now read the file /dev/lpc313x_adc (for example, using the 'cat' command) you should see a large value (in the region of 1023) if the button is not pressed:

```
root@gnublin:~# cat /dev/lpc313x_adc

0x3ff
```

If the button is pressed, the result should be zero:

```
root@gnublin:~# cat /dev/lpc313x_adc

0x000
```

To select the middle button for reading, use the command

```
echo 1 > /dev/lpc313x_adc
```

and for the rightmost button use

```
echo 3 > /dev/lpc313x_adc
```
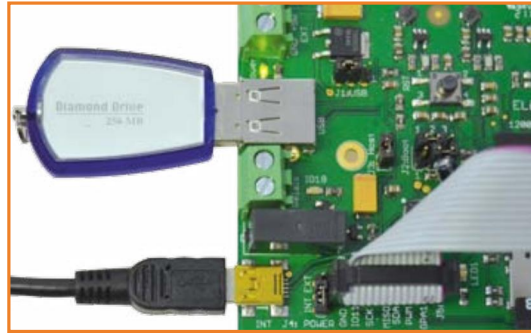

Figure 4.
The Elektor Linux board with a USB memory stick.

Buzzer

It is very simple to generate a tone via the file system:

```
echo ff > /dev/lpc313x_pwm
```

To turn it off again:

```
echo 00 > /dev/lpc313x_pwm
```

Brief beeps can be generated using the above commands in combination with the 'usleep' command, which provides a delay of a specified length.

Real-time clock

The RTC requires its own kernel driver. This is already available in the standard file system and can be loaded using the following command:

```
modprobe rtc-mcp7940
```

To arrange for the driver to be loaded automatically when the system is started up, add the following line to the file /etc/modules:
```
rtc-mcp7940
```
The kernel must be told the address of the RTC device on the I^{2}C bus. This is done as follows:

```
echo mcp7940 0x6f > /sys/bus/i2c/devices/
i2c-1/new_device
```

Next we need to check what major number the kernel has given the device so we can configure the device file correctly.

```
cat /proc/devices
```

Normally the assigned major number will be 251. So, to configure the device file:

```
mknod /dev/rtc0 c 251 0
```

We now set the system date and time manually. The format is [MMDDhhmm[[CC]YY]], so for example

Figure 5.
Setting the system time
and date from the RTC at
start-up.


```
case "$1" in
        start)
                # start here
                if [ ! -d /sys/bus/i2c/devices/1-006f/ ]; then
                        echo mcp7940  0x6f > /sys/bus/i2c/devices/i2c-1/new_device
                fi
                if [ ! -c /dev/rtc0 ]; then
                        mknod /dev/rtc0 c 251 0 2>/dev/null 1>&2
                fi
                # end
                if [ "$VERBOSE" != no ]
                then
                        echo "System time was `date`."
                        echo "Setting the System Clock using the Hardware Clock as reference..."
                fi
```

Figure 6.
Using the 'pca953x' kernel
module it is easy to control
the port expander's GPIO
pins.


```
Elektor GNUBLIN ELDK (Built by Poky 5.0) 5.0 gnublin ttyS0
http://www.gnublin.org

gnublin login: root
root@gnublin:~# modprobe pca953x
root@gnublin:~# echo 98 > /sys/class/gpio/export
root@gnublin:~# echo out > /sys/class/gpio/gpio98/direction
root@gnublin:~# echo 1 > /sys/class/gpio/gpio98/value
root@gnublin:~#
```

Table 2. Values needed when driving the PCA9555 pins using /sys/class/gpio/export	
Pin name	**Value for /sys/class/gpio/export**
IO0_0	98
IO0_1	99
IO0_2	100
IO0_3	101
IO0_4	102
IO0_5	103
IO0_6	104
IO0_7	105
IO1_0	106
IO1_1	107
IO1_2	108
IO1_3	109
IO1_4	110
IO1_5	111
IO1_6	112
IO1_7	113

date 122014342012

which gives the following result:

Thu Dec 20 14:34:00 UTC 2012

To copy the system time to the hardware clock, use the command

hwclock -w

and to copy the hardware clock time back to the system time, use

hwclock -r

which gives a response along the lines of

Thu Dec 20 14:34:17 2012 0.000000 seconds

If the back-up battery is fitted the RTC will keep time while power is not present. The battery holder can be mounted on the top of bottom surface of the board (see the circuit diagram).
In order to ensure that the system time is set from the RTC at start-up extend the file '/etc/init.d/hwclock.sh' as shown in **Figure 5**. Make sure also that the module rtc-mcp7940 is included in /etc/modules.

Port expander
In the seventh article in the Linux board series [9] we looked at how to drive the port expander, using either the I²C tools or C.
The 'pca953x' kernel module makes it as easy to use the port expander GPIOs as it is to use the

hardware GPIOs on the LPC3131 [10]. **Figure 6** shows the commands required, and **Table 2** shows the values that need to be used to address the various pins.
In the next issue we will look at a simple application using the extension board. We use the buttons and LCD to create a menu, which can of course be adapted for your own applications.

(120518)

Internet Links

[1] sauter@embedded-projects.net

[2] www.elektor-magazine.com/120181

[3] www.elektor-magazine.com/120596

[4] www.lcd-module.de/fileadmin/eng/pdf/doma/dog-me.pdf

[5] www.reichelt.de/index.html?ARTICLE=35919

[6] http://ww1.microchip.com/downloads/en/DeviceDoc/22266D.pdf

[7] www.ti.com/lit/ds/symlink/pca9555.pdf

[8] http://gnublin.googlecode.com/git/lpc3131/gnublin_package/src/gnublin-dogm/usr/bin/gnublin-dogm

[9] www.elektor-magazine.com/120518

[10] www.elektor-magazine.com/120146

COMPONENT LIST

Resistors
R1,R2,R5,R6,R7,R10,R11,R12,R13,R14,R15 = 10kΩ
R3,R4 = 2.2kΩ
R8,R9 = 27Ω (optional)

Capacitors
C1,C6–C9 = 2.2µF
C2,C5 = 100nF
C3,C4 = 7-12pF trimmer

Semiconductors
T1 = BSS123
IC1 = PCA9555D, SO-24 case
IC2 = MCP7940, SOIC-8 case

Miscellaneous
X1 = 32.768kHz quartz crystal
K1 = 14-pin boxheader
K2 = 3-pin pinheader, or 22-pin (optional)

K3,K4 = 3-pin pinheader
K5,K6 = 2-pin pinheader
   (optional)
BT1 or BT2 = Battery holder,
   CH25-2032LF
BZ1 = Buzzer
S1,S2,S3 = pushbutton, type
   9314 SMD
LCD1 = 2x16 characters,
   LCD Electronic Assembly
   DOGM162L-A
PCB # 120596

Module/Kit # 120596-91:
comprises SMD-populated
   and tested board. Com-
   ponents LCD1, X1, K1-K4,
   BZ1, BT1 to be mounted.

Arduino on Course (5)
Cookie thief *framed*!

By **David Cuartielles**
(Spain) and
Bobbie Cuartielles
(Sweden)

We found out someone is eating the cookies from the jar in the kitchen! We discussed the issue and decided to set up a trap to reveal the thief. Rest assured, we don't want to hurt anyone, we just want to know and have evidence of the 'crime'.

So here we are, with a clear mission and a whole lot of tools at our disposal. We decided to put on our inventor coats on and make a machine that takes a picture of whoever opens the cookie jar.

Materials
This month the list of materials to replicate this project is pretty straightforward:

- Arduino Uno board
- Computer with Processing IDE and Arduino IDE installed
- Webcam
- USB cable
- Breadboard
- One 1-megohm (1 MΩ) resistor
- Jumper wires
- TinkerKit Module: Ultra Bright White LED + TinkerKit wire

The idea
We are going to create a trap where a home-made toy (pictured above) is constantly guarding our cookie jar. If anyone touches the jar, a camera hidden inside the toy shoots a picture and hopefully photographs the robber's face.

Capturing the image is done with a program written in Processing, which will be taking pictures through the camera inside the toy. We will then activate the camera from an Arduino board that's effectively detecting when someone touches our precious cookie storage. This will be achieved using the capacitive sensing capabilities implemented on all ATMega chips.

The concept sounds simple enough and integrating the parts is also kind of easy. First we will make a Processing sketch though that's effectively capable of triggering the photo shot using Arduino over the computer's serial port.

Installing Processing
Download 'Processing' from its website [1]. If you haven't been following any of my recent *Arduino On Course* installments, you should know that Processing is a cross-platform software development environment dedicated to the creation of Java, Android, and Javascript programs. It is aimed at people learning how to code and make complicated things in uncomplicated ways.

We are going to use a webcam; therefore you should double check whether your camera works inside Processing. If you are running Windows or Mac, please read [2] in case it didn't work out of the box, and if you happen to be a Linux user

(like me) read this other link [3] where it explains which packages you need to install for Processing to take control of your webcam.

From Processing we are going to use the Video and Serial libraries, and both come with the IDE by default. Therefore you don't need to worry about finding and installing extra libraries this time.

Capturing a (webcam) image

The latest version of Processing implements its video library using Gstreamer [4], a suite of software tools capable of accessing various video devices connected to your computer, pipe the streams, implement codecs, etc. You can access the video stream coming from a camera connected to the computer through a file handler. For example, on *nix based machines (like Linux or Mac OSX), all your video devices are listed under the file /dev/videoXX, where XX represents a different number identifying different cameras. For this project, as I am programming from my laptop, /dev/video0 represents the camera that comes embedded with the computer, while /dev/video1 is the camera I have placed inside the toy guard.

The first thing you need to take into account is that the same camera can have different configurations in terms of resolution and image shooting speed. Therefore, you should run the program shown in code **Listing 1**, and check out the different cameras and possible configurations in your computer as they will come out printed in the console inside your Processing IDE. The program processes code to do a survey of cameras available in your computer.

You will be surprised by the long response coming from that program; in my case, listing only the possible configurations for my computer's internal camera, I get a list like the one in **Table 1** (list trimmed).

Next, decide on the camera configuration based on the size you want your sketch to be. In my case, I wanted to have one configured at 640 x 480 pixels, i.e. index number 3 in Table 1. Note that there may be configurations with the same values in the result; you should check which one best suits you.

The camera selection has to be done inside the program's setup — simply modify the conditional statement inside setup to contain the code

Table 1.

index	name	size	fps
[0]	/dev/video0	1280x720	10
[1]	/dev/video0	960x540	10
[2]	/dev/video0	800x448	15
[3]	/dev/video0	640x480	30
...			
[86]	/dev/video0	320x240	15

Listing 1

```
/**
 * Check cameras in Processing
 */

import processing.video.*;

Capture cam;

void setup() {

  String[] cameras = Capture.list();

  if (cameras.length == 0) {
    println("There are no cameras available for capture.");
    exit();
  } else {
    println("Available cameras:");
    for (int i = 0; i < cameras.length; i++) {
      println("["+i+"] "+cameras[i]);
    }
  }
}

void draw() {
}
```

Listing 2

```
  if (cameras.length == 0) {
    println("There are no cameras available for capture.");
    exit();
  } else {
    println("Available cameras:");
    for (int i = 0; i < cameras.length; i++) {
      println("["+i+"] "+cameras[i]);
    }

    // The camera can be initialized directly using an
element
    // from the array returned by list():
    cam = new Capture(this, cameras[3]);
    cam.start();
  }
```

●Projects

shown in **Listing 2**, this will effectively activate the selected camera.

Using the keyboard in Processing
We want to capture an image when someone touches the jar, and it is good to test that functionality beforehand. I would suggest you implement and test something like using the keyboard to take a picture. Just modify Listing 2 with the code in **Listing 3**, and the camera will store a

picture whenever you either press "p" or "P".

Look at the highlighted line:
saveFrame("pic-######.png") — this is the method that saves an image in the sketch folder using the name "pic-######" in PNG format, and with that extension. The string of hash signs ###### represents the picture number. Different pictures will have different numbers.

Note that **saveFrame()** saves the information contained inside the application's frame only — not the rest of the desktop.

Timestamp that picture!
As **saveFrame()** writes a pixel-by-pixel copy of your application into a file, it is possible to add a layer of information on top of the image. In our case, we are interested in knowing *when* the jar was opened and its precious contents thieved! In *CSI Miami* speak: the crime has to be evidenced in terms of time. So let's put a timestamp on the camera picture as shown in **Listing 4**. This will also print a note to STDOUT.

An experimental but successful snapshot is shown in **Figure 1**. The low resolution is obviously due to the webcam, but the detectives will be equally happy.
Next thing is writing some Arduino code to determine whether someone touched the jar, and immediately return a command to the computer.

Detecting touch with Arduino: CapacitiveSensor Lib
Inside archive file 120745-11.zip created for this article you will find a folder named "Arduino" which has a subfolder called "libraries". The archive file may be downloaded for free from [6]. You should copy the contents of that folder inside your Arduino libraries folder. The latter is located inside your Arduino sketchbook folder, usually stored at "My Documents/Arduino" in Windows computers, or "Documents/Arduino" in Mac and Linux.

The CapacitiveSensor library was originally made in 2008 by Paul Badger [5] and it comes with one single but clear example. I recommend you take a look at the Arduino playground page referenced at [5]; there you will find everything you need to know about how capacitive sensors work. Essentially, the library uses two digital pins: one

Listing 4
```
void draw() {
  if (cam.available() == true) {
    cam.read();
  }
  image(cam, 0, 0);
  String timeStamp = String.format("%02d", hour());
  timeStamp += ":" + String.format("%02d", minute());
  timeStamp += ":" + String.format("%02d", second());
  timeStamp += " " + year();
  timeStamp += "/" + String.format("%02d", month());
  timeStamp += "/" + String.format("%02d", day());

  text(timeStamp, 10, height-10); // superimpose text on
image

  // for the keyboard detection to work, you need to have
  // clicked on the application window first (aka focus)
  if(keyPressed) {
    if (key == 'p' || key == 'P') {
      saveFrame("pic-######.png");
      println("capturing Frame at: " + timeStamp); // report
to the console
    }
  }
}
```

Listing 3
```
void draw() {
  if (cam.available() == true) {
    cam.read();
  }
  image(cam, 0, 0);

  // for the keyboard detection to work, you need to have
  // clicked on the application window first (aka focus)
  if(keyPressed) {
    if (key == 'p' || key == 'P') {
      saveFrame("pic-######.png");
    }
  }
}
```

to send a pulse, and one to read that pulse. Both pins are connected through a resistor of a certain value (1 MOhm for us) and leaves the receiving pin 'open' though a wire or piece of conductive metal that will act as an open-ended capacitor. The mere presence of the human body will modify the capacitance value of such a pin and therefore modify the amount of time it takes for the signal at the receiving pin to reach a value that can be understood as logic High. The CapacitiveSensor library measures that time — if it is long enough, most likely a human being is interacting with the jar opening pin and we are in a position to photograph the culprit red handed.

Capacitive sensors are literally everywhere in our digital lives. These devices are so sensitive they can be hidden behind plastics, wood, and any other non-conductive materials. At the same time, extending the coverage of the touch sensors is as easy as adding more metal to them and configuring a couple of passive components (resistors and capacitors).

The circuit we are going to add to Arduino boils down to a single 1-MOhm resistor and a couple of wires. On top of that, considering that our guard might have to shoot a picture at night, we added an ultrabright LED board from TinkerKit to easily illuminate the scene. We made the guard's mouth from translucent acrylic sheet (plexiglass) for the light to shine through. Check the 'schematic' in **Figure 2** for more details on the construction. Back to software, the Arduino code needed for performing all of these operations is fairly simple as you can see in **Listing 5**.

For those of you with some experience with Arduino, notice that I highlighted the lines that belong to the new library in use. You will need to configure the **THRESHOLD** constant to fit your specific circumstances. If you check **Figure 3** you will see we cut an 'antenna' out of aluminum foil to place underneath our cookie jar. The shape and size of that antenna determine the times measured by the library. You will have to configure the value of the **THRESHOLD** constant for the program to operate properly. Follow these steps:

- connect your circuit to the aluminum foil;
- measure the default value, depending on how much foil you have this value should be in the range of 200;
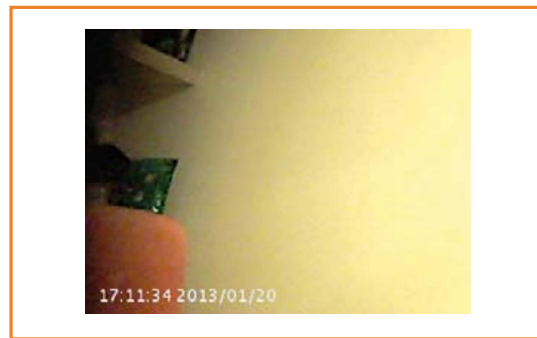
Figure 1.
Detail of captured image with superimposed timestamp.
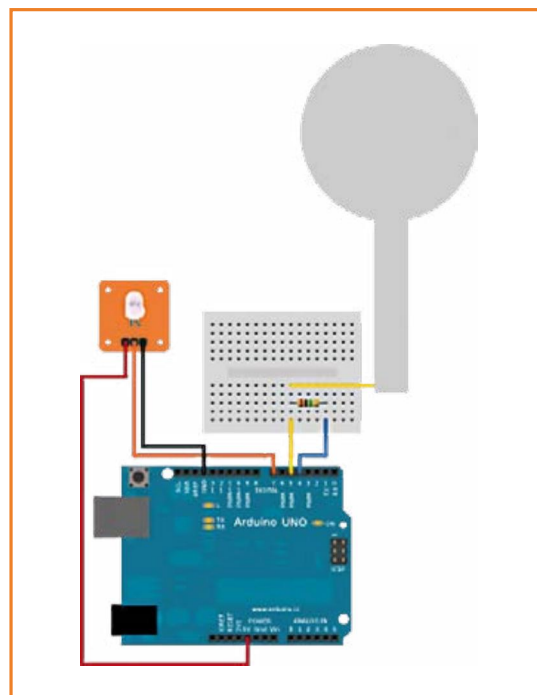
Figure 2.
Arduino connection diagram for the project (made with Fritzing).

