Control by Tablet and Smartphone

FT311D-based Android User Interface Builder

- Android User Interface Builder
- Lathe Tachometer
- Touch-2-Switch
- Seismic Detector
- Microcontroller BootCamp
- XXL LED VU Meter
- DIY I²C for RPi
- Dimensioning Photovoltaic Cell Arrays
- Hack-Your-Own Reflow Oven
- Multi-switch
- Lights Control
- The MAS6510
- 3-D Print Your Own Ink
- A Jumbo PCB
- Elektor ComputerScope (1986)
ARDUINO UNO REV.3
The most popular board with its ATmega328 MCU

Features
- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage: 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Channels: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (of which 8 KB used by bootloader)
- SRAM: 5 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

£23.95 • £27.50 • US $38.00

ARDUINO DUE
32-bit power thanks to an ARM processor

Features
- Microcontroller: AT91SAM3X8E
- Operating Voltage: 3.3V
- Input Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 20 (of which 7 provide PWM output)
- PWM Channels: 7
- Analog Input Pins: 12
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (of which 4 KB used by bootloader)
- SRAM: 2.5 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

£46.95 • £52.90 • US $72.00

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

Features
- AVR microcontroller
- Operating Voltage: 5V
- Input Voltage: 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 54 (of which 15 provide PWM output)
- PWM Channels: 12
- Analog Input Channels: 16
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 512 KB (all available for the user applications)
- SRAM: 96 KB (two banks: 64KB and 32KB)
- Clock Speed: 16 MHz

£46.95 • £53.90 • US $73.00

ARDUINO YÚN
Two processors

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

£60.95 • £69.95 • US $95.00

ARDUINO ETHERNET
Networking has never been easier

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

£46.95 • £52.90 • US $72.00

ARDUINO LEONARDO
Especially good for USB applications

Features
- Microcontroller: ATmega32u4
- Operating Voltage: 5V
- Input Voltage: 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 20 (of which 7 provide PWM output)
- PWM Channels: 7
- Analog Input Pins: 12
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 200 mA
- Flash Memory: 32 KB (of which 4 KB used by bootloader)
- SRAM: 2.5 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

£21.95 • £24.90 • US $34.00

ARDUINO ETHANET
Networking has never been easier

Features
- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage: 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 20 (of which 7 provide PWM output)
- PWM Channels: 7
- Analog Input Pins: 12
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (of which 4 KB used by bootloader)
- SRAM: 2.5 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

£46.95 • £53.90 • US $73.00

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

10% OFF for GREEN and GOLD Members
Join the Elektor Community
Take Out a GOLD Membership Now!

Your GOLD Membership Contains:

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

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

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

Connect with us! www.facebook.com/elektorim www.twitter.com/elektor
XXL LED VU Meter
The circuit described here has an exceptionally wide input range of 60 dB and a step size of 1 dB thanks to the use of 6 LM3915 driver ICs, which control a total of 60 LEDs. An interesting aspect of this design is that it is entirely analog (there is no microcontroller to be found) and that has a didactic purpose.

DIY I²C for Raspberry Pi
What do you need for this? Actually not that much: a Raspberry Pi board, an SD card with the Raspberry Pi operating system and I²C drivers, a PIC 16F88 microcontroller, a PIC programmer, and a breadboard with an LED, an LDR and a few bits of wire.

Dimensioning
Photovoltaic Panel Arrays
What’s the promise of monocrystalline silicon, polycrystalline silicon, amorphous silicon, and CIS cells
when arranged in a solar panel and charging a battery?

56 Hack-Your-Own Reflow Oven
Thanks to SMDs, printed circuits are getting smaller—and thus the cost of projects. The fact remains that for soldering these miniature components, you need little fairy fingers... or a reflow oven! You’re hesitating?

74 Multi-switch Lights Control
Complex electrical wiring is required if a single load like hallway or corridor lighting is to be switched on and off by any one of a large number of switches. Here we propose a work-around using a DIY thyristor.

Industry

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

DesignSpark

64 DesignSpark Tips & Tricks
Day #11: Working with Large Designs
This month we learn how to deal with multiple-schematic sheets in DesignSpark PCB.

Varactor Diodes
Weird Components—the series.

Labs

68 3D-print Your Own Ink
Our 3D printers really should be put to good use, like printing stuff that’s useful to electronics engineers.

70 Jumbo PCB To Escape From Labs
The Elektor lab PSU design incorporating the PCB transformer we wrote about is one step closer to prototyping and testing.

72 Chip Tips: The MAS6510
Highlights of a new capacitance-to-digital converter.

Regulars

80 Retronics
Elektor ComputerScope (1986). Almost 30 years ago a trainee at Elektor Labs designed an ambitious test instrument like an oscilloscope with PC support. Series Editor: Jan Buiting.

84 Hexadoku
The Original Elektorized Sudoku.

85 Gerard’s Columns: Lazy Days
A column or two from our columnist Gerard Fonte.

90 Next Month in Elektor
A sneak preview of articles on the Elektor publication schedule.
A www Guide to Elektor

This space being remarkable for intensive reading (even by my CEO) but drawing little response overall, I will venture to present a condensed what-where-why (www) guide to Elektor magazine within the constellation called Elektor International Media. You may have noticed that Elektor is no longer at a single .com website but divided across Magazine, Store, Labs and Forum. All are intensely online driven to serve our 300 K audience around the globe.

• Elektor Magazine at www.elektor-magazine.com is a service website where our valued Green and Gold Members find their digital copies and the archive of past editions, and are able to renew or change their membership. It is also home to the famous free software downloads with published articles, as well as all “Project” editions of Elektor.POST. Status: Okay; Up For Revision 2015.

• Elektor Store at www.elektor.com is our online retail activity as well as the portal to all of EIM’s websites and publications. Use it to shop around for books, PCBs, semi-kits, DVDs, and PCB fabrication equipment. Create and manage your Store account, track orders, find special offers, get in touch with Customer Services, view legal stuff. Status: Under Construction.

• Elektor Labs at www.elektor-labs.com showcases Elektor’s wide open kitchen, where the editors taste and select the dishes to refine and serve out in the printed magazine, with the help of authors, graphics people and the Elektor lab crew. Status: Okay; Lively.