- touch the sensitive area and observe the measurement in the Serial Monitor;
- make **THRESHOLD = (MAX_VALUE + DEFAULT_VALUE) / 2** and change it in the code.

When your program detects a value coming from this homebrew sensor to be over the **THRESHOLD**, it will return the character 'p' to the Processing application through the serial port. The only two things missing now are: fixing the Processing program to trigger the camera shot through the serial communication with Arduino, and mounting the trap.

Triggering the camera from Arduino
Basically you need to modify your Processing application to include the serial library, and reading data whenever it arrives through the port as shown in **Listing 6**.

Mounting the trap

Our friend Tien Pham designed a Cyclops-like, monstrous *Cookie Guard* that got cut on a laser cutter at the local FabLab. Tien cleverly used the mechanical properties of our webcam to become the neck of *Cookie Guard*. The camera sensor is hidden behind the guard's only eye, and the ultrabright LED is behind the mouth. There is enough room behind the guard's body to hide the Arduino board and the small breadboard we have

used. Tien's construction is pictured in **Figure 4**. As explained earlier, we made the touch sensor from a piece of aluminum foil. It's shaped like the letter 'i' but with a colossal dot. The dot is the antenna where we put our jar on. The aluminum foil is glued with tape to the wire going to the breadboard and to pin number 5.

Note that a metal jar was used; the project should also work with other materials, but you may have to make your sensor more sensitive by increasing the value of the resistor from 1 MOhm to, say, 10 MOhm.

Closing words

This is a project that came to us as a fun thing to do during a weekend, when we noticed someone had eaten our cookies. We were able to build everything out of small parts left over from other projects. There is a lot of room for improving this design: you could start by soldering all the components on a prototyping shield, looking for a better material to build the sensor, make a bed for the sensor, etc. For now, the result is shown in **Figure 5**.

However, I wanted to show that for building a convincing proof of concept you need very little, even when dealing with digital electronics. If I find more time, I will modify the Processing code to post the pictures taken straight to a webserver, making the information available remotely in an instant.

(120745)

Listing 5

```
#include <CapacitiveSensor.h>

#define THRESHOLD 500 // Note you have to configure this
value

// declare the pins to be used
// 1M resistor between pins 4 & 5
// ultrabright LED on pin 7
CapacitiveSensor cs_4_5 = CapacitiveSensor(4,5);
int ledPin = 7;

void setup() {
   // uncomment to turn off autocalibrate on channel 1
   //cs_4_5.set_CS_AutocaL_Millis(0xFFFFFFFF);

   // configure the serial port
   Serial.begin(9600);

   // configure the pin for the LED
   pinMode(ledPin, OUTPUT);
}

void loop() {
   long sensorReading = cs_4_5.capacitiveSensor(30);

   if (sensorReading > THRESHOLD) {
     // turn on the light
     digitalWrite(ledPin, HIGH);

     // tell the computer to take a picture
     Serial.write('p');

     // wait and then turn the light off
     delay(2000);
     digitalWrite(ledPin, LOW);
   }
   else {
     // uncomment these lines for configuring your system
     // Serial.println(sensorReading);
     // delay(200);
   }
}
```

References

[1] Processing project: http://processing.org

[2] Issues when using webcams in Processing: http://wiki.processing.org/w/Video_Issues

[3] List of packages to get GStreamer for Linux to take webcams inside Processing: http://forum.processing.org/topic/how-is-video-on-linux-handled#25080000001764427

[4] Gstreamer official website: http://gstreamer.freedesktop.org/

[5] Paul Badger's CapacitiveSensor Library for Arduino: http://playground.arduino.cc/Main/CapacitiveSensor

[6] Cooke Guard project software: www.elektor-magazine.com/120745
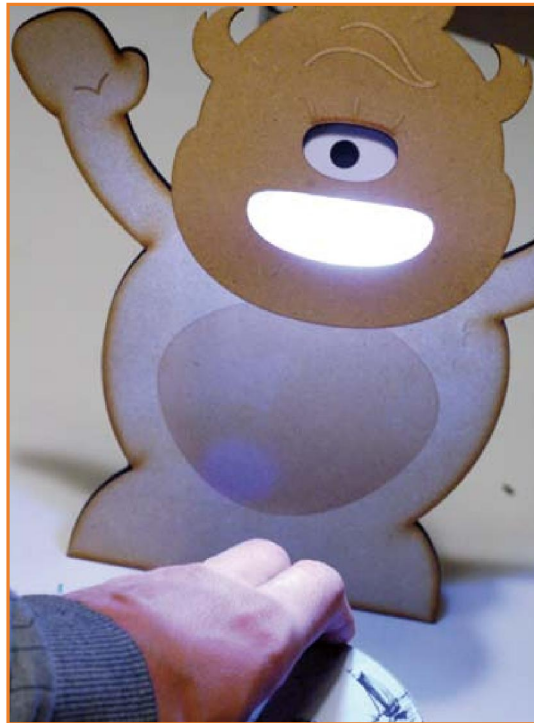
Figure 5.
Framed!! Who's stealing our cookies?

Figure 3. The all-DIY capacitive sensor.

Figure 4. Detail of Cookie Guard's neck construction.

Listing 6

```
import processing.video.*;
import processing.serial.*;

Capture cam;
Serial myPort;   // The serial port

void setup() {
  [...]

  // we don't need the camera at full blast
  frameRate(1);

  // List all the available serial ports
  println(Serial.list());
  // Open the port you are using at the rate you want:
  myPort = new Serial(this, Serial.list()[0], 9600);
}

void draw() {
  [...]

  if (myPort.available() > 0) {
    int inByte = myPort.read();
    if(inByte == 'p') {
      saveFrame("pic-#####.png");
      println("capturing Frame at: " + timeStamp);
    }
  }
}
```

Universal Driver for Power LEDs

Tiny, efficient and practical

By **Ingo Burret**
(Germany)

Practically the first thing you are taught about LEDs is that they need a constant current power source. Power LEDs are no different they just need more current. At higher current levels the driver circuit efficiency is of critical importance if we are not to squander the energy advantages of LEDs. Power LEDs are now more affordable and this project uses a specialist IC to simplify the design of a compact and efficient power LED driver.

Batteries and power supplies, in fact the vast majority of DC power sources provide a more or less constant voltage output. LEDs however require their power source to provide a constant current. For normal indicator LEDs drawing just a few milliamps the usual solution is to put a resistor in series with the LED. For higher power LEDs this simple solution is no longer practical; the power dissipated in a series resistor would defeat the object of modern-day high efficiency, power LEDs. The ideal solution would be an efficient switch mode power supply providing a constant current rather than constant voltage.

Switched constant current

Over the last few years developments in LED technology has led to their use in high power lighting applications. This in turn has fuelled the development of specialist LED switch regulator ICs. One example of this type of IC is the TS19377 produced by Taiwan Semiconductor and is used here. A special feature of this design is that the internal reference voltage for the error amplifier which is compared against the voltage drop across the load shunt is particularly small. The IC in fact requires a voltage drop of just 0.25 V i.e. ten times smaller than the standard 2.5 V required by most other switch regulators designs for constant voltage operation. With a 1 amp output current, losses in the shunt amount to just 250 mW.
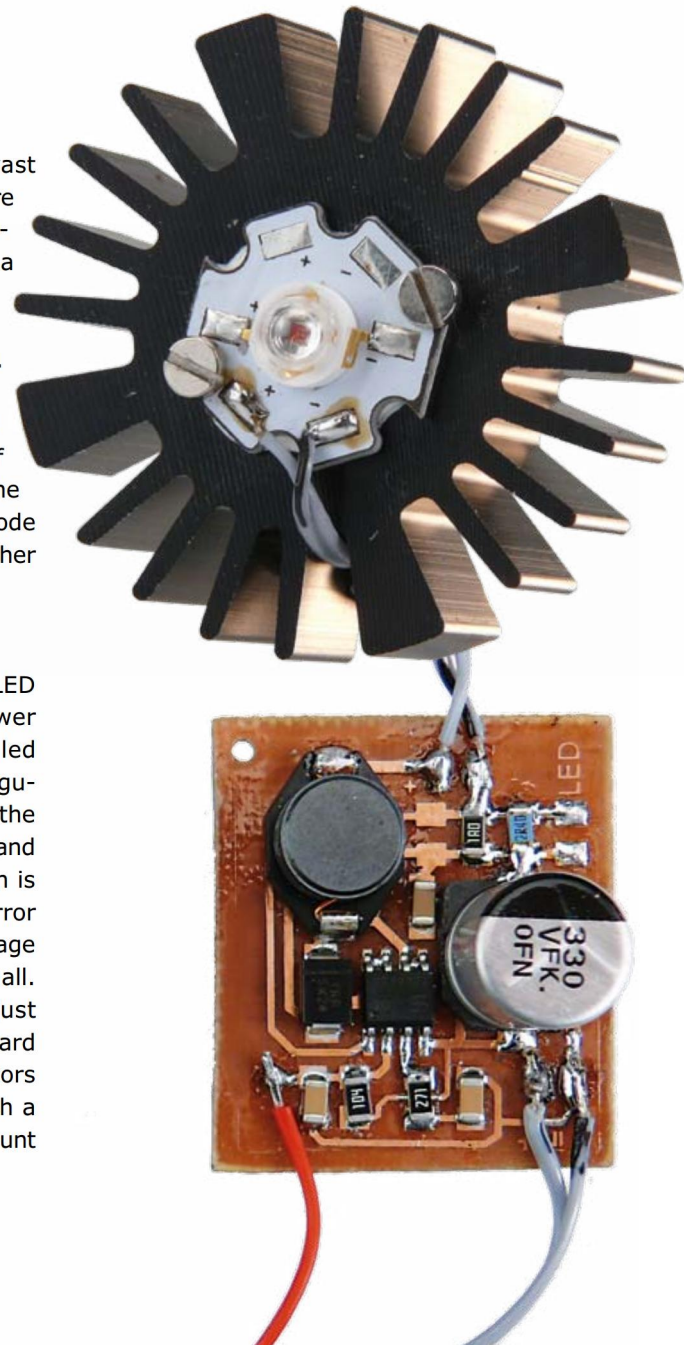
Despite the IC being so small (it comes in an SMD outline package) it features a fast on-board P channel power MOSFET capable of switching currents up to 2 A. The regulator uses a 330 kHz clock frequency which allows a relatively small external filter coil. Internal protection is also quite sophisticated; it includes not only output short circuit protection but also a 'soft' thermal protection which reduces output power if the chip starts to get too hot. When you see an IC described as 'specialist' you know it's really another way of saying 'expensive' and 'difficult to find' but not in this case. Reichelt.de, a distributor in Germany has them priced cheaply.

The circuit details

The circuit shown in **Figure 1** deviates only slightly from that suggested in the TS19377 data sheet [1]. Note that there is no reservoir capacitor at the output. The LED is driven by a pulsed DC signal. At 330 kHz it is not possible to detect any flicker of the LED light. This high pulse frequency means that the inductor L1 need only have an inductance of 68 µH which makes it quite compact and reduces losses in the coil impedance.

At the power input to the circuit there is a reservoir capacitor C1 and decoupling capacitor C2. The high frequency of operation means that a value of just 330 µF is sufficient for this application. With a single power LED connected to the output the chip can be powered from a supply voltage in the range of 3.6 to 23 V. The power supply does not need to be smoothed; in fact it can be just an unfiltered but rectified pulsing DC voltage. The input capacitor C1 functions as a reservoir capacitor. It is important to ensure that the peak input voltage on C1 does not exceed the maximum allowed in the spec. This means that the transformer should have a secondary winding rated at 15 V max.

The input voltage ground is connected to pin 1 of JP1. At the output the LED is connected with its anode on pin 2 of JP2. The only pin available at JP4 can be taken low to turn the regulator off. In the case when C4 is not fitted, a PWM waveform can be applied to pin 2 so that the signal's mark/space ratio controls the LED brightness. If neither of these features are required then pin 2 can be strapped to VCC by replacing R6 with a 0 Ω resistor. Capacitor C4 will then serve no purpose and can be omitted.

Minimum V_{in} in Volts given the number of LEDs and colour								
LED count	1	2	3	4	5	6	7	8
V_{in}, LEDs red	3.6	6	8	11	13	16	18	21
V_{in}, not red	4	8	12	16	19			

Figure 1.
The circuit diagram.
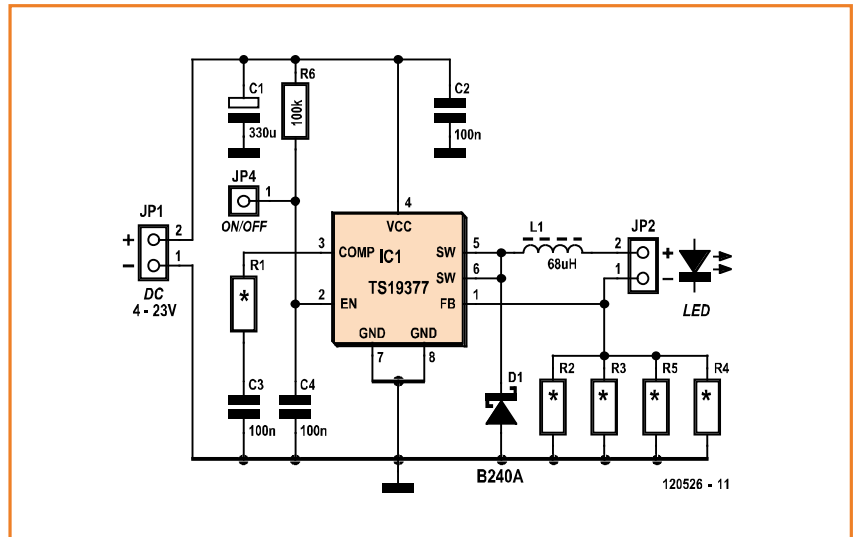
Technical Data
Universal power LED driver

- Operation with 1, 3 and 4 W LEDs
- Drives up to 8 LEDs in series
- Input voltage range 3.6 to 23 V
- Maximum output current 1 A
- Efficiency with 1 W LEDs:
  - V_{in} = 5 V: 81.6 %
  - V_{in} = 12 V: 74.7 %
  - V_{in} = 16 V: 69.2 %

Construction and configuration

Component layout on the PCB is a little tight (see **Figure 2**), but there should not be too many problems if you already have a little experience of soldering SMD components. Make sure that the polarity of C1 and D1 are correctly observed before soldering. Ensure that the IC is not rotated by 180°, IC pin 1 must correspond to pin 1 on the PCB. The PCB layout files are available for free download from the Elektor web for this project [2].

The power LED driver can be used to supply LEDs which require a drive current between 0.35 to 1 A. A 1-watt LED will typically draw 350 mA whereas most 3 W versions require around 750 mA and

It is never a good idea to wire LEDs in parallel directly

1 A will be needed for a 4-watt LED. The low shunt voltage means that we can get away with standard 1206 SMD outline resistors (R2 to R5 wired in parallel) for the shunt. In the simplest case with a drive current of 1 A and a voltage drop of 0.25 V, four 1 Ω resistors are fitted in parallel. For 750 mA fit 1 x 1 Ω and 3 x 1.5 Ω. For 350 mA only two resistors are necessary; a 1.2 Ω in parallel with 1.8 Ω to give a current of 347 mA. For a load current of 350 mA resistor R1 should be 220 Ω but for the two higher current configurations this should be 470 Ω.

From the technical data it can be seen that operation with a 1-watt LED shows the efficiency decreasing as the voltage difference between the input and output increases. This effect is quite normal for switch-mode regulators and indicates that to improve efficiency it is a good idea to connect several LEDs in series which will have the effect of increasing the output voltage of the regulator. This improves not only the regulator's efficiency but also minimizes the influence of the voltage drop across the shunt on the calculations and reduces the effect of the threshold voltage of the rectifier diodes. There is of course a limitation here: The input voltage must be greater than the sum of the output voltage plus the voltage drop across the shunt.

From this it can be seen that with three LEDs in series the input voltage to the regulator must be at least 12 V. White LEDs typically have a forward voltage of 3.7 V. Three in series plus the shunt drop gives 3 x 3.7 + 0.25, an input of 12 V is just 0.65 V above the output. It follows that it is not possible to connect more than five white LEDs in series with this driver otherwise the input required would exceed the rated maximum input voltage. Be aware that some LED modules contain several LEDs wired in series. Six or even eight red LEDs can be driven in series when each has a forward voltage below 2.5 V. A table shows minimum input voltage required given the number of series LEDs and their colour.

Finally it should be noted that it is never a good idea to wire LEDs in parallel directly; no two LEDs will have precisely the same characteristics, when two are wired in parallel the LED with the lowest forward conduction voltage will hog most of the current while its partner will take hardly any.

(120526)

Internet Links

[1] www.taiwansemi.com/home/en/products/product_info.php?partid=TS19377CS

[2] www.elektor-magazine.com/120526

COMPONENT LIST

Values for 350 mA

Resistors
(SMD shape 1206)
R1 = 220Ω *
R2 = 1.2Ω *
R2 = 1.8Ω *
R4,R5 = not fitted *
R6 = 100kΩ

Capacitors
C1 = 330µF 35V, SMD electrolytic (e.g. Panasonic EEEFK1V331AP)

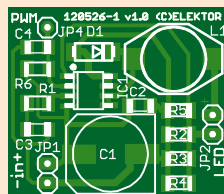Figure 2. Thanks to SMD components the finished universal LED driver is very compact.

C2,C3,C4 = 100nF 50V ceramic (X7R), 1206 *

Inductors
L1 = 68µH 1.6 A, 0.24Ω, SMD (e.g. Würth 74456168 or Fastron PISM-680M-04)

Semiconductors
D1 = B240-13-F, DO-214AA (SMB)
IC1 = TS19377CS, SO8 (e.g. Reichelt.de)

Miscellaneous
JP1,JP2 = 2-pin pinheader, 0.1'' pitch
JP4 = pin
PCB # 120526-1

* see text

Thermo Book

Measure temperature and relative humidity

By **Willem Tak** (Netherlands)

Another thermometer? Not at all, this design does not only have increased functionality but also has an original enclosure! On the display of this thermo book you can choose to indicate either the temperature or the relative humidity, switching between them at a fixed interval or 'manually' by clapping your hands from a distance.

It is very likely that every true electronics hobbyist will have built a clock somewhere along the line, but thermometers too will certainly score very high in the top-10 of electronics projects. Why then add yet another one to this long list? Let's say the technology of this thermometer / relative humidity meter is not challenging, but the attempt here is to make the styling a little bit different. The thermometer will look like a book and therefore can easily find a spot on a bookshelf. Experience has already shown that such an odd book will frequently elicit a smile. And if the thing also displays the temperature and relative humidity accurately, then funny is united with functional.

Hardware
The most important component in this circuit (see **Figure 1**) is the sensor. The author is addicted to the use of the Sensirion SHT75, and this is despite its high price. This sensor is very accurate and is easy to use. By the way: the slightly cheaper SHT15 is compatible, but does come in a different package (module on an SMD circuit board). A PIC18F14K50 has been used as the controller.

This runs from its own internal clock at 8 MHz. Transistors T3 through T5 supply the pulses for the common anodes (CA) of the three 7-segment LED displays. The segment data for the LED displays comes, via resistors, directly from a port on the PIC. The value of potentiometer P1 is read continuously by the controller and is used to adjust the brightness of the LED-displays. The SHT75 is controlled by a couple of lines from the PIC via a separate I²C extender, IC2. Note that this IC is not absolutely necessary (and can optionally be omitted), but in practice the circuit with the extender has turned out to be very robust and reliable.

Because the SHT75 can measure, in addition to temperature, also the relative humidity, this value is also stored on the controller. There are two methods to display the measured values. In one case the display will alternately display the temperature and the relative humidity. The choice was made to display the temperature for twice as long as the relative humidity. The second option is to display the relative humidity only 'on demand'. This demand is implemented with
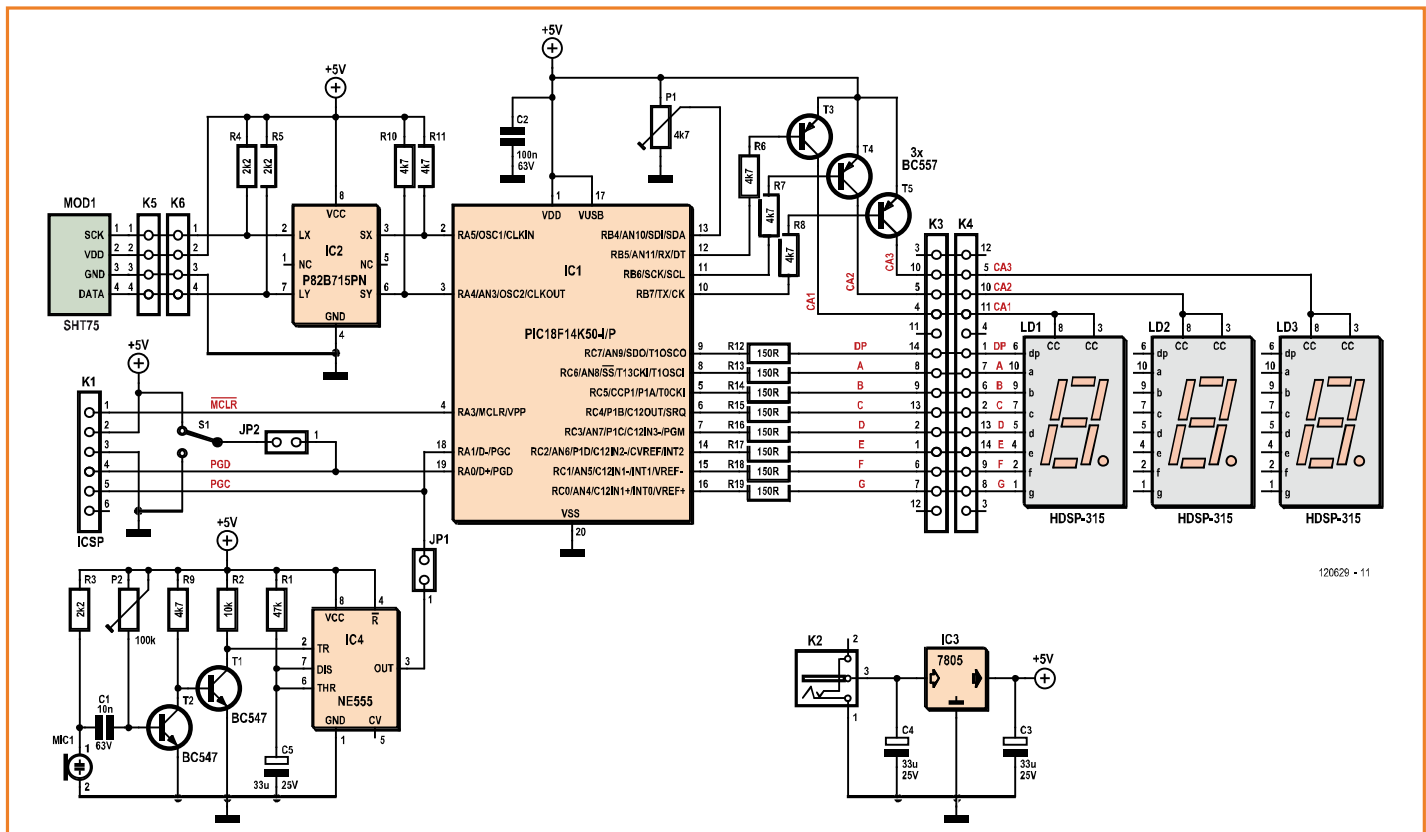
a clap-switch. This switch is built up around T1 and T2. The sensitivity can be adjusted with P2. When a 'clap' has been detected, IC4 (NE555) will generate a pulse, which forces the PIC to display the relative humidity for the next 4 s. Which if these two display methods is active can be selected with switch S1.

An ICSP connector (K1) is present to allow the microcontroller to be programmed in the board. This can be used to connect, for example, a PICkit programmer. When the programming header is in use, jumpers JP1 and JP2 have to be open, in all other cases JP1 and JP2 must be fitted with a jumper. You can, of course, order a pre-programmed controller (120629-41) from Elektor, but if you would like to program the microcontroller yourself you can download the source and hex code free from our website [1].

There is no reset-switch. The controller resets from the power supply voltage and then starts the initialisation process. During this process the first measurement results are acquired. While this is in progress the display will shows ascending dashes.

The power supply is provided by a line adapter which is allowed to have an output from 7 to 12 V. Voltage regulator IC3 turns this into a clean 5 V. The current consumption is less than 100 mA.

Software

The biggest time-consuming part of the software is the control of the LED-display. There is a routine, LDD cycle, which for about 0.5 s supplies the data for the display. The On and Off times can be different, this is used to adjust the brightness of the display. The analogue value of potentiometer P1 is read using the built-in A/D converter and determines the intensity.

In the main loop the very first task is the start-command, subsequently 2 display cycles are processed and then the sensor values are read. After that, the sensor values (obtained in Sensirion format) need to be converted to display values. No lookup tables are used here for the T (temperature) value, but instead we use a divide-by-100 method with subsequent determination of the remainder. By contrast, the determination of the H value (relative humidity) does use a lookup table. There is an idle time between the

COMPONENT LIST

Resistors
R1 = 47kΩ
R2 = 10kΩ
R3,R4,R5 = 2.2kΩ
R6–R11 = 4.7kΩ
R12–R19 = 150Ω
P1 = 4.7kΩ preset, horizontal
P2 = 100kΩ preset, horizontal

Capacitors
C1 = 10nF, MKT, 5mm pitch
C2 = 100nF, MKT, 5mm pitch
C3,C4,C5 = 33μF 25V radial, 2.5mm
   pitch

Semiconductors
IC1 = PIC18F14K50-I/P, DIP-20, pro-
   grammed, Elektor # 120629-41
IC2 = P82B715PN, DIP-8 (NXP)
IC3 = MC7805CTG, TO-220 (On
   Semiconductor)
IC4 = NE555P, DIP-8 (TI)
LD1,LD2,LD3 = 7-segment display,
   HDSP-315L (Farnell/Newark #
   1241274)
T1,T2 = BC547C
T3,T4,T5 = BC557B

Miscellaneous
JP1,JP2 = 2-pin pinheader, 0.1'' pitch,
   with jumper
K1 = 6-pin pinheader, 0.1'' pitch
K2 = AC adaptor socket, PCB mount,
   2.1mm pitch (Digikey # CP-102A-ND)
K4,K5 = 14-pin (2x7) boxheader, 0.1''
   pitch (optional, see text)
K5,K6 = 4-pin pinheader, 0.1'' pitch
   (optioneel, zie tekst)
MOD1 = Sensirion temperature and
   humidity sensor type SHT75 (Farnell/
   Newark # 1590514)
S1 = single pole double throw switch
   (e.g. Multicomp 1MS1T1B5M1QE, Far-
   nell/Newark # 9473378)
MIC1 = electret microphone, 9.7mm di-
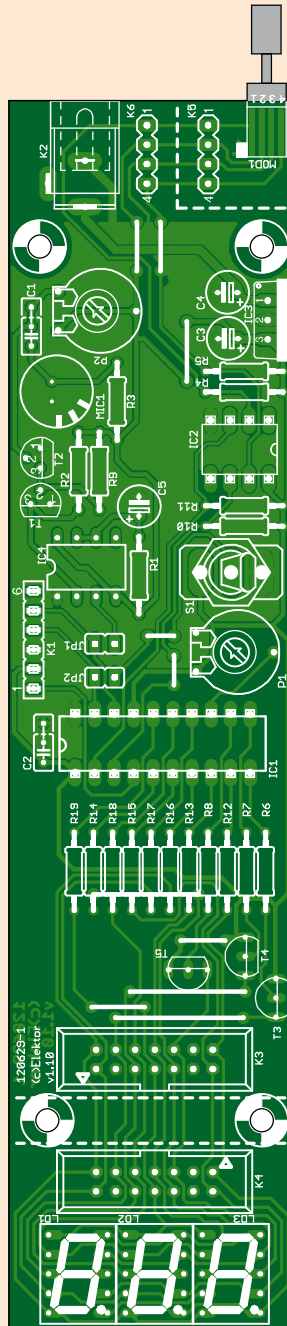   ameter (Farnell/Newark # 1736563)
PCB # 120629-1, ref. [1]

Figure 2.
The circuit board can be split into two parts so that the display can be mounted in a different place. The sensor part can also be separated.

slide switch S1. Note that this is only checked at the end of a cycle (about 3 s), so is not effective immediately.

In the clap mode, the input that registers the clap is checked at the end of every LDD cycle. When this has been detected, a loop is started that will display the H value for about 4 s. In the meantime a new T value is being prepared so that when the display returns, a new value of T is immediately available. Note that there are also H measurements even when only T is displayed. This prevents the situation, for example when there have been no claps for many hours, that the first value displayed is from hours ago.

Printed circuit board

A circuit board has been designed for this circuit, the shape of which is long and narrow (**Figure 2**, available via [1]). Depending on the construction of the 'book', the display part can optionally be separated from the controller part. These are then connected together with a short length of 14-way ribbon cable and two IDC connectors between connectors K3 and K4. If this is not necessary then there is also no need to fit connectors K3 and K4.

The section of the circuit board that has the connection to the sensor can also be separated from the main circuit board, so that you can let the sensor protrude a little from the top of the book (otherwise it will not be able to measure the correct temperature and humidity). In this case the sensor board is connected with a piece of 4-way ribbon cable with the main circuit board.

There is not much to be said about the assembly of the circuit board. Only leaded components have been used, which are easily inserted and soldered. JP1 and JP2 are both fitted with a jumper during normal use of the circuit.

Implementation

The 'book' itself can be made in many different ways. The author used five small sheets of Perspex (Plexiglass) in his prototype. When these are glued together, it is easy to print a dust jacket with the desired 'title' of the book, or you can salvage a cover from an old book with the correct dimensions (also see the photos for this project on the Elektor.LABS website [2]). For the prototype that we built in the lab we simply cannibalized the hard cover of an old book, in which we glued two brackets to which the circuit board was attached (see photo).

start and read-out of the chip. This is necessary to prevent self-heating of the chip.
There are two separate main loops for the different modes of operation (automatically alternate or clap mode). The selection is made with

It is necessary to make a rectangular hole in the cover and the front panel, if there is one, so that the LED displays fit neatly. As mentioned earlier, it is best if the sensor is mounted in such a way that it protrudes a little above the book. On the back add an input for the power supply wires. Also keep in mind that S1 (for selecting the display mode) has to remain accessible either by allowing one of the side panels to hinge or by placing the switch on the back.
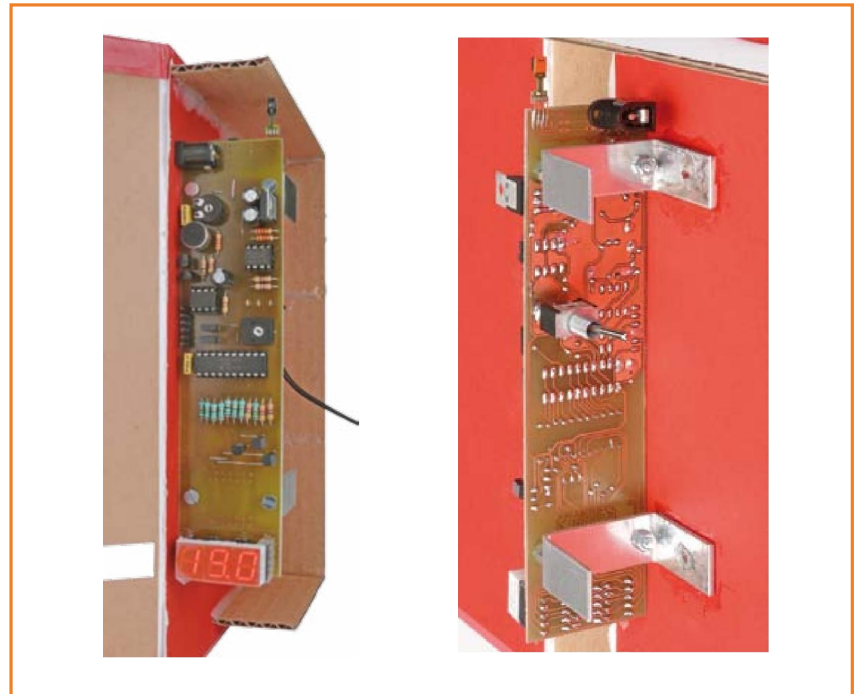
This circuit succeeds or fails with the implementation, so put some effort into it and make it into something nice. So much greater will be the effect that the thermo book has on visitors.

(120629)

Internet Links

[1] www.elektor-magazine.com/120629

[2] www.elektor-labs.com/project/thermo-book.12410.html

DIP Accelerometer Module

By
Hannes Nordmann
(Germany)

These days there is an enormous number of chips on the market crying out to be experimented with, but only available in packages which do not exactly lend themselves to home soldering. In particular, devices with contacts only on their underside, such as those in LGA packages, push the limits of what can be done without professional equipment.

The project described here solves this problem in the case of the Freescale MMA7455 and MMA7456 acceleration sensors. It takes the form of a small adapter board which allows these devices, which come in LGA packages, to be used on ordinary prototyping board and communicated with over their I²C bus. The adapter also allows the device, which requires a 3.3 V supply, to be powered from a standard 5 V source: the board includes the necessary regulator circuit and a level shifter for the I²C bus signals.