• Elektor Forum at http://forum.elektor.com was relocated and restyled recently. It’s amazing what help you can find there, like for the recent 500 ppm LCR Meter. Good for older projects. Status: Okay; login procedure provisionally shared with Elektor Labs. A grand operation is currently underway as our web staffers are harmonizing all login data. After all, Magazine, Store, Labs can be browsed freely to an extent, but logging on is required at some point to place an order or reach content. Or, in case of the forum, reply to, create and write topics. When lost aim for www.elektor.com as our portal site. Bookmark it. When utterly lost aim for www.eimworld.com. When interested in hands-on electronics, start on page 8. Enjoy this edition of Elektor (page 17),

Jan Buiting, Editor-in-Chief

The Team

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

Connects You To

Supporting Companies

Not a supporting company yet?
Contact Peter Wostrel (peter@smmarketing.us, Phone 1 978 281 7708, to reserve your own space in Elektor Magazine, Elektor•POST or Elektor.com
A user interface is almost always needed when controlling and testing a piece of hardware. Ideally the interface would show data on a display and provide a keyboard for entering commands. Surely it should be possible to use a smartphone for the job? The FTDI FT311D is a flexible bridge that can interface your circuit to an Android smartphone or tablet. It offers options for seven digital outputs, four PWM outputs, asynchronous serial and I²C and SPI interfaces. To make things simple we have designed a small board to carry the chip: introducing the ‘Elektor Android breakout board’.

A large proportion of electronics projects, including those published in Elektor magazine, includes a serial interface over which readings are transmitted and commands are received, typically in the form of ASCII strings. A terminal program, running on a PC, is normally used for testing and sometimes even in normal operation. However, even in laptop form such a computer can be cumbersome to use as well as noisy and expensive. Smartphones and tablets offer a more conve-
nient alternative, and their touchscreen displays allow a pleasant control interface to be provided for the user.

On the technical side the main obstacle to this idea is that the only usable interface to the outside world offered by smartphones and tablets is a USB slave port. This means that our circuit must include a microcontroller implementing a USB host interface and we have to get involved in writing the firmware to handle the interface protocol. We also have to make sure that the operating system running on the mobile device is capable of sending and receiving bytes over USB under the control of a user-written application.

Open accessory
Fortunately the last problem at least has been solved for all modern Android smartphones and tablets. Google, the company responsible for developing the majority of Android, already has its finger on the pulse: recognizing the demand for an interface to RGB LED light shows, servo-controlled robots and other such ‘maker’ projects, it introduced in version 3.1 of Android the so-called ‘Android Open Accessory’, or AOA, mode. This offers the possibility of simple communications with external hardware over USB [1], with no need to write or install special USB drivers in the operating system. Internally two USB bulk endpoints are provided (for input and output), allowing data to be transmitted and received at up to 12 Mbit/s (USB Full Speed). For more information on how USB works see the Elektor article ‘Inside USB’ [2]. The external hardware must provide USB host functionality and the mobile device can also be charged over the USB port. When a board with suitable hardware (called an ‘accessory’) is plugged in, an AOA-enabled Android device will recognize it from its USB descriptors [3]. If the developer so wishes, an specified application can be launched automatically when the hardware is connected. Support is of course available to Android developers working on such hardware and software [4]. Nevertheless, designing hardware to provide USB host functionality and writing the corresponding firmware will be a daunting prospect for many, as the USB standard is rather complex, particularly when it comes to making ‘plug-and-play’ work reliably.

A neat device
Taking the complexity out of designing USB interfaces is something that Scottish chip designer FTDI has been working on for many years. Its best-known product is perhaps the FT232, an easy-to-use USB-to-serial converter. More recently the company introduced a programmable device with USB host functionality (the ‘Vinculum’), which makes it easy to design hardware that, for example, is capable of accepting USB memory sticks.

Around two years ago FTDI brought the FT311D [5] on to the market. Like the Vinculum, this device provides a USB host port and a number (seven, in this case) of spare pins for controlling other hardware. To make it easier to use the chip FTDI supplies it ready-programmed with firmware to implement the whole USB communications stack including the AOA protocol: see the block diagram in Figure 1. The bridge chip has a range of different interface modes. In the simplest case the seven pins of the FT311D can be used as digital inputs and outputs (GPIOs). Other modes provide for the output of PWM signals, asynchronous serial communications, and I²C and SPI bus interfaces (both as master and slave). The I²C option alone opens up the possibility of connecting a huge range of special purpose devices to the FT311D, from port expansion chips and A/D converters to real-time clocks. Practically any project, therefore, can be connected to an Android device using the FT311D bridge. The interface mode is simply selected by the
level on three digital input pins. For flexibility these can be connected to jumpers or switches (see the text box ‘Interface modes’). It is not possible to configure the device using a PC program or similar.

The simple route to an app

The manufacturer has designed a simple protocol for each of the interface modes. Packets containing a couple of bytes of payload data flit back and forth over the USB connection to instruct the FT311D how to set its outputs and read back its inputs. In each case the first byte in a packet from the smartphone or tablet to the device represents the command: in GPIO mode there are three different commands to configure the pins as inputs or outputs, to set or clear the outputs, and to read back the inputs. In PWM mode a different command sets the mark-space ratio from 5% to 95%, the byte immediately after the command specifying the PWM channel (from 0 to 3). Further commands configure the I²C, UART and SPI interfaces and read and write data over them. The commands are fully documented in the ‘FT31XD Programmer’s Guide’, which can be downloaded from FTDI’s website [5].

This protocol is of course handled inside the FT311D by its pre-programmed firmware. On the Android device side the main jobs are to assemble the command packets and to handle the protocol requirements of AOA mode on the USB interface. When an app is launched it can determine whether an FT311D-based accessory is in fact connected.