Headers K1 and K2 provide the mechanical interface between the device and the prototyping board. The 5 V supply arrives on the board via pin 3 of K2. Zener diode D1, in combination with resistor R1, forms the regulator circuit that provides power at 3.3 V to IC1. Note that the current draw of the circuit is very low, and at such low currents the voltage across the Zener diode will be a bit more than its nominal value. C1, C2, C3 and C4 are decoupling capacitors. The regulated 3 V supply is made available on pin 4 of K1 so that it can be used to power other circuitry if needed. The field-effect transistors convert the 5 V levels on the external I²C bus pins SCL and SDA to the 3.3 V levels needed by the IC. The I²C bus signals are available on pins of header K2. Pin 7 of IC1 (I2C_EN) is held at 3.3 V, which configures the device to operate in I²C mode. The interrupt outputs of the device (on pins 8 and 9) control the gates of field-effect transistors T3 and T4 respectively. The consequence of this is that each interrupt signal appears on header K1 in
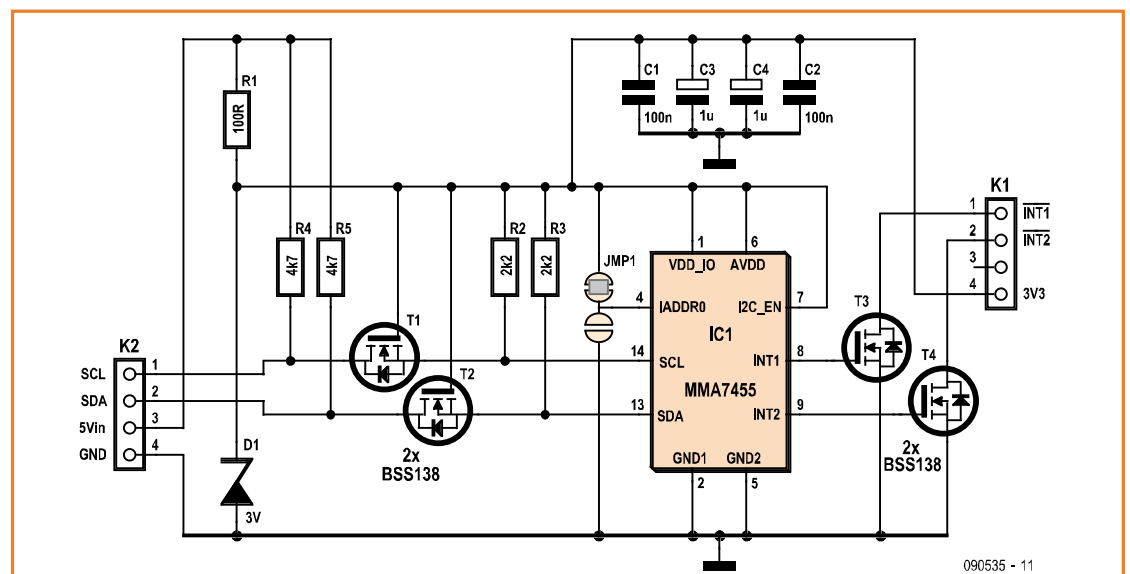
090535 - 11

inverted form: the open-drain outputs give extra flexibility, in particular allowing connection to both 3.3 V and 5 V microcontrollers. Jumper JMP1 (which is implemented as a group of solder pads on the board) determines the least-significant bit of the device's I²C bus address (see [1]), which can come in handy to avoid collisions when several I²C peripherals appear on the same bus.

The pin spacing of header strips K1 and K2 is chosen to match that found on conventional prototyping boards. The distance between K1 and K2 is approximately 15 mm, which is the standard spacing for a (0.3 inch) DIP socket.

A ready built module is available from Elektor. A 3D model (constructed for Google SketchUp using the tool EagleUp [2]) as well as the PCB artwork can be downloaded from the Elektor website [3].

(090535)

Internet Links

[1] www.freescale.com/files/sensors/doc/data_sheet/
    MMA7455L.pdf

[2] http://eagleup.wordpress.com

[3] www.elektor-magazine.com/090535

COMPONENT LIST

Resistors
Size 0603
R1 = 100Ω
R2,R3 = 2.2kΩ
R4,R5 = 4.7kΩ

Capacitors
C1,C2 = 100nF (16V, 0603)
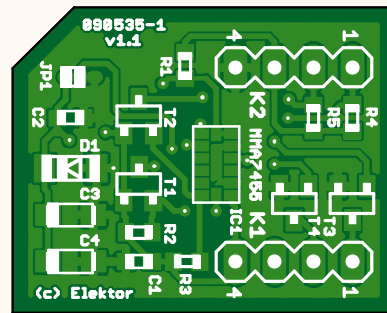C3,C4 = 1µF 16V, tantalum,
  SMC-A)

Semiconductors
IC1 = MMA7455LR1 (LGA 14)
T1–T4 = BSS138 (SOT-23)
D1 = BZV55-C3V0, 115
  (SOC-80C)
Miscellaneous
K1,K2 = 4-pin pinheader

200%

Gadgeteer

Rapid prototyping the Microsoft way

By Clemens Valens
(Elektor.LABS)

In early 2011 Microsoft launched a rapid prototyping platform named .NET Gadgeteer based on its .NET Micro Framework (NETMF). Like many other rapid prototyping systems, the platform is intended to facilitate the creation of electronic systems by non-specialists, basically by keeping the complexity of hardware well out of sight. Unlike other platforms, Microsoft did not (and still does not) provide any hardware, only detailed specifications were published together with the software.

It took a while before the first compatible mainboard became available, because porting the framework to custom MCU hardware is not trivial. Now, two years later, a few mainboards exist and quite a number of add-on boards can be found too, all from third-party hardware manufacturers.

What is Gadgeteer?

A Gadgeteer system consists of a mainboard to which users connect a number of extension boards, called modules. All connections are through standard 10-pin (2x5) shrouded headers, the so-called sockets, and ribbon cables. Although all connectors appear identical, they are not. Labels identify the type of module you can connect to a socket (see **Table 1**). The type is indicated by a letter, and a socket may support several types. As an example, a type-A socket provides three analogue inputs, one general-purpose input/output (GPIO), and three n/c pins. The P-type socket has three PWM outputs, two GPIO pins, and two unconnected pins. For all sockets pin 1 is 3.3 V, pin 2 is 5 V and pin 10 is GND. Since socket types A and P are laid out in a complimentary way, they can be combined into one AP socket. Two GPIO-only socket types are defined too, X (3 GPIO pins) and Y (7 GPIO pins), which can be combined with most other types. Hence most sockets will be labelled 'X' and/or 'Y' and a combination of one or more other letters. The Z socket is a so-called Manufacturer Specific (MS) socket

whose seven available pins are unspecified. The modules also have one or more labelled sockets, and you should only connect sockets of the same type. Some module sockets may additionally be labelled with a star (*) which shows that the particular socket can be used for daisy chaining using the DaisyLink protocol. This is an I²C based protocol combined with a single wire neighbour bus for initialisation and interrupt purposes.

The Gadgeteer hardware specification also allows for so-called shields; basically daughter boards piggy-backed on another board. This is intended for adding Gadgeteer capabilities to existing hardware.

Microsoft did not specify a processor for the Gadgeteer platform, because it is supposed to be hardware independent. The mainboards now on the market seem to be ARM based but other MCUs may follow. The differences at the hardware level are handled by driver libraries (DLLs) provided by the hardware manufacturer. All in all, Gadgeteer seems very much inspired by the PC architecture, where the PC running some form of Windows is replaced by an MCU board running NETMF, and where the DaisyLink replaces the USB bus.

The user creates a system by connecting the modules he/she needs to the mainboard. A graphic tool then allows easy configuration of the

system simply by drawing the modules and the connections. It can also wire the modules for you (see **Figure 1**). (It would have been really cool if you could use a webcam or a camera to make the tool automatically recognize your system.) A click on a button makes the tool produce a software framework for the system that includes the software objects for all the functions exported by the modules (**Figure 2**). Of course, the user has to install the drivers first. Once the framework is ready, the user can start programming and this is where the simplicity stops.

Common Language Runtime?

At the time of writing, the Gadgeteer mainboard is a reasonably powerful microcontroller board with a 32-bit processor and at least 390 KB of program memory and 64 KB of RAM (I found these numbers in the documentation of the NETMF Porting Kit). All this memory is necessary to run (amongst others) the Common Language Runtime (CLR), a piece of software that executes the user program. A smaller CLR — appropriately named TinyCLR — exists that occupies less program memory, so with luck you may get away
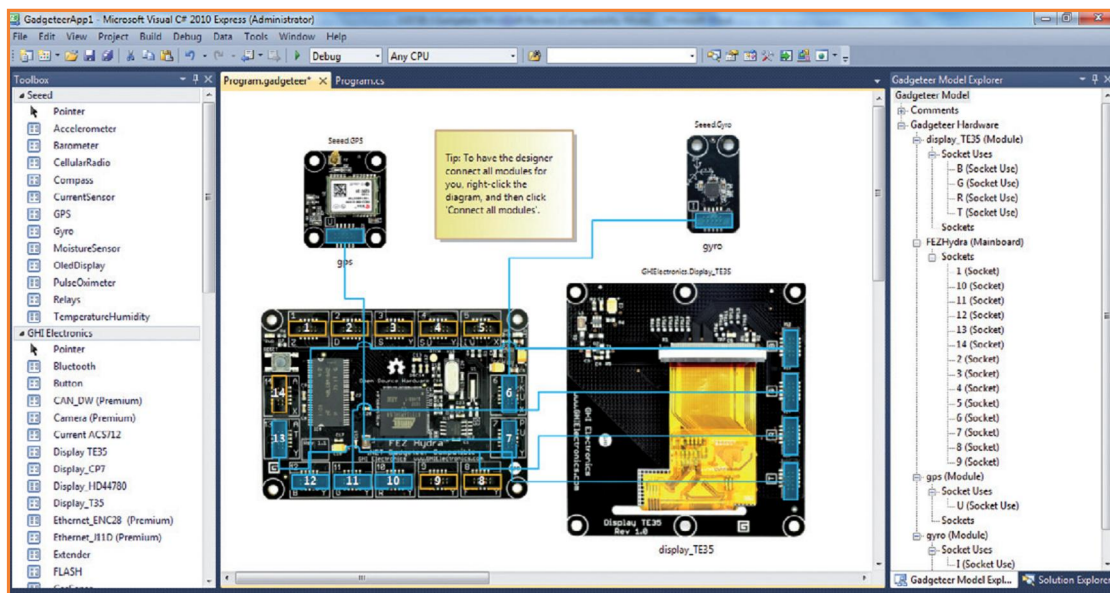

Figure 1.
After installing all the necessary software, SDKs, and expanding the right windows, this is the system I was able to create. I let the software connect the modules for me. The toolbox on the left contains all the modules I installed for the occasion; on the right you can see the sockets available on the hardware.

Figure 2.
This is the program that got created for my system. Namespace, partial class... you need some programming experience to understand this. When you add a command (like I did here for the GPS module), the IDE proposes a list of options to complete your command. Again, you need some programming experience to understand this.
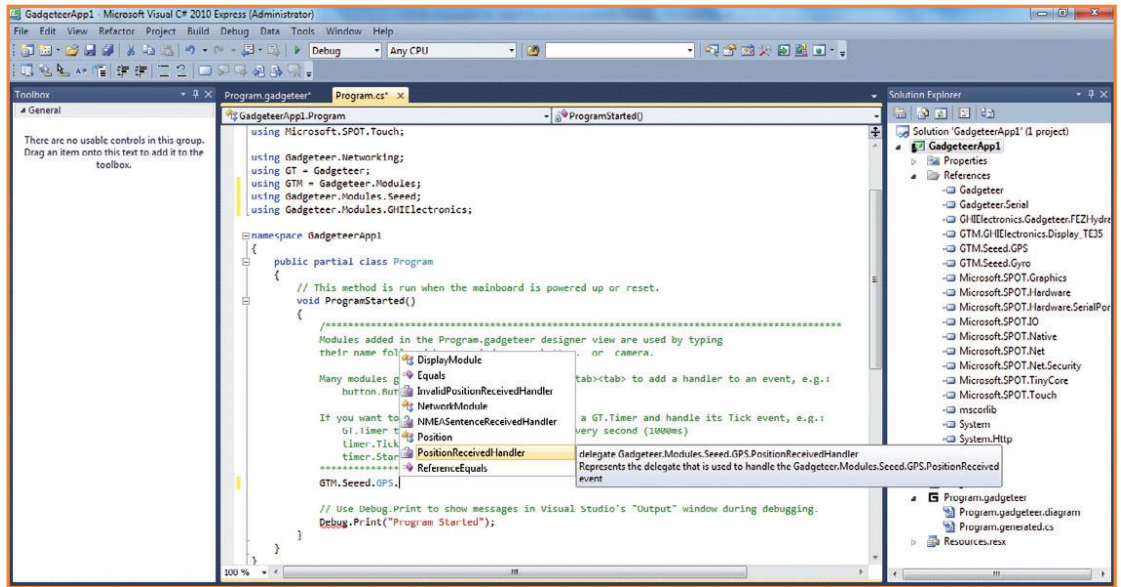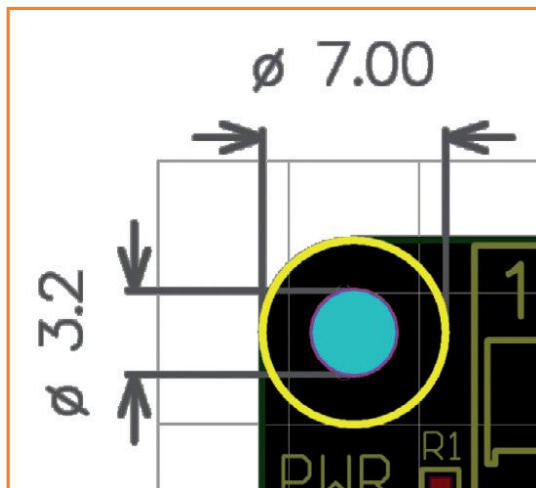
with only 256 KB. However, space is also needed for the user application and a typical mainboard would therefore probably have 512 KB of program memory and more than 64 KB of RAM.

Using the previously mentioned Porting Kit (PK) you can adapt and sculpt the CLR for your custom hardware. This is not an exercise for the fainthearted even though many examples are provided. If you really want you can also port the CLR to a not supported MCU and you may even be (among) the first to do so. Currently the PK supports ARM, Analog Devices' Blackfin and Renesas SH2 architectures.

The CLR looks a lot like an operating system, but in reality it is a kind of command interpreter that compiles the user application on the fly (just-in-

Figure 3.
Microsoft wrote down detailed specifications for the Gadgeteer hardware. As this illustration shows, the board corners have to be rounded and mounting holes are well-defined.

time compilation). It provides functionality like multi-threading, timers, memory and exception management, etc. that the application may use, allowing for compact programs. If a program makes use of this functionality it is said to be "managed" by the CLR. Programs that on the other hand do everything themselves are "unmanaged". Microsoft claims a processing speed of about 550 managed method calls per MHz of MCU clock frequency. ('Method' is programmer's lingo for 'function'.) Unfortunately, it is difficult to translate this definition to something easier to visualize like maximum pin-toggle frequency or similar.

For the moment the NETMF CLR only understands programs written in C# (pronounced as "see sharp", the source files have the extension 'cs'). C# is an object oriented (OO) and event driven language and as such using it is not much different from coding in another OO language like C++ or Java. People without OO programming skills may find this approach rather intimidating. The tools to develop C# code are available for free, just download and install Microsoft's Visual C# 2010 Express (I don't know if the 2012 version is supported). You are required to also install the .NET SDK and the Gadgeteer core and builder templates, and for every piece of hardware that you acquire you have to install the drivers.

Writing the application for the user system – the device — starts easy enough, but we are far from the simple Arduino setup/loop paradigm (see **inset**). The Gadgeteer website puts it like this:

Table 1. Gadgeteer socket types. Legend:

GPIO   A general-purpose digital input/output pin, operating at 3.3 volts.
(G)   In addition to another function, a pin that is also usable as a GPIO.
(OPT)   A socket type that is optionally supported by a mainboard or a module.
[UN]   Modules must not connect to this pin if using this socket type.
[MS]   A manufacturer-specific pin.
!   Interrupt-capable and software pull-up capable GPIO.

Type	Letter	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8	Pin 9	Pin 10
3 GPIO	X	+3.3V	+5V	GPIO!	GPIO	GPIO	[UN]	[UN]	[UN]	[UN]	GND
7 GPIO	Y	+3.3V	+5V	GPIO!	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO	GND
Analog In	A	+3.3V	+5V	AIN (G!)	AIN (G)	AIN	GPIO	[UN]	[UN]	[UN]	GND
CAN	C	+3.3V	+5V	GPIO!	TD (G)	RD (G)	GPIO	[UN]	[UN]	[UN]	GND
USB Device	D	+3.3V	+5V	GPIO!	D-	D+	GPIO	GPIO	[UN]	[UN]	GND
Ethernet	E	+3.3V	+5V	[UN]	LED1 (OPT)	LED2 (OPT)	TX D-	TX D+	RX D-	RX D+	GND
SD Card	F	+3.3V	+5V	GPIO!	DAT0	DAT1	CMD	DAT2	DAT3	CLK	GND
USB Host	H	+3.3V	+5V	GPIO!	D-	D+	[UN]	[UN]	[UN]	[UN]	GND
I2C	I	+3.3V	+5V	GPIO!	[UN]	[UN]	GPIO	[UN]	SDA	SCL	GND
UART+ Handshaking	K	+3.3V	+5V	GPIO!	TX (G)	RX (G)	RTS	CTS	[UN]	[UN]	GND
Analog Out	O	+3.3V	+5V	GPIO!	GPIO	AOUT	[UN]	[UN]	[UN]	[UN]	GND
PWM	P	+3.3V	+5V	GPIO!	[UN]	[UN]	GPIO	PWM (G)	PWM (G)	PWM	GND
SPI	S	+3.3V	+5V	GPIO!	GPIO	GPIO	CS	MOSI	MISO	SCK	GND
Touch	T	+3.3V	+5V	[UN]	YU	XL	YD	XR	[UN]	[UN]	GND
UART	U	+3.3V	+5V	GPIO!	TX (G)	RX (G)	GPIO	[UN]	[UN]	[UN]	GND
LCD 1	R	+3.3V	+5V	LCD R0	LCD R1	LCD R2	LCD R3	LCD R4	LCD VSYNC	LCD HSYNC	GND
LCD 2	G	+3.3V	+5V	LCD G0	LCD G1	LCD G2	LCD G3	LCD G4	LCD G5	BACK-LIGHT	GND
LCD 3	B	+3.3V	+5V	LCD B0	LCD B1	LCD B2	LCD B3	LCD B4	LCD EN	LCD CLK	GND
Manufacturer Specific	Z	+3.3V	+5V	[MS]	[MS]	[MS]	[MS]	[MS]	[MS]	[MS]	GND
DaisyLink	*	+3.3V	+5V	GPIO!	GPIO	GPIO	[MS]	[MS]	[MS]	[MS]	GND

"*.NET Gadgeteer uses the .NET Micro Framework to make writing code for your device as easy as writing a desktop, Web or Windows Phone application.*" (When was the last time you wrote a desktop, Web or Windows Phone application?) Right from the start the user is confronted with a rather intimidating default program to which he/she has to add his/her own initialisation code and event handlers. This is all very well for the trained software engineer, but the average hardware developer may feel a bit lost in the beginning. However, when you spend some time examining how things work here, you should be able to get a simple application deployed on your device quickly. That's right. If you want to be taken seriously as a Gadgeteer you must "deploy your managed solution on your device" instead of "reflashing your board" as all other hackers do.