Fortunately FTDI supplies a suite of Java library files, one corresponding to each mode of the chip, to handle these tasks, and given a basic knowledge of Android programming it is easy to integrate these libraries into your own program. In the case of GPIO mode, for example, the library file FT311GPromiseInteractive.java provides the functions ResetPort, ConfigPort, WritePort and ReadPort. The library files are also embedded in small Android demonstration programs, each of which provides a graphical user interface on the smartphone or tablet. The GPIODemo app (see Figure 2), for example, displays seven buttons to control the outputs and seven ‘LEDs’ to indicate the status of the inputs. These demonstration projects, eight in all, not only provide a way to test the chip and its connections, but can also

Compatibility

All Android devices running version 3.1 (Honeycomb) or above support Android Open Accessory mode. Your device can tell you which Android version it is running: on our Samsung Galaxy Tab it is under ‘Settings’, ‘About device’.

Some devices running version 2.3.4 and above of Android also support AOA mode: devices vary depending on whether the manufacturer happened to include AOA support when customizing the operating system.

For older Android devices an alternative possibility for connecting to external hardware is the Elektor Andropod module, which uses a pre-programmed Vinculum device to communicate using the Android Debug Bridge (ADB). The module is available from Elektor [10].
form a basis for further app development. The demonstration projects and the libraries are in the Android.zip archive: this can be downloaded from FTDI’s website.

**Breakout board**

To make it easier for our readers to try out this neat device, Elektor Labs staffer Ton Giesberts has designed the ‘Elektor Android breakout board’, a small board available ready-built from Elektor (Figure 3). The circuit diagram is shown in Figure 4. Ton was (in part) inspired in his design by an article on the FT311D by Jeff Bachiochi published in our sister magazine Circuit Cellar in December 2013.

The Android tablet or smartphone is connected to K1, a USB A socket, on which a charging voltage VUSB (nominally 5 V) is also provided. The Android device, the FT311D and any other connected hardware are powered over K3, a micro USB socket. This choice has the advantage that the power adapter supplied with the smartphone or tablet, which normally uses the corresponding type of plug, can be repurposed.

Low drop-out regulator IC2 produces the 3.3 V supply required by IC1, the FTDI chip. External hardware is connected to K2, which carries the relevant signals from the FT311D as well as power at 5 V and 3.3 V. It is also possible to power the board and the Android device over the 5 V pin on this header.

Pins CFG0 to CFG2, which set the interface mode of the device, can be pulled to ground using jumpers. If a jumper is omitted, an internal pull-up in the device pulls the input high.

FTDI suggests connecting an LED to the /USB_ERR output, which it calls the ‘Error LED’. However, this is misleading: the LED will light continuously when the FT311D has successfully established a connection with an AOA-compati-

---

**Component List**

**Resistors**

- R1, R2 = 27Ω, 0.1W, 1%, 0603
- R3, R4 = 10kΩ, 0.1W, 1%, 0603
- R5 = 620Ω, 0.1W, 1%, 0603
- R6 = 1kΩ, 0.1W, 5%, 0603

**Capacitors:**

- C1, C2 = 47pF, 50V, 5%, C0G/NP0 0603
- C3, C6–C9, C11, C13 = 100nF, 25V, 10%, X7R 0603
- C4, C5 = 27pF, 50V
- C10 = 4.7µF, 6.3V, 10%, X5R 0603
- C12 = 10µF, 10V, 20%, X5R 0603

**Inductors**

- L1 = 600Ω @100MHz, 0.38Ω/0.5A, 0603, Murata BLM18AG6601SN1D

**Semiconductors**

- D1 = PRTR5V0U2X TVS diode
- LED1, LED2 = LED, 0805
- IC1 = FT311D-32L1C-R
- IC2 = KF33BDT-TR, LDO

**Miscellaneous**

- K1 = USB-A socket, PCB mount
- K2 = 10-pin pinheader, 0.1” pitch; round pins for plugging onto a breadboard (e.g. Harwin D01-9923246)
- K3 = Micro-USB-B socket, SMD
- JP1 = 6-pin (3x2) pinheader, 0.1” pitch, incl. 3 jumpers
- X1 = 12MHz quartz crystal, 20ppm, 18pf, 5x3.2mm
- PCB # 130516-1 v1.1 [6]
- or Elektor Android Breakout Board, ready assembled and tested, Elektor Store # 130516-91 [6]

---

Figure 3. The board (not including the screws) can be purchased from Elektor ready-built and tested.

Figure 4. The circuit is built around the FT311D. The 5 V supply is provided over micro USB socket K3.
ble Android device, and so Ton felt it was more appropriate to label it simply ‘USB LED’ in the circuit diagram. The LED flashes when an error occurs; if an Android device that does not support accessory mode is connected, it remains dark. The Elektor Android breakout board can be purchased ready-built and tested [6]. A header is soldered at K2, making it easy to plug the unit into a breadboard. It is a good idea to fit two spacers or bolts to the holes on the opposite side, as shown in Figure 5.

**Test and demonstration**

For initial testing download the demonstration projects from the FTDI website [5]. In each case inside the /bin directory there is a .apk file: this is roughly the Android equivalent of what would be a .exe in the Windows world. Connect the smartphone or tablet to the PC and copy the .apk file into the Android device’s file system, for example into the Download directory. The smartphone or tablet now has to be configured to allow such downloaded applications to be installed. In the case of our Samsung Galaxy Tab this involved ticking the ‘Unknown sources’ box (under ‘Settings’, ‘Security’). Now navigate to the .apk file in the device’s file system (under ‘My Files’, ‘Download’) and tap on it. Normally the Android device will ask if you wish to install the application. If this does not work, one of the installer apps available for free download from the Google Play Store may be able to help.

The best example to try first is the GPIODemo app. Install the app and configure the breakout board for GPIO mode by fitting all three configuration jumpers. Next apply power to the board, either from a PC using a micro USB cable or using a suitable micro USB power adapter, and the power LED should light. Now connect the tablet or smartphone to the board. LED1 should now light,
and simultaneously a dialog box should appear on the Android device asking whether the GPIO-Demo app should be launched automatically in the future whenever the hardware is connected.

When the app is running the user interface appears as shown in Figure 2. The upper area allows you to select whether each GPIO pin is an input or an output: on touching the ‘Config’ button the appropriate command is sent to the breakout board to put the settings into effect. The levels on the outputs can be toggled by touching the white buttons and then touching ‘Write’. You can verify the results using a multimeter or more conveniently by plugging the FT311D board into a breadboard and wiring the outputs to ground or the 3.3 V supply via LEDs with series current-limiting resistors, as shown in the picture at the beginning of this article.

More information on all the demonstration apps and corresponding test circuits can be found in the ‘FT311/FT312D Demo APK User Guide’, which can be downloaded from the FTDI website [5]. In the Elektor Labs we tried out the I²C mode of the device by connecting the I²C communications pins to a GnuBlin relay board [7][8]: see Figure 6. Note that in the demonstration app the field labeled ‘Device Address’ must be filled in with the address of the target I²C device omitting the least-significant bit (which is used to distinguish between read and write commands on the bus).

We also carried out some interesting experiments with two FT311D boards, one configured as an SPI bus master and the other as an SPI bus slave. We connected the two boards and then found we could send bytes from one Android device to another: data sent from the master app when ‘Write’ is touched are immediately displayed by the slave app. And conversely, data entered in the slave app with ‘Write’ will be read into the master app when it is given the ‘Read’ command. We hope that the above has given you a glimpse of the possibilities that this board opens up and we are sure you will think up lots of exciting new projects using it.

Figure 5. A breadboard is the ideal foundation for experimenting.

Figure 6. The GnuBlin relay board can be controlled over its I²C bus.

Web Links
Lathe Tachometer
Feat. Arduino Micro & OLED

By Andre Jordaan
(Switzerland)

Here’s a non-invasive, all electronic add-on to display the running speed (rpm) of your lathe or milling machine, be it CNC or vintage. It’s state of the art—witness the use of an Arduino Micro board and a .96 inch OLED display for instantaneous readout. The little instrument also has a clock displaying the equipment running time.

Attention Bridgeport lathe and milling machine users; get in touch with an electronics geek and have him build this circuit for you.
Attention Arduino users: build this circuit for a nearby Bridgeport lathe owner and get access to his prize machinery.

In this project a reflective IRED / phototransistor device is used as a sensor to produce a retrofit readout of the number of revolutions of a spindle (or “shaft”) on a lathe, a milling machine or a similar piece of equipment in the metalworking toolshop.

Without even scratching the surface of elementary metalworking and finishing, electronicz people should know that the turning speed is crucial for anything that cuts, drills, mills, finishes or polishes metal, wood, glass, and recently, hard plastics, Teflon® and rubber too. In the case of a round steel bar being milled to end size, the turning speed used to do the first couple of rough passes using toolbit ‘A’ at angle x may differ considerably from that applied much later for an ultra-smooth surface finish using toolbit ‘B’ at angle y. In both cases, the shaft speeds have to be set and verified by the operator using...
whatever gears or reduction drives are available on the machine. To the metalworker, the *turning speed* his prized machine is running at equals the *supply volts* and *amps* to the electronics engineer. Get the turning speed (=voltage) wrong and you risk destroying your precious stock material (=64-bit ARM MPU), object being created (=Linux dev system) and/or $50K Bridgeport lathe (=128-bit LeCroy logic analyzer). =You don’t want that to happen.

Where we write spindle or shaft in the remainder of this article of course we mean chuck, 3-claw head, 4-claw head, boring head, end mill, polish pad, drill, tap, *etcetera*, since all of these tools are fitted eventually to the powered shaft or spindle of the lathe or milling machine.

**How it works**

As with most applications of embedded technology the schematic in Figure 1 does not reveal a lot about the workings or aim of the circuit. In fact, you could be looking at NASA’s latest alligator counter for use at Cape Canaveral. One of the reasons for the happy scarcity of components in the schematic is the use of an Arduino Micro board in position MOD(ule)1, resulting in all manner of things being controlled and decided by software rather than discrete parts. The Arduino Micro is first programmed with the project firmware of course via its micro USB connector.

The unregulated supply voltage from a 9-V 6LR22 alkaline battery (or a 7 V to 9 V power adapter) enters the circuit on K1, and gets reduced to 5 volts by stabilizer IC1. LED1 acts as a Power On indicator.

The Adafruit OLED (organic light emitting diode) display module [1] is connected straight onto 8-pin connector SV1. The OLED module with its monochrome, 128 x 64 graphic display gets its 5-volts supply voltage from the same regulator as the Arduino Micro. Other OLED modules may be used but be sure you match their pinout to the board.

The only input device to the circuit, a reflective IRED / phototransistor ("reflective optical sensor" device) like the one at Yourduino [2], is connected on K2. The LED inside the device is again powered from the +5 volts rail via R3. It is on as long as the 5-V supply voltage is present. The phototransistor’s SENS+ pin is pulled to +5 volts by R4, and the signal fed to the INT/D3 line of the Arduino Micro board. A piece of matt black adhesive tape or similar is secured around the lathe’s shaft, leaving a narrow gap (of about 3 mm). Whenever the light emitted by the LED reflects off the shaft at the position of the gap, it illuminates the internal phototransistor, causing the SENS+ line to drop logic Low, and High again when the LED light is not reflected. Hence the speed of the rising edges of SENS+ signal indicates the shaft speed. Note that due to reduction gears being present in the lathe or milling machine the shaft speed is *not* the motor speed.

The Arduino Micro can be reset to force it to start executing its firmware from scratch by pressing S1.

Finally, the function of LED2 on Arduino Micro line D7 is free to assign by the programmer—like OVER REV or CALL KEITH [4].
Software

The program “Firmware.ino” is recommended educational reading if you want to understand all the ins and outs of the control program, and how it got developed and tweaked. The program is richly commended by the author.

Arduino users will know how to transfer the file to their Arduino Micro board, starting from file 130470-1.zip, which can be downloaded free of charge at [3].

The Arduino program waits for a rising edge at its INT1 input and uses five rising edges to compute the average time between them. Using some math the rpm (revolutions per minute) is calculated and sent to the OLED display.

A snippet of the program code is printed in Listing 1. Here we see how the interrupt produced by the sensor gets handled, and the Tachometer’s internal clock is ticking to display the “up” time.