Conclusion

Microsoft tries to create a NETMF-based ecosystem ranging from PCs and smartphones to MCU boards, abstracting away the hardware, so that software engineers feel at ease on any system. This is a valid and logical goal, and other
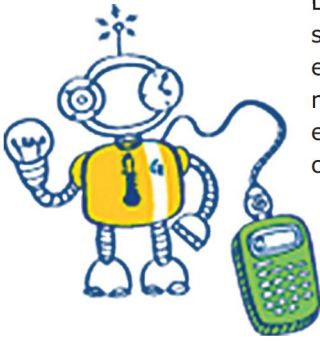
companies have similar objectives. Gadgeteer fits in this philosophy as a NETMF-based rapid prototyping platform intended for software engineers. However, if you are a software engineer, you can also go for Linux and Embedded Linux, or Android. You would not have the nice sockets, although they are easily added, but the ecosystem is way larger, and instead of a few mainboards you can choose from hundreds or even thousands of mainboards and some pretty cool smartphones.

Being a Microsoft technology Gadgeteer will probably attract a reasonable audience in the coming years and it will be interesting to see how well it combines with Windows Phones. However, for the moment it seems a bit of a slow starter if we can trust the NETMF and Gadgeteer forums activity. At the time of writing this article there are rather few subjects, and the most recent post dated back to almost one month. The latest news was from July 2012.

(110738)

Internet Links

[1] Gadgeteer main page:
www.netmf.com/gadgeteer

[2] .Net-MF for Electronics Engineers:
www.elektor.com/120033

The graphic elements that illustrate this article were sourced from the Gadgeteer website.

Arduino versus Gadgeteer

Even though both Arduino and Gadgeteer are open source rapid prototyping platforms targeted at non-specialists, there are several significant differences between the two. First of all the target audience. Arduino is aimed at people not necessarily skilled in electronics and/or programming, whereas Gadgeteer targets programmers without hardware skills. The Arduino team has gone through a lot of effort to simplify programming as much as possible where Gadgeteer developers have to use a typical programmer's IDE. On the hardware side of things there is a difference too. The Arduino platform is designed in such a way that every interested amateur can build his or her own mainboard thanks to freely available reference circuit- and PCB designs. Gadgeteer on the other hand only specifies what the mainboard should look like — what's on the board is up to the manufacturer — and that it should be able to run NETMF.

Another interesting difference is in the programming paradigm. Although Arduino doesn't brag about it, it uses Object Oriented programming techniques like Gadgeteer. Where they differ however is in the single-threaded polling loop versus the multi-threaded event-driven approach. Arduino programs (called *sketches*) execute an endless loop from which the peripherals are controlled. Of course it is possible to use interrupts to signal and handle events, but most users will not use them. Gadgeteer programs (called *solutions*) are event-driven, there is no loop; the main program (thread) may be sleeping while events are handled by interrupt handlers. This difference makes Arduino a good choice for simple controlling applications without complicated user interaction. Gadgeteer on the other hand is noteworthy for more complex, menu-driven devices. As a matter of fact, Arduino and Gadgeteer are not so much competing platforms as they are complementary platforms.

A bright future or doomed from the start?

The .NET Micro Framework (NETMF) has its roots in Microsoft's Smart Personal Object Technology (SPOT, introduced in 2003), developed to personalize consumerware electronics and other everyday devices. References to SPOT show up in NETMF programs. The SPOT technology used MSN Direct network services, an FM radio-based digital service which allowed SPOT devices like wristwatches, desktop clocks, GPS navigation units, and even small appliances such as coffee makers to receive information from MSN services. The information, made available through paid channels, included weather, horoscopes, stocks, news, sports results and calendar notifications. Short messages from Windows Live Messenger could also be received. Superseded by other technologies, the MSN Direct network services were shut down on January 1, 2012.

NETMF is also used by Windows SideShow devices. Windows SideShow, coupled to the Windows Sidebar (Microsoft Gadgets, do we see a connection to Gadgeteer here?), is a technology that enables Windows PCs to drive a variety of auxiliary display devices connected to the main PC providing access to information and media even when the PC is (mostly) turned off. Unfortunately, on July 10, 2012, Microsoft urged Windows Vista and 7 users to disable the Windows Sidebar and gadgets because insecure gadgets could be used to run malicious code on their computer... [*Source: Wikipedia*]

Handlebar Heating
For motorbikes and scooters

By **Ingo Burret**
(Germany)
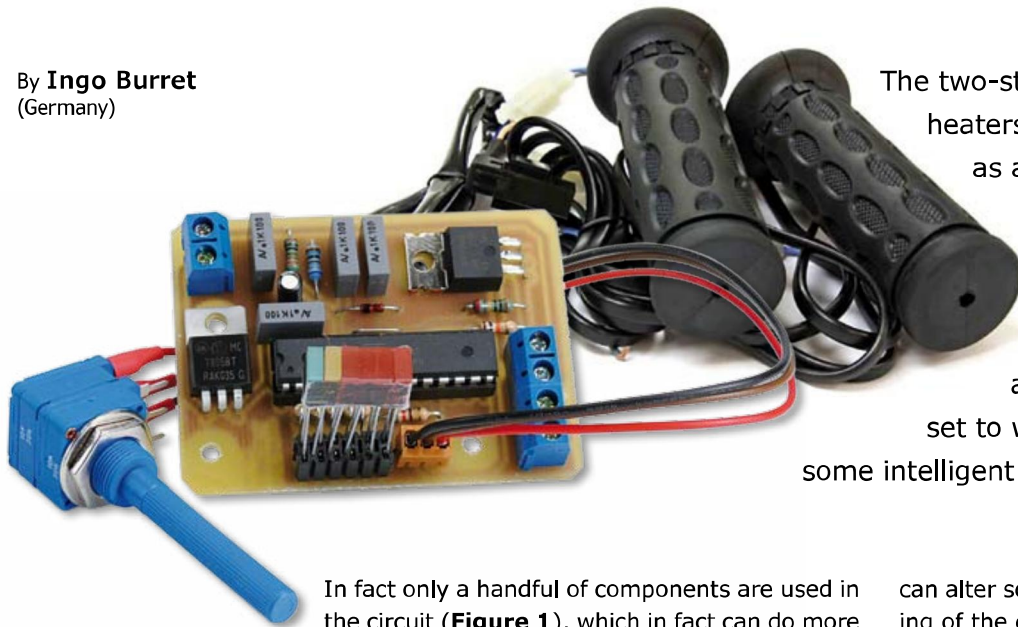
The two-stage control for the handlebar heaters on my motorbike struck me as a bit basic. As a keen recruit to the world of microcontrollers (I'm still finding my way but I find it brilliant to be able to achieve so much with a small a number of components), I set to work to solve the problem with some intelligent electronics.

In fact only a handful of components are used in the circuit (**Figure 1**), which in fact can do more than just warming up handlebars. This primary function is handled by a PWM signal with a frequency of 1.8 kHz that appears at port OC1A of the ATmega8 controller and drives an IRFZ34N MOSFET. This transistor is fully capable of handling the wattage of the heated grips found commonly on sale (between 15 and 20 W per grip). The PWM signal regulates the current flowing from the 12 V battery through the warmed grips to ground. The on-off duty cycle and hence the heating capability are set by potentiometer P1. The setting of this pot is taken to input ADC0 of the analogue-to-digital converter, where it is digitised and displayed using the four LEDs connected to outputs PD2 to PD5. The software also arranges that the heating operates at full blast for a certain time (about 30 s) when you switch on or start the machine. During this initial heating period the LED on PD1 is lit. The sixth LED on PD0 is simply a 'power on' indicator.

The software of the ATmega8 also monitors the voltage divider R9/R10 and the A/D converter ADC1 to sense the voltage at the battery terminals. Normally the battery is not charged properly when the bike is idling, so the PWM signal is switched off when the battery volts fall below 11.6 V. The software recognises when the voltage rises again and reactivates the heating. You can alter some of these parameters in the heading of the controller program listing.

The circuit is built on a very compact PC board (**Figure 2**). All components are mounted in 'through hole' fashion, which simplifies construction greatly. A socket is provided for the controller, making it easy to remove again for reprogramming if necessary. You can obtain the mini-PCB and the controller from Elektor, where programmers can also find the 'C' source code and the hex code [1].

As motorbikes live outdoors mostly, we need to protect the electronics against water spray. For this we recommend using a case of appropriate Ingress Protection (IP) rating, such as the Hammond RP1025C (rated IP65). The second figure of this code denotes its water resistance on a scale from 0 to 8 (the first relates to resistance to impact from foreign bodies) [2]. A '5' is fine for our purpose; it won't be adequate for the bottom of the ocean but it will at least be fully splashproof. The housing mentioned has a transparent lid, so the (rectangular) LEDs can be soldered directly on top of the PCB. The PCB needs to be fastened without piercing the case. Drilling cannot be avoided altogether though, as we need two or three entry points for cables. The box mentioned has room for three cable glands, which you can find at home improvement and
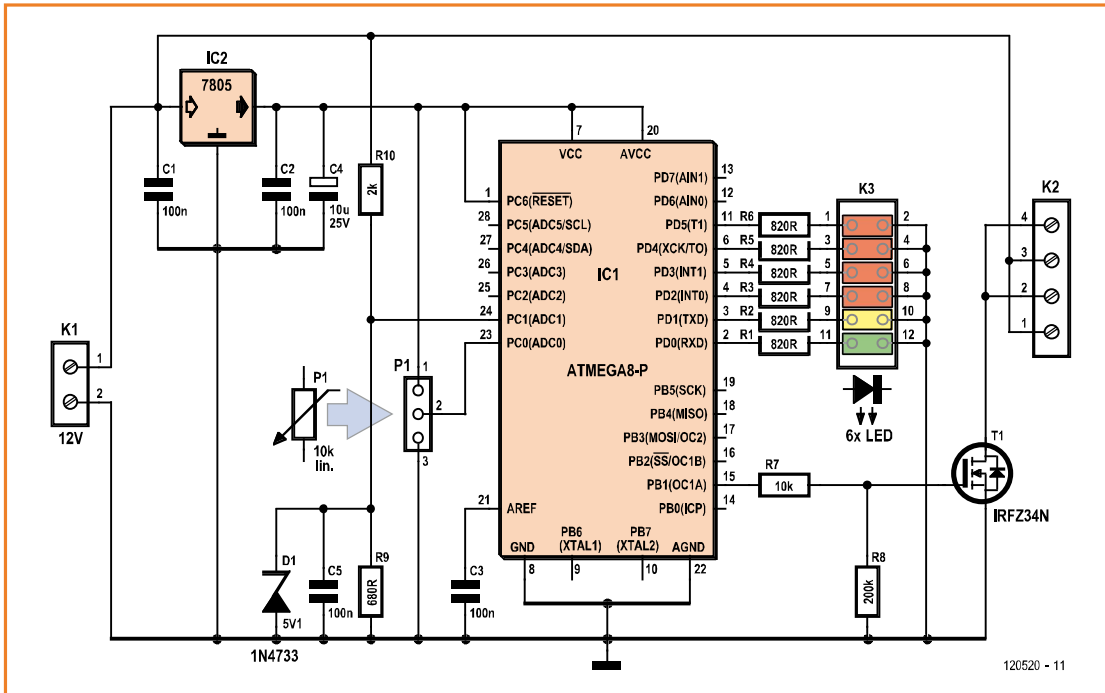
Figure 1.
A controller can warm your fingers.

electrical stores (the smallest ones have a half-inch or 12 mm thread). All you have to be sure of is that the cable inlet is properly sealed (using waterproof reducers if necessary) to maintain the IP rating. For the pot and its spindle you can use special fixtures or spindle bushings and smear these thoroughly with Vaseline. Of course this is only one way of keeping handlebars warm.

(120520)

Internet Links

[1] www.elektor-magazine.com/120520

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

COMPONENT LIST

Resistors
R1–R6 = 820Ω
R7 = 10kΩ
R8 = 200kΩ 1%
R9 = 680Ω
R10 = 2kΩ 1%
P1= 10kΩ potentiometer, linear

Capacitors
C1,C2,C3,C5 = 100nF 100V
C4 = 10µF 25V

Semiconductors
IC1 = ATmega8-16PU, programmed, Elektor #
   120520-41)
IC2 = 7805
T1 = IRFZ34N
D1 = 1N4733A (5.1V zener diode)
LED1–LED6 = LED, 3mm, green, rectangular

Miscellaneous
28-pin DIL socket
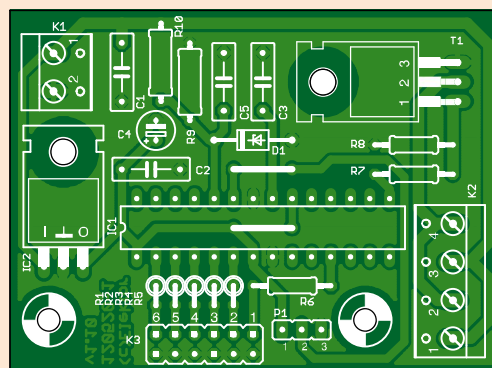
K1 = 2-way PCB terminal block, 5mm pitch
K2 = 4- way PCB terminal block, 5mm pitch
K3 = 12-pin (2x6) pinheader
PCB # 120520-1

* see text

Figure 2.
Everything fits on one compact PCB.

Frontline breaking news

By Clemens Valens
(Elektor .Labs)

Elektor.Labs, the heart of Elektor, is gaining momentum and we are very happy about that. It is not easy to make people change their habits, which is why we spend a lot of effort on promoting our new setting. The goal is to open up our techno dungeon where all the really interesting stuff resides and we would like you to become a part of.
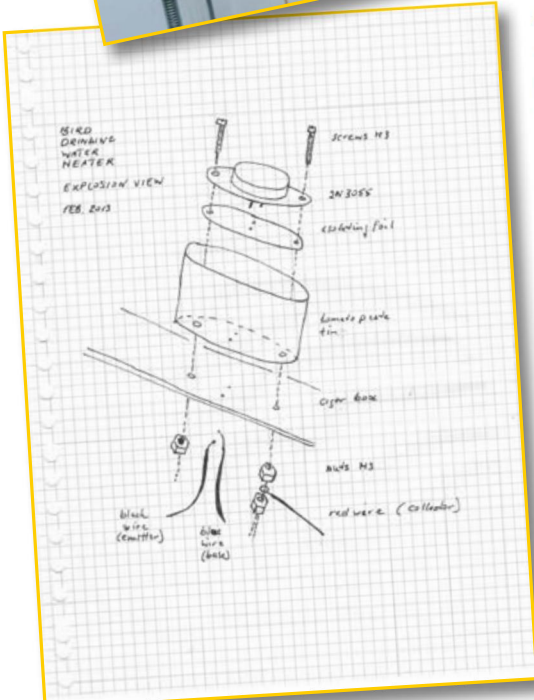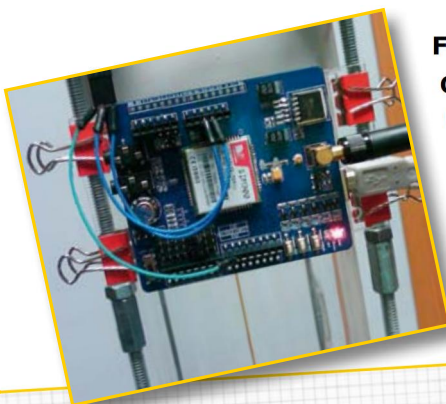
Frequently Asked Questions

Q *When I post a project on the Elektor.Labs website will you publish it in the magazine without telling me?*

A No, we will not. When we feel that a project is interesting enough to be published in the printed magazine, we will contact you. You are the author and we will not publish your work without a contract. That's why it is important that you check the email address that you used for registering every once in a while. If we cannot contact you, we cannot publish your work.

Q *But, to my (pleasant) surprise, I saw a photograph of my project in last month's edition. That was nice, but how come?*

A We are entitled to use the material published on the Elektor.Labs website for promotional purposes. You want your project to be successful, right? We want that too, so we make noise to attract people. Every week we highlight a project in Elektor.Post and every month a few projects are selected for a short presentation in Elektor.Magazine.

Q *I noticed that some projects have a 6-digit number in the title. What does that mean?*

A You probably mean the Elektor article number. When a project has been attributed an article number it is marked for publication in Elektor.Magazine or Elektor.Post. This does not mean that it will be published, but the chances are high that it will happen. Usually the authors of projects with such a number have already been contacted by an Elektor editor. There is no point in adding such a number yourself, because it will not be in our internal database and we will ignore it.