Build it, use it

The printed circuit board designed by Elektor Labs has through-hole parts only and its component overlay is given in Figure 2. Let the electronics people handle the programming, board stuffing, soldering and wiring, and the mech people do the mounting of the board in a case, and the

**Listing 1. Excerpt from Firmware.ino**

```c
//to be done when when sensor is interrupted:
void interrupt_rpm_time()
{
    // current_interrupt_time = micros();
    // if (micros() - previous_interrupt_time > (1/fMax))
    {
        current_interrupt_time = (micros() - previous_interrupt_time);
        previous_interrupt_time = micros();
        // Serial.print("+");
        // Serial.println(current_interrupt_time);
        // digitalWrite(ledPin, HIGH);
    }
}

void calc_run_Time()
{
    //calculating running time
    unsigned long elapsed;
    unsigned long over;
    elapsed=millis();
    h=int(elapsed/3600000);
    over=elapsed%3600000;
    m=int(over/60000);
    over=over%60000;
    s=int(over/1000);
    ms=over%1000;
}
```

Component List

**Resistors**
- R1, R5 = 390Ω 5%, 0.25W
- R2 = 10kΩ 5%, 0.25W
- R3 = 100Ω 5%, 0.25W
- R4 = 22kΩ 5%, 0.25W

**Capacitors**
- C1, C2 = 100nF 10%, 100V, 5mm pitch
- C3 = 10nF 10%, 100V, 5mm pitch

**Semiconductors**
- IC1 = 78L05Z
- LED1 = LED, yellow, 3mm
- LED2 = LED, 3mm, color t.b.d., optional, see text

**Miscellaneous**
- MOD1 = Arduino Micro, Farnell/Newark # 2285194
- S1 = pushbutton, PCB mount, 6x6x9.5mm
- SV1 = 8-way pinheader receptacle
- K2 = 4-way pinheader receptacle
- K1 = 9V battery connector clip + wires
- Reflective Optical Sensor, like TCRT5000
- Monochrome 0.96” 128x64 OLED graphic display, Adafruit # 326 (UG-2864HSWEG01)
- 34-pin DIP socket for Arduino Micro (DIY from SIL pinheader receptacles)
- PCB # 130470-1
should also be easy to add more channels and other controls like relays etc. One wish of the author is send the count pulse on to Mach 3—the control software for his CNC machine.

A slotted optocoupler and a vane fitted at a suitable place on the machine shaft may also be suitable but the assembly is more intricate to mount—if not invasive. If your machine shaft is not reflective, stick a small piece of paper on it to do the light reflection.

The Arduino Micro is plugged on to the rear (solder) side of the Tachometer board (Figure 3); the OLED unit, on to the front (component) side (Figure 4).

After switching on, the Arduino Micro initializes, then displays the Elektor logo and “Elector Electronics” (sic), then “Tachometer by A. Jordaan” and the firmware version number (Figures 5a; 5b). Next, the current rpm value is displayed, and the ‘up’ time of the tachometer (so you can bill your customers and come across serious).

Over to you
The project being open in terms of hardware and software, there should be nothing in your way to do custom adaptations like getting that k right. It will be insightful for sure. The hardware for example may be adapted to support different types of sensor types like Hall effect. Other possible applications include a step counter, a rev counter for model planes, a PWM percentage meter, or a simple event driven animation on the display. It

Web Links
In these times of home automation and smart electrical installations, the big manufacturers are vying with each other in ingenuity. It seems to be a ferocious fight, judging by the proliferation of models of wall switch on offer, from the simple white changeover switch to automatic systems of controlling lighting for any home by phone or touch tablet—and not forgetting the silent, round, square versions you can customize to your heart’s content. I’d always dreamt of bringing my home electrical installation up to date using

**Touch-2-Switch**

Versatile capacitive switch in a wall box

This capacitive-detection dual wall touch-switch, designed to fit into domestic electric wall-boxes, uses analog technology throughout. A single multi-purpose version that you can configure as you wish to function as a simple on/off switch, a 2-way switch, or momentary button, on two independent channels. At low cost, modernizing your home electrics has never been so tempting.

**Specifications**

- changeover, 2-way, or momentary switch
- replaces a standard mechanical switch in a wall box
- choice of controlling one or two separate lighting circuits
- configurable illuminated indicator
- modern appearance, easy to reproduce, 230 V/50 Hz or 120 V/60 Hz
- no programmed devices
- configurable using just a soldering iron

By David Ardouin (France) [2]
touch switches, but couldn’t quite bring myself to spend a fortune on this sort of accessory. So that’s why I decided to design one that would exactly meet my requirements. The new touch switch will:

- replace a mechanical switch and fit into a standard flush-mounting wall-box, attractive and modern in appearance, yet easy to reproduce;

- give the choice of controlling one or two separate lighting circuits and operate as a simple changeover switch, a 2-way switch, or as a momentary push-button;

- have an indicator configurable as either a constantly-illuminated pilot light or as an output state indicator.

Look as hard as you like, even with the major names in this field, but you won’t find any product with these specifications.

I must admit, however, that changing to an electronic version inevitably implies the switch is going to consume some power all the time—but I’ve kept this to the strict minimum. It does also require the presence of a Neutral (cold) wire—which may not be present in lighting circuit boxes.

**Tips**

I suggest exploring the circuit diagram (Figures 1a and 1b) by following the line of thinking that led me to this solution. The first crucial choice was picking a technique for switching a power of a hundred or so watts. Exit the triac because of instability at low loads and a monstrous consumption of several tens of milliamps. Using a MOSFET transistor and diode bridge promised minimal losses, but though it worked fine for a single channel, it caused me unacceptable problems running two channels in parallel. What’s more, the diode bridge would mean the low-voltage supply could no longer be achieved using a classic capacitive dropper; it would have been necessary to revert to a power resistor with deplorable efficiency. No question either of using a relay, because of its holding current. A conventional relay, no—but maybe a bistable relay? Much less common than their monostable counterparts, latching relays use power only at the moment of changing state. A pulse of just a few

---

**SAFETY**

As there’s no transformer, this sort of power supply has the advantage of being very compact—but it does not provide voltage isolation. As the internal ground of the circuit is directly connected to the Live (hot), it is strictly forbidden to handle the circuits, including the capacitive part and the electrodes, when the supply voltage is present. User safety is only ensured once the switch is enclosed within its back box. For testing and for any handling in the presence of hazardous voltage, get into the good habit of disconnecting the circuit to be tested before connecting your measuring instruments to it, then powering it back up to read the measured value, and lastly, disconnecting the power again before moving on to the next step.