Q *How can I boost my project's popularity?*

A Be clear and informative. Use a self-explanatory title; upload a good-quality photograph of your project. Write in English. While any language is allowed and Google Translate is pretty impressive, projects written in English simply are the

most popular. Furthermore, do not be afraid to publish schematics or source code. Like you, this is what triggers fellow engineers best. We all love to have a look at schematics even if we don't use 99.9% of what we see. Also, update your project regularly. Every time you update your project — but also when someone adds or modifies a comment or a contribution — your project is moved back to the top of the list and is visible on the home page.

elektor⊖labs

Q *Now that you mentioned them, what exactly is the difference between a contribution and a comment?*
A A contribution is a discussion subject supposed to add something to the project, highlight a problem or clarify a point. A comment is a continuation of a discussion and as such can only be added to a contribution. This way a matrix-like structure is created with a vertical list of discussion topics that can each develop horizontally. The goal is to keep related information grouped together instead of being spread out over a long list of comments.

Q *I am receiving automatic emails telling me that my project has been updated, but I didn't do anything?*
A That's correct. The manager (admin) of a project and the project members all receive emails generated by the system every time the project was updated or when a comment or a contribution was added or modified. It is not uncommon for projects to be updated by an Elektor editor to correct a typo for example or to move it to the top of the list. In this case the system will also generate an automatic email.

**Q** *I no longer want to be manager of my project. How can I bail out?*
A Only the Elektor.Labs administrator can bail you out. You can contact him by sending an email to labs@elektor.com.

Q *I did a fabulous project over the weekend that I would like to share, but I am not an Elektor member and I do not have write access on Elektor.Labs. Does this mean that I cannot publish in Elektor?*
A Anyone with a good project can be published by Elektor, in the magazine and/or on the website. A special application form for non-members is available on the Elektor.Labs homepage. Just click the *Not-a-member?* button, send in the form and we will evaluate your proposal. If we think it is good, you will be granted (limited-time) Green access to Elektor.Labs to do your project. And if you are really good, you can even win a Gold membership and receive Elektor magazine for free! That's right, at Elektor experts get paid to publish.

Q *I have a question that you did not answer here. How can I contact you?*
A Send an email to labs@elektor.com.

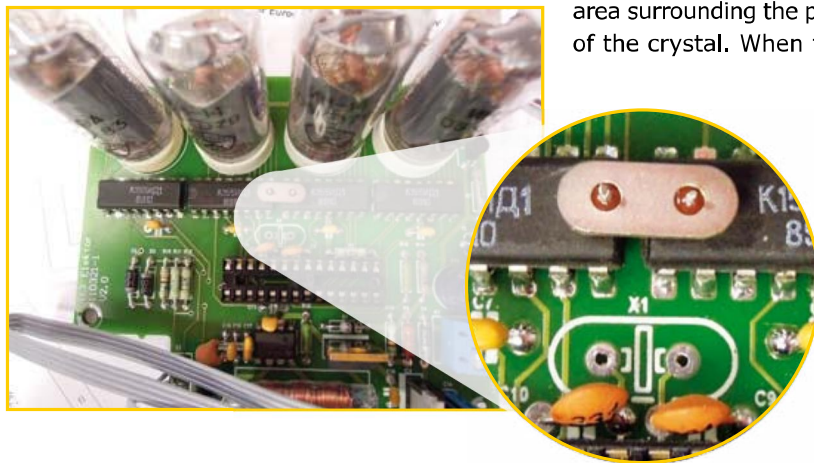(130089)

www.elektor-labs.com
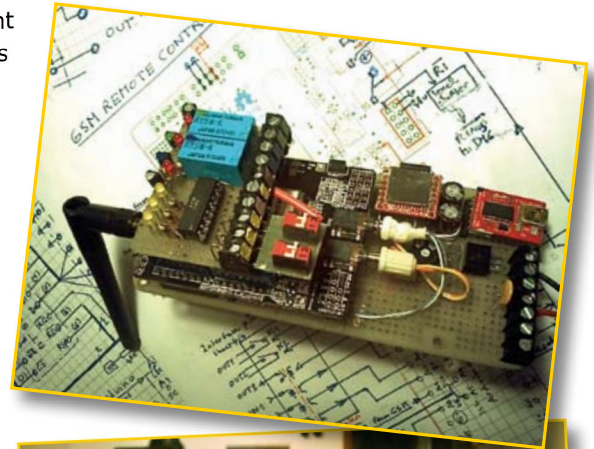
XTAL Killer

By **Thijs Beckers**
(Elektor Editorial)

Assembly of some components can be quite tricky, even if they're not SMD. For instance, mounting an ordinary quartz crystal on a professionally manufactured PCB can still cause some headaches if done incorrectly.

'Now what could possibly go wrong', you might think. Have a look at the photograph (showing part of our Nixie Thermometer / Hygrometer, published June 2012 [1]). The quartz crystal has been desoldered to show the issue: the solder pad size exceeds the isolation area surrounding the pins of the crystal. When the crystal is mounted very close to the PCB surface the metal casing could be shorting the pads, preventing the crystal from oscillating.

The solution for this misconstruction is as simple as it is overlooked. Just insert an insulating sleeve between the crystal and the PCB, or, even simpler, don't mount the crystal flat against the PCB, but a little above it with a little distance between the case and the PCB — a millimeter or so should suffice.

So be warned! Next time your circuit isn't working and the clk signal isn't running, don't immediately assume the crystal is dead. Check for pin shorts first — and in case a microcontroller is involved, after double checking your fuse settings and configuration bits are correct of course. (In this case of a malfunctioning circuit however, the culprit turned out to be a bad resonance capacitor. Very unusual. But there's a first time for everything…)

(120750)

Internet Link:

[1] www.elektor.com/110321

What's cooking?
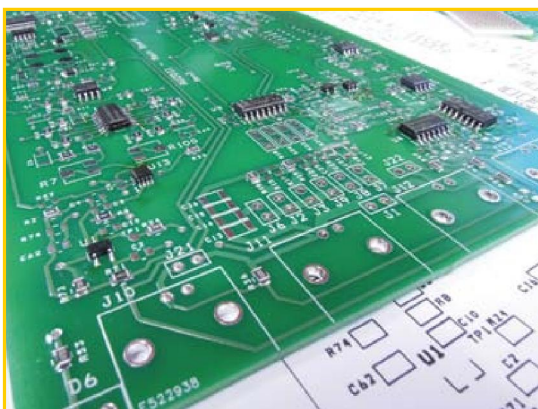
As you have probably noticed, elsewhere in this edition we describe a very nice LCR meter project. At this point in the series we elaborate on the display extension board intended for stand alone use, the meter's capabilities, and we spend a few words on the software side of this comprehensive project. Next in line will be the hardware, so keep an eye on our next issue! In the mean time at the Labs Jan Visser has been busy assembling the most recent prototype PCB (see photograph) for a final test.

As you can see in this nice picture, there are a lot of components to be soldered and almost all of them are SMD. Our assembly takes place in a well facilitated lab environment, but not everyone interested in this extraordinary LCR project will have (access to) the tools and skills to finish this SMD assembly successfully. That is why we cooperate with Eurocircuits on offering this project as a ready made module.

The author and Elektor Labs have put in a lot of effort to deliver this extraordinary instrument that doesn't even shy from comparison with a similar device of professional quality (it stood firm when compared to a Hameg HM8118 Programmable LCR Bridge), so rest assured this device will surely make an excellent addition to your collection of lab tools.

(130092)

LDO replacement THT

One of our lab interns, Philip Jaschewski, got inspired after reading our article on an LDO replacement (November 2012 edition, [1]) and took a shot at developing his very own version of this circuit.

He started off with the following ideas. Being a student, cutting costs was sure to be a major point of attention. Straight off Philip redesigned the PCB to be a single layer version, allowing him to do the etching at home. He also decidedly veered off to the use of THT — through-hole technology. Furthermore, the design had to be compact as well as pin compatible with the TO-220 version of the ubiquitous 78xx family.

Philip allowed one exception to the 'THT rule', which also necessitated the design of an adapter PCB. In order to keep the design as compact as possible, for the regulator IC he picked a type TPS62150 3-17 V 1 A 3 MHz step-down converter IC from Texas Instruments — technically the same as the one used in the original circuit, but in a 16QFN package with a lead pitch of 0.5 mm.

The schematics and PCB layouts of his attempt to reach proof of concept are printed here. Besides paying close attention to the design of a suitable layout for the adapter PCB, special care had to be taken regarding the layout of the "main board": a short feedback loop as well as a short C_{out} loop proved mandatory.

The components used in this design were all 'standard' through-hole technology, with the obvious exception of the regulator IC, and the capacitor used in the output filter, which must be a low ESR type.

A quick price calculation showed he was on the right track, totalling below 7 dollars. But then came the most important part: practical testing. And that's where his design grinded to a halt very quickly. The circuit didn't even come close to generating the desired 5 V. Not by a far shot. With a little assistance of Raymond, the designer of our LDO replacement, the PCB layout could be improved and the circuit was able to supply 5 volts, but only at a measly 80 mA or so.

Besides this disappointing performance, regulation and output voltage ripple were anything but stellar. Sadly scope images of the output voltage are not available, but suffice it to say that this take on a THT version of our LDO displacement has failed.

Luckily for Philip this didn't adversely affect his internship. Internships are for learning how to deal with real problems in the 'real' world, so Philip got his fair share...

Anyone who'd like to have a shot at perfecting young Philip's idea can download his design files (drawn in Cadsoft Eagle v6) from our website [2]. If you want to discuss this project, please visit our .Labs-website [3].
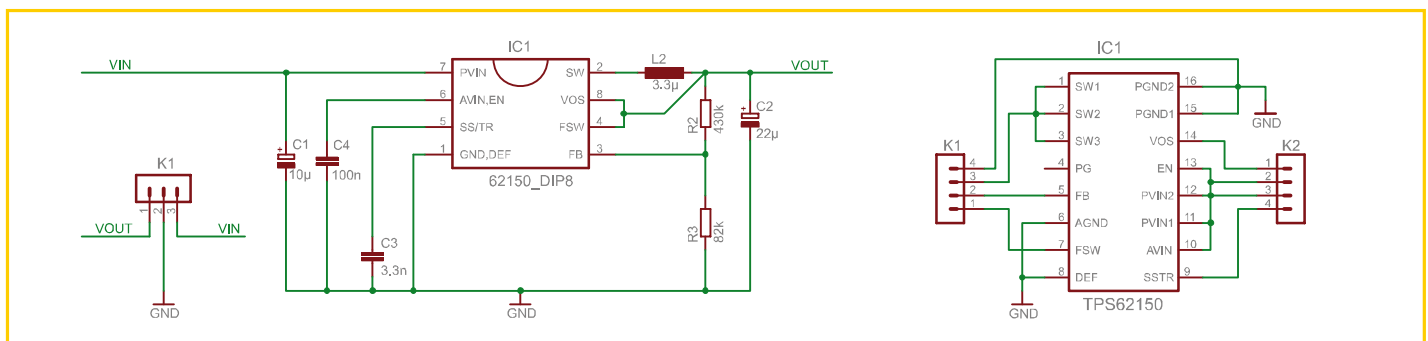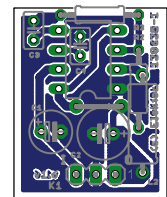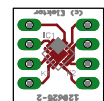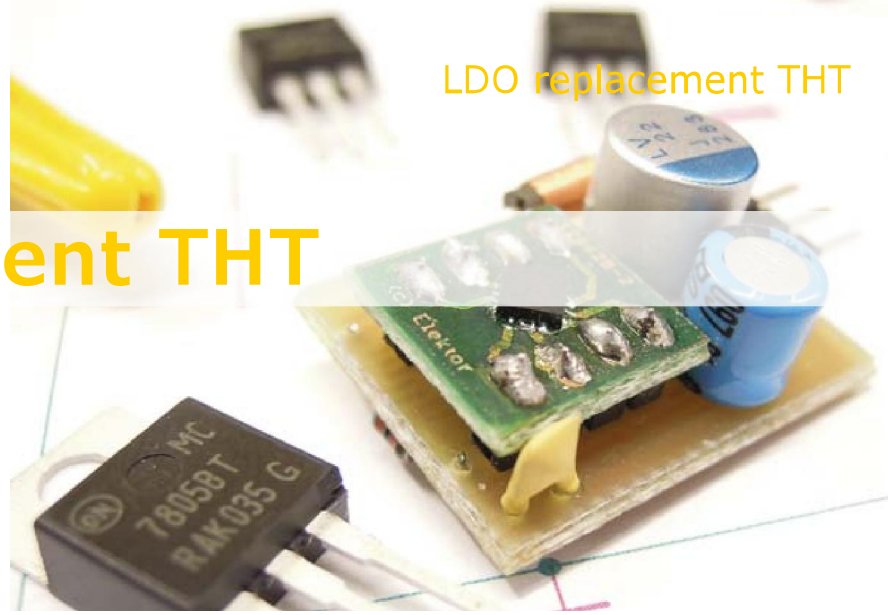
(120625)

By **Thijs Beckers**
(Elektor Editorial & Labs)

Internet Links

[1] www.elektor.com/120212

[2] www.elektor.com/120625

[3] www.elektor-labs.com/120625

Cypress TrueTouch™ Gen4 implements Multitouch in Fujitsu 4G smartphone

Cypress Semiconductor Corp. announced that Fujitsu Limited has selected the TrueTouch™ Gen4 touchscreen solution from Cypress to implement the touchscreen in the new Arrows V F 04E smartphone available from NTT DOCOMO. The new Fujitsu phone, which uses the Android operating systems and operates on the 4G LTE network, leverages the Gen4 solution's leading signal-to-noise ratio (SNR) to deliver highly responsive and accurate multitouch performance in any operating environment. Gen4 also provides industry-leading waterproofing capability, enabling accurate touch input and finger tracking in the presence of moisture from rain, condensation, or sweat.

The Fujitsu Arrows V F-04E has a quad-core CPU and 2 GB of RAM to provide the power to process information at 4G LTE speeds of up to 100 Mbps. The phone features a dynamic 4.7-inch HD display that offers precise tracking of up to ten fingers with the Gen4 controller. The TrueTouch solution also provides a Charger Armor feature that enables mobile phones to operate in the presence of very noisy chargers. The Gen4 family also offers features that only TrueTouch can deliver, such as built-in waterproofing functionality that allows the product to meet IP-67 standards without extra sealants or shield layers.

The Gen4 family was designed from the ground up to deliver leading signal-to-noise ratio (SNR) in real-world applications. The Gen4 touchscreen controller delivers 10 V Tx along with Cypress' proprietary Tx-Boost™ multi-phase Tx solution for the highest in-application SNR in the industry. Combined with TrueTouch's unique Charger Armor noise-mitigating technology and display noise cancellation algorithms, Gen4 leads the field in performance in the presence of noise. Gen4 also offers value-added features such as built-in waterproofing capability and support for display integrated designs.

www.cypress.com   (130048-VI)

calibrated to measure mass flow in bypass configuration

Differential pressure sensors for flow measurement in bypass configuration

[this item was published with an incorrect photograph in the March 2013 edition; we extend our apologies to Sensirion. *Editor*]

The Swiss sensor manufacturer Sensirion recently added new differential pressure sensors to its proven SDP600 series. The SDP601 and SDP611 sensors are differential pressure sensors specifically calibrated for measuring mass flow in a bypass configuration.

In a bypass configuration, an orifice or a linear flow restrictor is used to generate a differential pressure in a flow channel. The resulting pressure is measured over the orifice or the linear flow component. The difference between the pressures before and after the orifice correlates to the volumetric flow in the channel, depending on the specific characteristics of the flow restriction component. The mass flow can therefore be calculated from the measured pressure drop (differential pressure) over the orifice.

A bypass configuration is highly suitable for applications where individually adapted flow channels are necessary or where small differential pressures must be measured with very high precision. Especially for HVAC applications, which often involve measuring large flow volumes, it is the ideal solution.

The sensors expand the broad product range of Sensirion's digital differential pressure sensors in the SDP600 series. Along with the other products in this series, they offer a digital I2C output and are fully calibrated and temperature compensated. Operating on the principle of calorimetric flow measurement, the CMOSens® differential pressure sensors achieve outstanding sensitivity and accuracy even at very low pressure differences (below 1 Pa). They also have very high long-term stability and are free from zero-point drift. Like all devices in the SDP600 series, the SDP6x1 sensors are available in two versions. The SDP601 is intended for direct threaded connection to a pressure manifold with O-ring sealing, while the SDP611 is designed for tube connection.

www.sensirion.com/en/sdp600   (120749-I)

Newark element14 celebrates Raspberry Pi's first birthday

Straight after the phenomenon that is Raspberry Pi celebrated its first birthday, element14 launced "Raspberry Pi Month" with a series of initiatives to celebrate the outstanding success of the microcomputer.

Raspberry Pi made its European debut on the element14 stand at Embedded World 2012. To celebrate an extraordinary year for the category defining ARM based microcomputer, element14 presented key innovators behind Raspberry Pi, Co-Founder Pete Lomas, and ecosystem developer and PiFace designer Dr Andrew Robinson with the Innovation Award as voted for by the element14 Community at the end of last year. A giant Raspberry Pi birthday cake was also handed over, with visitors to the event being able to get their own "Slice of Pi".

The success of the Raspberry Pi and the related accessories has been phenomenal. When the Model B board launched in 2012 there were 500,000 hits on the element14 website every 15 minutes with 600 orders, pre-orders or hits on the site every second. These stats and others relating to the 'Life of a Raspberry Pi' can be seen in the infographic. As part of Raspberry Pi month element14 will also be asking people to vote for their favorite project using the microcomputer and accessories. 'MyPi' will highlight ten of the top projects that will be promoted via the community and members will be asked to discuss and select their favorite as well as put forward their own outstanding projects.