For the testing, I recommend making up an isolating transformer from two 230 V/12 V (US: 115 V / 12 V) transformers rated at a few VA wired back-to-back. A dangerous voltage will still be present on the output, but only across the two end terminals. This accessory will make it easier and safer for you to check the circuit is working properly. You will be able to switch the relays, but not connect up an actual load.
Projects

supply the current needed by the relays. These devices are normally used to drive MOSFET gates, but they fulfill their role marvelously here for a derisory cost.

The AC powerline supply (J1) makes use of the impedance of a capacitor (C5) to drop the voltage to the required value, with minimal losses. The choice of this component depends on what powerline voltage you have; choose the 150 nF for 115-120 V/60 Hz or 100 nF if your domestic grid is 230 V/50 Hz. A rectifying diode (D1) and zener (D2) then provide regulation to 12 V. Unfortunately, this kind of circuit always draws tens of milliseconds is enough to make them switch over; then they don’t consume anything at all until the next time they switch. Perfect!

In the 12-V version, relays K1 and K2 (Figure 1b) draw a current of 18 mA for 20 ms when switching. A “luxury” solution would have been to drive the coil via an H-bridge. But I preferred to store the energy needed for switching in capacitors C6 and C14, which provide the polarity inversion depending on the sense of the edge applied by U5 and U6.

These latter, helped by Q1 and Q2, form level-shifters to amplify the drive signal voltage and supply the current needed by the relays. These devices are normally used to drive MOSFET gates, but they fulfill their role marvelously here for a derisory cost.

The AC powerline supply (J1) makes use of the impedance of a capacitor (C5) to drop the voltage to the required value, with minimal losses. The choice of this component depends on what powerline voltage you have; choose the 150 nF for 115-120 V/60 Hz or 100 nF if your domestic grid is 230 V/50 Hz. A rectifying diode (D1) and zener (D2) then provide regulation to 12 V. Unfortunately, this kind of circuit always draws

Figure 1a & 1b:
Circuit diagram in two sections: right, the high-voltage circuit; left, the low-voltage circuit. Unless electrically isolated from the power line supply, all the conducting parts of this circuit are live! Despite their high values, C2 and C3 are non-polarized capacitors.

Guaranteed microcontroller-free!
The capacitive detection is provided by two half-moon-shaped electrodes, positioned behind the switch front panel and connected to J7-1 and J8-1. U2 and U3 are specialized ICs for capacitive detection. Although at first sight deceptively simple, in fact they include several sophisticated algorithms that are essential for guaranteeing faultless detection over time. Their outputs have one specific feature that gave me a bit of a headache: in order to confirm the device is working correctly, regardless of its logic state, the output regularly goes briefly high-impedance, which produces pulses of erratic logic levels. In order to filter these out, I've added C11 and C15, the values for which are not arbitrary; they are just enough to filter the pulses without degrading the rise-time too much, as that would hinder proper switching of the flip-flops in U4. Capacitors C9 and C13 directly define the reactivity of the capacitive switch.
Projects

button. To make the detection more sensitive, you can increase their value up to 10 nF; on the other hand, if you experience unwanted triggering, you can reduce the value down to 1 nF. Diodes D5 and D6 form a logic OR which, depending on the configuration of solder bridge S4, will allow the outputs from both capacitive detectors to be combined so as to form a single electrode driving just the first output. A solder bridge between its pads 2 and 3 will activate the “single switch” mode, while bridging 1 and 2 give you two independent channels. In the same way, S1 and S3 let you choose between the momentary contact operating mode (bridge across 1 and 2 for S1 and across 2 and 3 for S3) or latching contact mode at the flip-flop output (bridge across 2 and 3 for S1 and 1 and 2 for S3). The first mode is useful for an installation where there are a number of buttons all operating a remote-controlled switch or timer in the electrical distribution board. The drive signals then go on to the power board via connectors J3 and J5.

Touch switches no longer give the mechanical feedback of their predecessors, which is why it’s worth using illuminated indicators to confirm the action requested. For a single switch, D7 only will be fitted in the middle position, while with two separate channels D8 and D9 will be used. Capacitors C17 and C18 are recommended [1] in the event that the electrodes detect impedance changes (of the MOSFET transistors Q5 and Q7, as it happens) that might disrupt their operation. Bridges S5–S10 define the behavior of the LEDs with a choice of permanent pilot light, touch confirm, or copying the output status (see Table). To drop the 12 V voltage present on J4, I chose a step-down regulator (U1) specially designed for low powers. With its associated components, it supplies a maximum current of 10 mA @ 3 V — plenty to power this little world.

Soldering iron in hand...

There’s a downside to the miniaturization of surface-mount components: they’re jolly small between our big fingers. So I used 1206 format passive components, a little easier to solder. The three PCBs are supplied as one board (Figure 2).

---

**Figure 2.**
The dual switch assembled.

---

<table>
<thead>
<tr>
<th>Channel 1:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S3</td>
<td>1-2 momentary mode</td>
</tr>
<tr>
<td>S3 bridged; S6 &amp; S7 open</td>
<td>indicator 1 always lit (pilot light)</td>
</tr>
<tr>
<td>S6 bridged; S5 &amp; S7 open</td>
<td>indicator 1 lit while pressing</td>
</tr>
<tr>
<td>S7 bridged; S5 &amp; S6 open</td>
<td>indicator 1 follows output state</td>
</tr>
<tr>
<td>S5, S6, &amp; S7 open</td>
<td>indicator 1 disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Channel 2:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1-2 two-position switch mode</td>
</tr>
<tr>
<td>S4</td>
<td>2-3 momentary mode</td>
</tr>
<tr>
<td>S4 bridged; S8 &amp; S9 open</td>
<td>indicator 2 always lit (standby light)</td>
</tr>
<tr>
<td>S9 bridged; S8 &amp; S10 open</td>
<td>indicator 2 lit while pressing</td>
</tr>
<tr>
<td>S10 bridged; S8 &amp; S9 open</td>
<td>indicator 2 follows output state</td>
</tr>
<tr>
<td>S8, S9, &amp; S10 open</td>
<td>indicator 2 disabled</td>
</tr>
</tbody>
</table>
Final assembly

In order to give a professional finish and integrate perfectly into an existing electrical system, the electronics are preferably built in to wall box normally used to hide an electrical junction. In order to match my own installation at home in France, I chose the Odace range from Schneider Electrics. The fitting consists of three parts: the light grey mount that fixes to the wall-box, the white circular blanking plate, and the trim, which comes in different colors. To mount the electronics, the first step is to cut off the two lugs at the rear of the mount (Figure 3). A junior hacksaw or mini cutting disk will be perfect for this job.

The first PCB, the electrode one, is going to be glued into the circular blanking plate (Figure 4). Apply a thin, even layer of epoxy adhesive all over the copper side of this board. I must emphasize it's vital that the entire surface must be evenly covered to avoid any air pockets that might disrupt the capacitive detection. You can then insert the glued-up PCB into the blanking plate, taking care to align the two parts as accurately as possible. Depending on the number of indicators installed (D7, D8, D9), drill holes in the blanking plate using a 3 mm drill and fit one or two light-guides (Figure 4). To finish this part, finally solder two 15 mm long rigid wires (e.g. resistor lead offcuts) to the pads J7-2 and J8-2.

The detection board is assembled behind the grey mounting plate using four spots of epoxy adhesive in the corners (Figure 5), after having carefully inserted the two electrode connection leads into the corresponding pads. Before completing the gluing, check that the LEDs are in line with the blanking plate light-guides—it's all too easy to get it the wrong way round. Lastly, solder pads J7-1 and J8-1. All that remains to complete the assembly is to plug the power part (Figure 6) onto the connectors provided. The cross-section view (Figure 7) shows the details of the stacked boards.

The connections to be made for configuring as 2-way switches or momentary buttons are given in Figure 8a. Very often with most mechanical switches, you will find in your back box only a a dodge to reduce manufacturing costs (at the expense of a few hours of extra work on routing). It’ll be easier if you keep them together during the soldering stages. Start by soldering all the surface-mount components on the top side, then continue with the SMDs on the underside. If you choose to use only a single way of your switch, fit only LED D7. For two separate channels, fit D8 and D9, and not D7. On the assembly drawing, the rectangle in the corner of the LED symbols indicates the position of their cathodes. Then fit the through-hole components on the power circuit (130272-3). The male and female headers that form J3–J6 will be soldered later, during assembly. The capacitive detection circuit doesn’t require any soldering for the moment, as the two copper electrodes are the sole components.

Checking

Temporarily link across J7-1 and J7-2 using a short length of flexible hook-up wire. Do the same with J8-1 and J8-2. Using a 9 V battery or a bench supply set to this voltage, power the logic section via connector J4-2, taking care to respect the polarity shown on the copper. If everything is OK, you ought to measure a voltage of 3 V across C3. Place your finger on the insulating side of one of the electrodes, the LED for the relevant channel should light at once; the capacitive detection here always takes place through an insulating surface – do not touch the copper electrodes directly, operation will be erratic.

On the “power” side, connect a voltmeter to the pins of J4-1 then power the circuit by applying the AC power supply line voltage to terminal block J1. Never forget that the whole of the circuit is now at a lethal voltage, and from now on you must not touch it. Check you have a voltage close to 12 V on J4-1.

Now separate the three circuits using cutters, using a small, flat file to smooth the cut edges. Insert J3 to J6 in the positions provided, but for the moment solder only one pin on each of the headers. Stack the two “driver” and “power” circuits together and check the connectors line up properly. Touch up your soldering if necessary, then finalize it all by soldering the remaining pins.

100% soldering iron configurable
Live (hot) wire (check the color used in your country) so you’ll also have to run a Neutral (cold) wire (check color in your country) which you’ll be able to pick up from the nearest socket outlet, for example. If you should inadvertently reverse the Phase and Neutral connections on J1, the touch part, the LEDs, and the relays would all work normally, but the loads being controlled wouldn’t light up. Check the wiring again carefully before closing up your installation.

The markings R and O refer to the French identification of the relay contacts translating to Rest

---

**Component List**

**Resistors (SMD 1206, 5%)**
- R1, R2 = 47Ω 0.25W
- R3, R9, R11, R12, R29, R30 = 470kΩ
- R4, R19, R24 = 120kΩ
- R5 = 330kΩ
- R6, R7 = 2.2MΩ
- R10, R13, R27, R28 = 10kΩ
- R20, R22, R23, R25, R26, R31, R32, R33 = 1kΩ

**Capacitors (ceramic, SMD1206, 20%)**
- C2, C3 = 4.7µF 16V
- C4 = 22pF 50V
- C7, C8, C10, C12, C16, C17, C18 = 100nF 50V
- C9, C13 = 4.7nF 50V X7R
- C11, C15 = 1nF 50V

**Capacitors (miscellaneous)**
- C1 = 470µF 25V, Panasonic EEEFK1E471AP
- C5 = 100nF 305V, X2 Class, 15mm pitch, Epcos B32922C3104M
- C6, C14 = 100µF 25V, Panasonic EEEFK1E101AP
- C8 = 33µF 16V, 1244419
- C16, C17 = 100µF 25V, Panasonic EEEFK1E101AP
- C18 = 47µF 25V, Panasonic EEEFK1E471AP

**Inductors**
- L1 = 22µH 0.11A, Multicomp MCFT000190
- L2 = 18µH 0.11A, Multicomp MCFT000190

**Diodes**
- D1, D5, D6 = BAT54
- D2 = BZX84C12
- D7, D8, D9 = LED, yellow, Kingbright KPT-3216SYCK

**Semiconductors**
- Q1, Q2, Q5, Q7 = 2N7002ET1G
- U1 = TPS62120DCNT
- U2, U3 = AT42QT1010-TSHR
- U4 = SN74HC74D
- U5, U6 = ZXGD3005E6TA

**Miscellaneous**
- F1-1 = fuse holder, Littelfuse 64900001039
- F1-2 = fuse, 5A slow, 5x20mm, Schurter 0034.3124
- J1 = 2-way PCB spring terminal block, 5.0mm pitch, Wurth 6914147200002
- J2 = 4-way PCB spring terminal block, 5.0mm pitch, Wurth 6914147200004
- J3 = 10-way receptacle, low profile, 0.1" pitch (1668506)
- J3 = 10-way receptacle, low profile, 0.1" pitch (1668506)
- K1, K2 = relay, bistable, 12V, TE VEO14F12
- R8 = varistor, 300V, Epcos B72205S0271K101
- GL1, GL2, GL3 = LED light pipe, Bivar PLP2-375
- PCBs 130272-1, 130272-2, 130272-3