Throughout March 2013 various roadtests of the PiFace and Gertboard will be taking place on the element14 Community and for those new to the accessories a PiFace and Gertboard webinar will take place on 20th March.

Claire Doyle, Global Head of Raspberry Pi at element14, said: "When we launched Raspberry Pi a year ago who would have believed the phenomenal response this micro computer would have generated. Since we displayed it, exclusively at Embedded World 2012 element14 has been right at the forefront of technology through all the new accessory launches, the PiView, the WiPi, the 512 MB Model B board to name a few. It has been a fantastic year and one we have been privileged to be a part of. The activities arranged for Raspberry Pi month show our ongoing commitment to the Foundation and all the customers that have created ingenious projects with their boards."

With over 9 million views of the Raspberry Pi group on the element14 Community the success of the micro-computer is clear for everyone to see. To join the conversation, go to http://www.element14.com/raspberrypi.

Computing enthusiasts can expect to see a lot more from Raspberry Pi in 2013 and you can find out more about the minicomputer at Newark element14 in North America, Farnell element14 in Europe and element14 in Asia Pacific. The Raspberry Pi is also available through CPC in the UK and MCM in the United States.

(120749-IV)   www.newark.com/raspberrypi
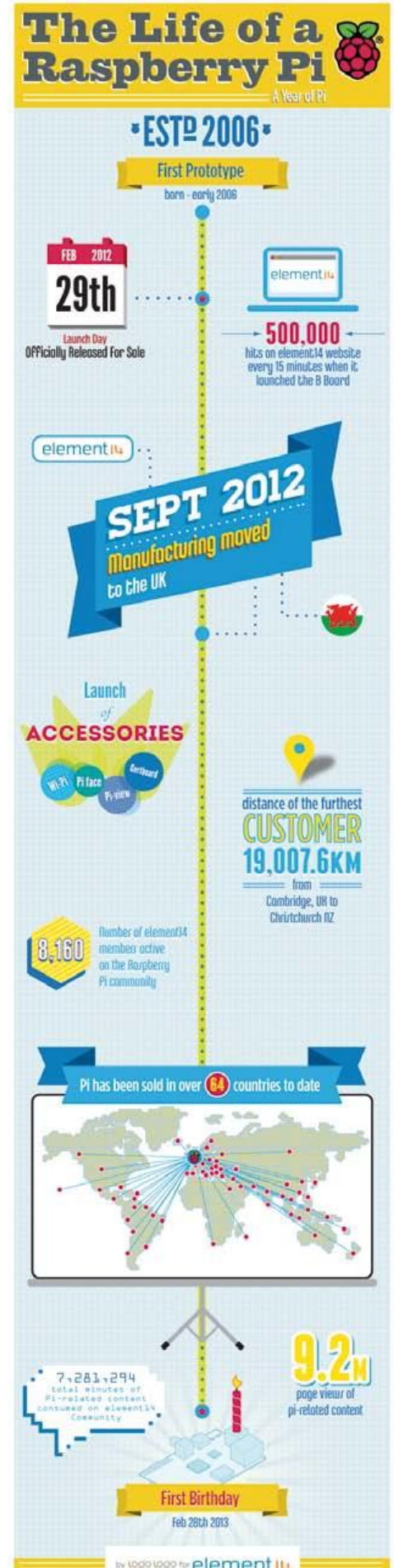
Virtual Money

By
Tessel Renzenbrink
(Elektor TTF Editor)

Virtual money is on the rise and brings with it new methods of transferring value into the world. There are different types of payment systems. Closed virtual systems, such as in-game money, operate only online. The money is earned in the game and can only be spent there. With uni-directional payment systems, national currencies can be converted into virtual money, but not the other way around. Perhaps the most interesting are the bi-directional payment systems, whereby virtual and traditional currencies can flow freely back and forth.

The virtual payment systems M-Pesa and Bitcoin belong to the latter category. The mutual exchange means that the virtual and traditional systems influence each other. The virtual payment systems are becoming subject to the financial regulations of the formal economy, while the incumbent financial institutions are being confronted with a changing playing field.

Electronic money combats poverty

More than half of the adult world population has no, or difficult, access to the financial services of banks. This mostly applies to the very poor, who earn less than $2 per day and are therefore largely excluded from participating in the formal economy. For example, they are unable to obtain a loan to start a business and keep their savings as cash with all the risks that come with it. This financial exclusion contributes to the vicious circle of poverty.

People without access to banking services are confronted with various hindrances that make joining the formal financial system more difficult. In developing countries, banks often have a small density of branches, which means that people have to travel far to carry out a transaction. In addition, transaction fees are relatively high compared to the amount of money being transferred. Also, the bureaucratic rigmarole that accompanies the opening of a bank account is a problem.

But where the traditional banking system has difficulties reaching the less well-off, mobile technology finds the path much easier. In 2012, 75 percent of the world population had access to a mobile phone. In Africa, the success of this technology is used to solve the problem of financial exclusion.

M-Pesa

The Kenyan mobile payment system M-Pesa was launched in 2007 by the telecommunications companies Safaricom and Vodafone. The system is very simple and accessible by anyone with a mobile phone. Somebody working in the city and wanting to transfer money to his family in a rural area can, at an M-Pesa agency, convert cash into electronic money in his M-Pesa account. He subsequently sends a text message to his family who can then, at another M-Pesa dealer, convert this back into cash. The receiver does not need to have an M-Pesa account, but the sender does.

This mobile payment system has, in one fell swoop, removed three big obstacles to obtaining financial services. There is no minimum deposit amount required to open an account and there are no periodic account fees: the payment to Safaricom is per transaction. Because many retailers have started to act as M-Pesa agents the density in Kenya is high. There is also no complicated bureaucracy, showing the national identity card is sufficient.

M-Pesa is therefore a huge success. According to Safaricom there are 15 million active users (out of a population of 42 million) who have more than 40,000 outlets at their disposal. Every month, on average, 80 million Kenyan shillings (just under $1 million) moves through the system, which is equal to about 30 percent of the Gross National Product.

Financial regulator

M-Pesa was created without a business partner from the traditional financial world. In the report *Regulatory issues around mobile Banking*, Paul Makin describes how the M-Pesa team, before launching the payment system, looked for a financial partner with knowledge of the market. But none of the existing parties were interested. Makin is head of Mobile Money at Hyperion, the consulting company which was involved from the beginning during the development of M-Pesa.

The absence of a reputable partner resulted in a problem with those that provide financial supervision. It is their function to guarantee the stability and security of the financial system. Many of them were suspicious of the new player and were of the opinion that such a large-scale payment system should really be under the auspices of a bank. Makin writes: 'As one regulator has put it, without a trace of irony, "in view of the recent global financial crisis, we feel that only a bank provides the necessary stability"' [1].

In order to prevent problems with the financial regulators, the M-Pesa team have done everything to comply with the regulations, before launching the service. Security is ensured by, among other things, end-to-end encryption and *hardware security models* on the server-side. The system also complies with the *Know Your Customer* regulation, which requires that the identity of the user has to be verified before a transaction is carried out. Safaricom did ultimately obtain a special license from the regulators to permit it to start operating.

M-Pesa is clearly the case of a newcomer who, with an innovative product, disrupts the market and forces the incumbents to change. In the meantime the players are growing towards one another. In November 2012 Safaricom and Vodafone announced a collaboration with the Commercial Bank of Africa. Under the name M-Shwari they offer M-Pesa users the option to save money and earn interest and to borrow small amounts of money.

National and international regulators from both the telecommunications as well as the financial sector are working to reform the regulations to make mobile banking possible. The fight to reduce poverty is a significant motivation. But the financial institutions also see this as a way of getting a greater share of the informal money stream into the formal economy.

M-Pesa agent in Mwanza
(Photo: Emil Sjöblom CC BY-SA)

Bitcoin

The decentralized electronic currency Bitcoin also offers a fresh look at the transfer of value. This was designed as a peer-to-peer payment system without a central authority.

Using traditional electronic payments between two people there is always a third party required to validate the transaction. This third party — a bank for example — is necessary because the receiver cannot verify whether the payer has already spent the electronic currency or not. It is the role of the central authority to maintain supervision of all the transactions in the system. With that knowledge they can establish whether the payer is not spending the money twice.

The Bitcoin protocol has an innovative solution that makes it possible to verify transactions without introducing an all-knowing third party, but by making the network itself the all-knowing party. All (anonymous) transactions, from the very first to the most recent, are recorded in a growing archive that is accessible to all participants of the Bitcoin network.
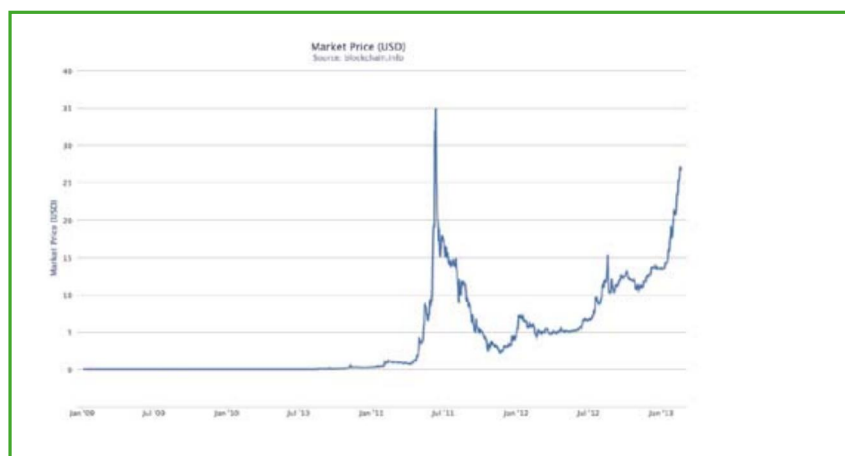
Figure 1.
Exchange rate of the Bitcoin.

Figure 2.
M-Pesa agent in Bunda
(Photo: Emil Sjöblom CC BY-SA 2.0)

To prevent changes to this archive of transactions for fraudulent purposes, the computing power of the entire network is used. Every node receives the most recent transactions in a record, which is called a block. To verify this, a mathematical puzzle has to be solved. The difficulty of the puzzle is set so that the cumulative computing power of the entire network will 'solve' one block every ten minutes. Every block contains all the previous blocks and so forms a chain of blocks that contains all previous transactions. In order to change one block in the chain of blocks a fraudster will have to recalculate all the previous blocks as well. This mode of operation means that a fraudster who wants to alter the archive can only do this if he has more than half of all the computing power of the network at his disposal.

Just like M-Pesa, Bitcoin is not entirely independent. The electronic currency overlaps with traditional financial systems. The currency can be exchanged into, for example, dollars or Euros on Bitcoin-exchanges such as Mt.Gox and Bitcoin Central. The latter announced in December 2012 that it is allowed to offer financial services in Europe. This makes it possible to deposit your salary in dollars or euros into your bank account and to spend it in Bitcoins. The company behind the exchange, Paymium, sees this as a step to a wider acceptance of Bitcoin.

The anonymous decentralized design of Bitcoin is a *game-changer* compared to traditional electronic payment systems. Although all the transactions can be traced, it is not known who made them so that all payments are anonymous. This makes it different from most of the traditional payment systems. Certainly in developed coun-

tries money flows, for the most part, through the formal economy. All those transactions can be traced, which makes taxation possible and there is a check on whether the money is being used for illegal purposes or not.

Bitcoins are not 100% anonymous however. Certainly when there is an exchange with more conventional currencies such as at the Bitcoin-exchanges, it is possible to trace the identity of the payer. But compared to the complete transparency of the present banking system Bitcoin offers much greater privacy.

European Central Bank
The value of the Bitcoin is created by its scarcity and the trust of its users. Other than with some national currencies, for example, the exchange rate is not set by some regulatory authority. Critics see this as a weakness of the decentralized system, because it can lead to a strongly fluctuating exchange rate which could encourage speculation. This could cause the system to collapse. But the statistics show that while the exchange rate is indeed irregular, there has been a rising trend since January 2012. (**Figure 1**).

In the meantime, Bitcoin has caught the attention of larger banlks such as the European Central bank. The ECB researched to what extent the virtual currency could threaten its core roles, which includes guarding the stability of the exchange rate and the financial stability [2]. For the time being the bank sees few reasons for concern, mainly because the monetary value circulating in the system is so low that it has no discernible effect on the economy outside of it. But nevertheless the bank takes the currency very serious. This is shown by the fact that the bank asserts that Bitcoins fall under its responsibility and that it can make a contribution to the financial innovation and development of alternative payment systems.

(120752)

Internet Links

[1] www.oecd.org/ict/4d/43631885.pdf

[2] www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemes201210en.pdf
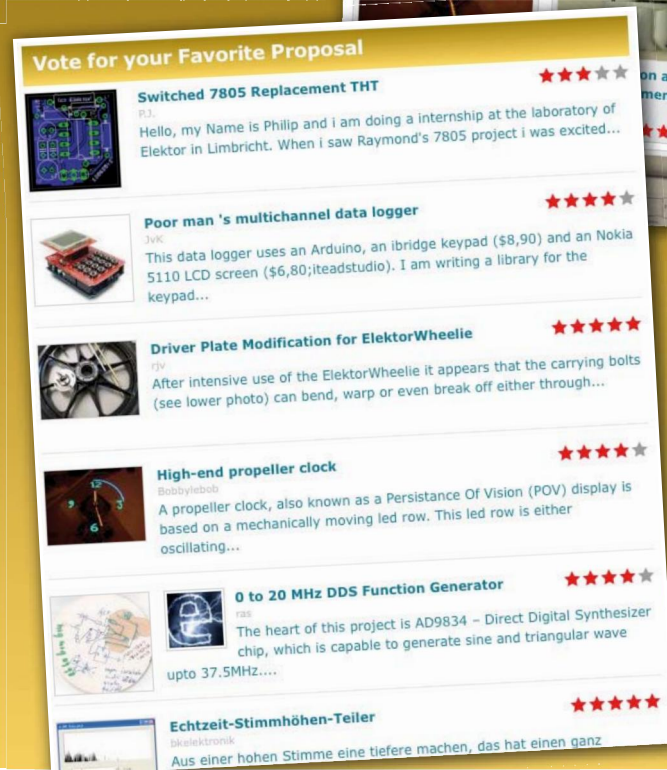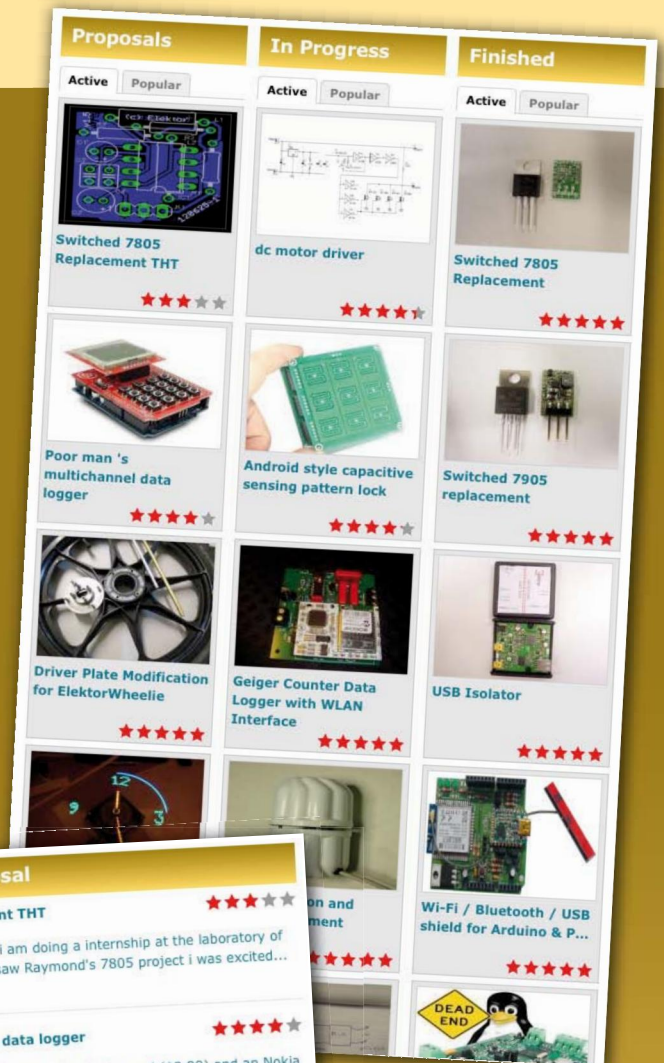
The Curse of the Collector

By **Reginald Neale**
(USA)

Fairchild 7050 DVM. I used this DVM in the late 60's. It was one of the first to employ dual-slope integration, resulting in an intrinsically stable zero. By making the integration time a multiple of the mains period, the effect of line-frequency noise is drastically reduced. Photo courtesy of Marvin Collins.

Civil Defense V-742 Pen Dosimeter. The scale image is what you see when you hold it up to the light and look through the lens at the end. The V-742 is intended to be clipped to clothing, and measures cumulative dose of radiation. The Dosimeter is charged with a high voltage, which sets the hairline indicator to zero. Radiation allows the charge to leak away, causing the indicator to move to the right. The instrument requires a special charger. It's still used by FEMA, and available in several dose ranges.

The Brown Converter. The 'converter' is actually a set of vibrating contacts used in a synchronous-detector amplifier system. It was part of a vintage chart recorder for recording millivolt-level DC signals.

Classic Ham Radio Receiver. Collins 2051 Owned, restored and photographed by Marvin Collins, W6OQI.

Heathkit. The Big Name in build-it-yourself electronics. I built this benchtop PSU for tubes in the late 1950's.

More virtual museums on the web
If you enjoyed this show you may also be interested in some of these sites:
www.conradhoffman.com
www.oneillselectronicmuseum.com/page10g.html
www.electricstuff.co.uk/
www.tubecollector.org/list.php?L=:
www.unusualmuseums.org
http://wps.com/archives/tube-datasheets
www.ieee-virtual-museum.org
www.seas.upenn.edu/~museum
http://frank.yueksel.org
http://www.heathkit-museum.com
http://semiconductormuseum.com/Museum_Index.htm
http://silicongenesis.stanford.edu
Zillions of relevant sites... and of course they're always changing. Warning: these are incredible time sinks. But you already knew that.

Fluke 1941A digital counter from early 70's with nixie tube display.

The equipment and parts pictured here were accumulated over a long career in electronics, during which I have proved tragically unable to discard anything elegant, exotic, wonderful, curious, weird or of exceptional quality, no matter how remote the chance that I might put it to some future useful purpose. This month's Retronics pages are dedicated to those of you who are similarly afflicted.

Books!
Some books which relate to the technology described on these pages:
70 Years of Radio Tubes, John W. Stokes, Vestal Press 1982
Procedures in Experimental Physics, J. Strong, Prentice-Hall 1938
The Amateur Scientist, C. L. Stong, Simon & Schuster 1960
Reference Data for Radio Engineers, Federal Telephone and Radio Corp. 1946

Geiger counter from the late 40's. After WWII, the public was fascinated with 'atomic energy.' Some people bought these instruments and went prospecting for uranium.

Top Hat Galvo. The Leeds & Northrup 'top hat' galvanometer is familiar to anyone who worked in yesterday's scientific laboratories. It detects tiny electrical currents by using a beam of light as a long, massless pointer. It's about 8" (20 cm) high. Today's galvos are mostly used in scanning applications. They are much smaller, more rugged, and orders of magnitude faster. Photo courtesy of Conrad R. Hoffman.

Philco oscilloscope used in 1950's TV servicing. No calibrated timebase or voltage scales. Very primitive signal synchronisation. Blue-gray hammertone cabinet finish is typical for instruments of the period. Photo courtesy of Marvin Collins.

Grid Dip Meter from the 50's. This was a non-contact way to check the resonant frequency of a circuit. The probe coil sticking out the top of the meter was coupled to the circuit under test. By tuning the dial on the left edge until the meter indicated a sharp dip, the frequency could be read from the dial. Several plug-in coils were needed to cover a range from 600 kHz to 300 MHz.
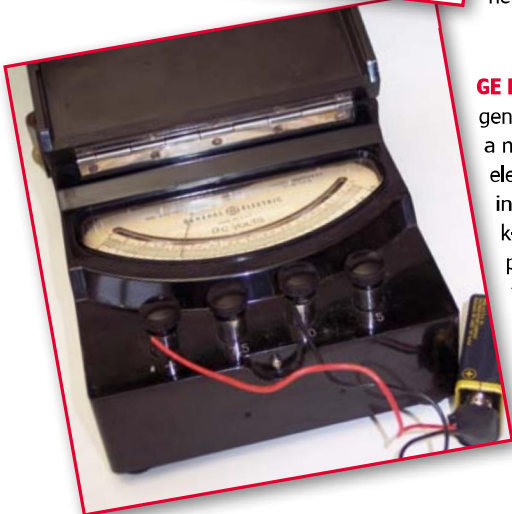
GE Lab Meter. Multi-range laboratory voltmeter, 1950's. Modern DVMs generally have a fixed 10 megohm input resistance on all ranges, with a negligible loading effect on most measurements. Not so for these electromechanical meters. They have range-dependent resistances, in this case 200 ohms per volt. On the 7.5 volts range, that's 15 k-ohms, low enough to load down most real-world circuits. Another problem with analogue meters: the pointer lies in a plane above the plane of the meter scale. Notice the mirrored arc alongside the scale. When you adjust your viewing position so that the reflection of the pointer in the mirror is superimposed with the pointer itself, parallax reading error is minimised.

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

Hexadoku   Puzzle with an electronic touch

Seriously, this month's Hexadoku is not an April spoof and it does have a unique solution. We kindly challenge you to defeat the blank boxes again and complete the puzzle. Enter the right numbers or letters A-F in the open boxes, find the solution in the grey boxes, send it to us and you automatically enter the prize draw for one of four vouchers.

The Hexadoku puzzle employs numbers in the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

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

Solve Hexadoku and win!

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

Participate!

Before May 1, 2013, supply your personal details and the solution (the numbers in the grey boxes) to

www.elektor.com/hexadoku

Prize winners

The solution of the January & February 2012 Hexadoku is: **02518.**   The Elektor £80.00 Voucher has been awarded to Jean-Claude Pouillet (France). The Elektor £40.00 Vouchers have been awarded to Mike Corteling (Australia), Hans Pflister (Germany), and Georgios Mpeinas (Greece).

Congratulations everyone!

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

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

By **Gerard Fonte** (USA) # Mind Altering

There are always times when someone in authority says "no" when you really want to hear "yes". Unfortunately, the methods that were so effective as a child (whining, stamping your feet, and repetitive demands) are not all that productive as adults. And, in most cases, once a person says "no" that's the final answer. However there are some strategies that can give you the best chance of changing that person's mind.

Face Up

By far the most important aspect of getting to yes is to have a meeting with Mr. No. It's very easy to refuse someone in an email or on the telephone. It's much harder when the person is sitting five feet away. Of course, getting that meeting may be difficult. You must be polite and deferential. While a confrontational attitude may work in the movies, it doesn't work in real life. Presumably, he has given you some reason for his decision. Perhaps it's too expensive or it's too risky or it's not safe. If there is no reason given, you will need to find out before you can proceed. This can be done directly (asking him) in indirectly (asking someone who is familiar with the issues). In order to have any possibility seeing him in person, you must squarely address this issue in a way that you know that is acceptable to him. (And if you can't meet his requirements, don't waste everybody's time. You'll just create bad feelings and make it harder for any future disagreement.)

For example, suppose safety is the issue. You could start by informing him that you have made a number of safety improvements. But then move on from there by saying something like: "Safety is a critical consideration for me as well. I'd like to go over these improvements with you and I am hoping that you will tell me exactly what additional safety items you think we need to look at." If safety is the real problem, it's difficult for him to deny this very reasonable request. And the subtle use of "we" instead of "I" includes him instead of excluding him. So it's no longer "me versus you", it's "us".

Of course, safety may only be an excuse. It's common to just throw out something to make the problem go away. In which case, the last thing Mr. No wants to do is to waste time with a fictitious discussion about improving something he doesn't like. If he says no to the meeting and it is apparent that something other than safety is involved, then it's a lost cause. He is hiding the true reason from you. Which means he doesn't want to open it up for discussion. If you can't talk about it, you can't change his mind. The matter is closed.

Face Off

Okay, you've got your meeting — which is half the battle. Now you have to convince him that there is a better answer than no. Treat it just like a job interview. Dress properly, arrive on time

and above all, be fully prepared. You should be able to anticipate his questions and have good answers ready. There's nothing wrong with illustrations, simulations and any other reasonable use of props. When you walk in his office, start with something like: "Thank you very much for meeting with me, Mr. No. This project is important to me and I hope that we can work something out." And smile warmly. Again, be pleasant, polite, deferential and cheerful. Attitudes make for good drama. But rarely get favourable results. You can be certain that he doesn't want any type of altercation.

It's very important to get Mr. No to talk. That's the only way you'll know how to approach the issue. Otherwise, you will be shooting in the dark. You may address all of his concerns and you may not. And don't be surprised if something new comes up. Take it in stride and answer it. Your goal is to provide the information needed to convince him to change his mind. This is not a competition, it's a request. He has the power and authority, you don't. Stay calm. Don't get frustrated if he fails to see some simple point. It's your job to enable him to see that point, not his.

But above all, listen to what he has to say. It's easy to get focused on your presentation and reasoning, and miss the thrust of a question or comment. It's common to think you know the answer before he finishes the question. These can be a fatal error. Whenever he asks a question or makes a comment, stop and think. Ask yourself: "Why is he asking/saying this?" That will point towards a topic that needs clarification. It may not be precisely what was asked.

Face Facts

You may get your answer immediately or he may require some time to consider it. He may ask for more information to make his decision. You may need to compromise to get a yes answer. This may or may not be acceptable to you. You should always send a thank-you email (or letter!) regardless of the outcome. If the answer is still no, call it quits. Pursuing a closed issue only annoys people. Nevertheless, a "no" can still be considered a positive, if you have opened a door to future discussions on other topics. Having someone with power and authority listen to you is a big plus.
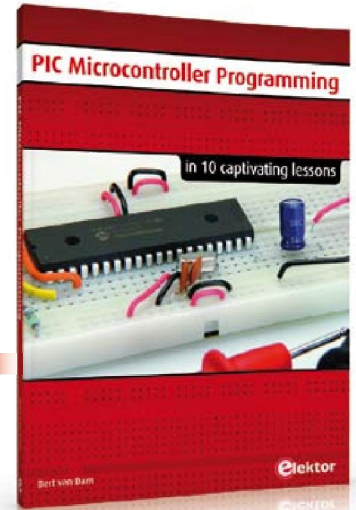
(130102)

NEW

Limited Time Offer for GREEN and GOLD Members!
15% DISCOUNT + FREE SHIPPING
www.elektor.com/retronics

80 tales of electronics bygones
1 Retronics

Quite unintentionally a one-page story on an old Heathkit tube tester in the December 2004 edition of Elektor magazine spawned dozens of 'Retronics' tales appearing with a monthly cadence, and attracting a steady flow of reader feedback and contributions to the series. This book is a compilation of about 80 Retronics installments published between 2004 and 2012. The stories cover vintage test equipment, prehistoric computers, long forgotten components, and Elektor blockbuster projects, all aiming to make engineers smile, sit up, object, drool, or experience a whiff of nostalgia. Owners of this book are advised to not exceed one Retronics tale per working day, preferably consumed in the evening hours under lamp light, in a comfortable chair, with a piece of vintage electronic equipment close and powered up.
193 pages • ISBN 978-1-907920-18-9
£24.80 • US $40.00

A whole year of Elektor magazine onto a single disk
2 DVD Elektor 2012

The year volume DVD/CD-ROMs are among the most popular items in Elektor's product range. This DVD-ROM contains all editorial articles published in Volume 2012 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 graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.
ISBN 978-90-5381-273-0 • £23.50 • US $37.90

10 captivating lessons
3 PIC Microcontroller Programming

Using the lessons in this book you learn how to program a microcontroller. You'll be using JAL, a free but extremely powerful programming language for PIC microcontrollers. Assuming you have absorbed all lessons you should be confident to write PIC microcontroller programs, as well as read and understand programs written by other people. You learn the function of JAL commands such as include, pin, delay, forever loop, while loop, case, exit loop, repeat until, if then, as well as the use of functions, procedures and timer- and port interrupts. You make an LED blink, build a time switch, measure a potentiometer's wiper position, produce sounds, suppress contact bounce, and control the brightness of an LED. And of course you learn to debug, meaning: how to spot and fix errors in your programs.
284 pages • ISBN 978-1-907920-17-2
£29.50 • US $47.60

Taming the Beast
4 FPGA Development Board

FPGAs are unquestionably among the most versatile but complex components in modern-day electronics. An FPGA contains a maze of gates and other circuit elements that can be used to put together your own digital circuit on a chip. This FPGA development board (designed in the Elektor Labs) shows how easy it is for any electronics enthusiast, whether professional or amateur, to work with these programmable logic devices.
Module, ready build and tested   Art.# 120099-91
See www.elektor.com/fpgaboard

LabWorX 2
5 Mastering Surface Mount Technology

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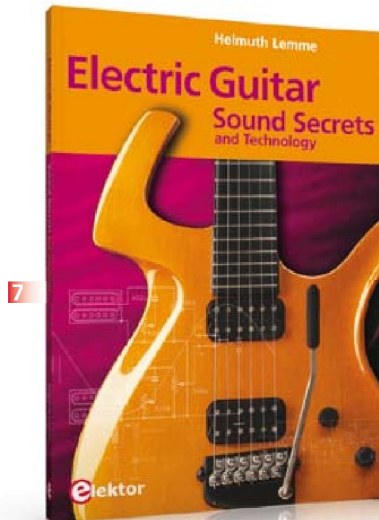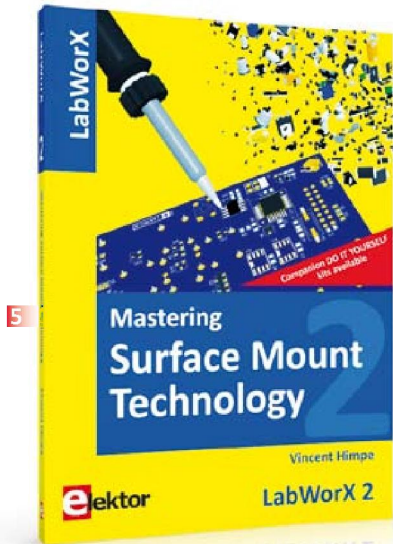
introduce surface mount technology in your workflow. Even if you are on a budget you too can jumpstart your designs with advanced fine pitch parts. Besides explaining methodology and equipment, attention is given to SMT parts technologies and soldering methods. Many practical tips and tricks are disclosed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

282 pages • ISBN 978-1-907920-12-7
£29.50 • US $47.60

Meet BOB
6 FT232R USB/ Serial Bridge/BOB

You'll be surprised first and foremost by the size of this USB/serial converter – no larger than the moulded plug on a USB cable! And you're also bound to appreciate that fact that it's practical, quick to implement, reusable, and multi-platform – and yet for all that, not too expensive! Maybe you don't think much of the various commercially-available

FT232R-based modules. Too expensive, too bulky, badly designed, that's why this project got designed in the form of a breakout board (BOB).

PCB, assembled and tested
Art.# 110553-91 • £12.90 • US $20.90

Sound Secrets and Technology
7 Electric Guitar

What would today's rock and pop music be without electric lead and bass guitars? These instruments have been setting the tone for more than forty years. Their underlying sound is determined largely by their electrical components. But, how do they actually work? This book answers many questions simply, in an easily-understandable manner. For the interested musician (and others), this book unveils, in a simple and well-grounded way, what have, until now, been regarded as manufacturer secrets.

The examination explores deep within the guitar, including pickups and electrical environment, so that guitar electronics are no longer considered highly secret. With a few deft interventions, many instruments can be rendered more versatile and made to sound a lot better – in the most cost-effective manner.

287 pages • ISBN 978-1-907920-13-4
£29.50 • US $47.60

programming step-by-step
8 Android Apps

This book is an introduction to programming apps for Android devices. The operation of the Android system is explained in a step by step way, aiming to show how personal applications can be programmed. A wide variety of applications is presented based on a solid number of hands-on examples, covering anything from simple math programs, reading sensors and GPS data, right up to programming for advanced Internet applications. Besides writing applications in the Java programming language, this book also explains how apps can be programmed using Javascript or PHP scripts.

244 pages • ISBN 978-1-907920-15-8
£34.95 • US $56.40

Further information and ordering:
www.elektor.com/store

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

## NEXT MONTH IN ELEKTOR MAGAZINE
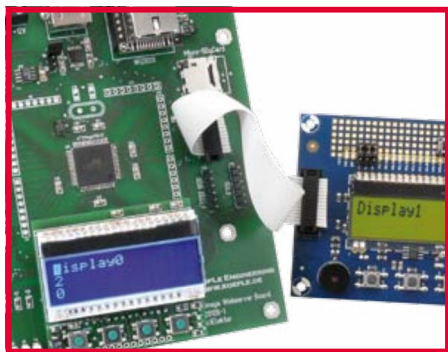
EFL: Embedded Firmware Library

The modular structure dubbed EFL isn't just another library for microcontroller programming. EFL allows your applications to be developed in a hardware independent way so you can easily port them from one board to another. Porting is not restricted to one controller manufacturer, and running EFL within embedded Linux is possible in principle. Thanks to various functions for ADC, PWM, display, TCP/IP, SD cards and many more, the library is also suitable for beginners, including C.

Introduction to Python

If you want to make your project user friendly, equip it with a PC connection and offer sumptuous PC software. For the latter, programmers increasingly use the Python language, which we set out to describe in a short series of articles starting next month. Python syntax results in compact and clear code, and in addition there are powerful libraries available for a wide range of applications. Importantly, however, Python is an interpreter, meaning if you change the code, the results are instantly visible on the screen. This makes Python eminently suitable for beginners.
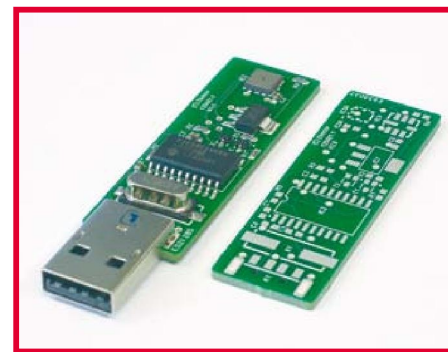
BaroStick

USB sticks for measurement of air pressure being difficult to find, the author set out to design one himself. The PCB can optionally be fitted with a pressure sensor type BMP085 or BMP180 from Bosch Sensortec. The communication between the I²C bus sensor and the USB is handled by a PIC18F14K50 microcontroller. A Windows application looks after the processing and storage of measured values on the PC. For sure, the source code is available free

Note: due to design issues the Lost Model Finder announced last month is now scheduled to appear in the June 2013 edition.

Article titles and magazine contents subject to change; please check www.elektor-magazine.com

Elektor UK May 2013 edition published April 24, 2013.

See what's brewing @ Elektor Labs 24/7

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