(numbers in round brackets are Farnell/Newark order codes)
and Operate. If the state of the LED indicator is reversed with respect to the load connected, all you have to do is swap the connection over between these R and O terminals. **Figure 8b** shows how to wire it up as a 2-way switch with other mechanical switches or IMPACT switches. As the configuration of each of the two channels is independent of the other, you can even mix the two types of wiring on the same switch. Lastly, in the case of a configuration as a single channel, the connection will be made only to the terminals O1 and R2. If you’re feeling adventurous, don’t hesitate to send me photos of your own projects [2]!

**Web Links**


[2] ardouin.david.projects@gmail.com
Seismic Detector
Using a piezoelectric sensor

The author developed this circuit in response to the many earthquakes that have occurred in the Groningen area in the Netherlands in recent years as a result of 50 years of gas extraction in this province. It can also be used to detect passing trucks or other sources of vibration. The sensitivity is adjustable to allow the detector to be used under various conditions.

Along with an LED and a buzzer for the visible and audible indications, there is a relay that can be used to energize other circuits or devices. A small 433-MHz (or 868-MHz) LPR (transmitter) could also be connected to the relay output for wireless transmission of the detector signal to a remote location. A simple transmitter/receiver combination suitable for this purpose is also described in the article ‘Wireless Signaling’ in this edition of Elektor. Among other things, this arrangement allows the detector and transmitter to be mounted in a location that is not easily accessible.

Operation
The key component of this compact circuit is the vibration sensor (Figure 1). The LDTM-028K sensor from Measurement Specialties used here consists of a thin piezoelectric plastic film sealed in plastic, which has two solder pins at one end for fitting to a PCB and a threaded bush at the other end for attaching a pendulum or a weight. When the sensor is moved back and forth, for example by vibrations, the piezoelectric film generates a voltage that depends on the degree of bending and can be as high as 70 V or so.

With very small motions (e.g. light vibrations) the signal level from the sensor is only a few millivolts, so a lot of amplification is necessary if you want to use it to switch something. For this purpose, the sensor is connected to the input of IC1, a type MCP601 opamp from Microchip (see the schematic diagram in Figure 2). This single-supply opamp with rail-to-rail output can operate with a supply voltage as low as 2.7 V. The gain of this opamp stage can be adjusted over a
A wide range with trimpot P1. Diodes D1 and D2 on the non-inverting input of IC1 protect it against excessive voltage from the sensor.

The opamp output is low under quiescent conditions. The output is connected to a pulse stretcher built around IC2a–IC2d. The output of the pulse stretcher is low when the opamp output is low; C1 is discharged and the outputs of IC2a, IC2b and IC2d are high. When the vibration sensor generates a pulse, the output of IC2c goes high, and it is kept high by IC2b until C1 has charged enough to cause the output of IC2a to go low.

When the voltage on C1 reaches the necessary level, pin 12 of IC2d is pulled low by R3 and the output of IC2d goes high. The output of IC2b also goes high, and the pulse stretcher returns to its initial state. The next pulse from the vibration sensor can be processed after C1 has discharged again.

Diode D3 and resistor R3 ensure that on pin 12 of IC2d, a high level at the output of IC2a overrides the signal level at the output of the pulse stretcher (pin 8 of IC2c). The pulse length of the pulse stretcher is approximately 20 seconds.

**Component List**

<table>
<thead>
<tr>
<th>Resistors</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>82kΩ</td>
</tr>
<tr>
<td>R2</td>
<td>470kΩ</td>
</tr>
<tr>
<td>R3</td>
<td>100kΩ</td>
</tr>
<tr>
<td>R4</td>
<td>1kΩ</td>
</tr>
<tr>
<td>R5</td>
<td>100Ω</td>
</tr>
<tr>
<td>P1</td>
<td>100kΩ trimpot</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacitors</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>4.7µF 16V</td>
</tr>
<tr>
<td>C2</td>
<td>100nF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semiconductors</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1, D2, D3, D4</td>
<td>1N4148</td>
</tr>
<tr>
<td>LED1</td>
<td>LED, low-current, red, 3mm</td>
</tr>
<tr>
<td>T1</td>
<td>BC547</td>
</tr>
<tr>
<td>IC1</td>
<td>MCP601 (Microchip)</td>
</tr>
<tr>
<td>IC2</td>
<td>74HC132</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Miscellaneous</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BZ1</td>
<td>Active piezo buzzer, e.g. Multicomp MCKPT-G1210-3916</td>
</tr>
<tr>
<td>S1</td>
<td>Pushbutton with break contact</td>
</tr>
<tr>
<td>RE1</td>
<td>PCB relay, 3V (e.g. Omron G5V2H13DC)</td>
</tr>
<tr>
<td>Sensor</td>
<td>LDTM-028K (Measurement Specialties)</td>
</tr>
<tr>
<td>PCB</td>
<td>130517-1 [1]</td>
</tr>
</tbody>
</table>
Some relays (such as the AZ822 from American Zettler) have these pins reversed. There's not much that needs to be said about fitting the leaded components, which shouldn't cause any problems. If desired, you can use sockets for IC1 and IC2. The sensor is soldered directly to the PCB, and a metal strip acting as a pendulum can be attached later. It's a good idea to place the two AA cells for the power source in a battery holder, which can be fitted together with the PCB in a plastic case. When fitting the board in the case, bear in mind that it will be mounted upside down in use and the metal strip needs to be able to pass through the case and move freely. The unit can be mounted on the ceiling of the room you want to monitor, or on a wall using an angle bracket.

The sensor can also be fitted on the underside of the board if this is more convenient. Adjustment: After mounting the unit at the location to be monitored, adjust the sensitivity by turning trimpot P1 to the point where the buzzer just stops sounding. It's a good idea to experiment with this a bit so that the alarm does not go off every time a truck drives by. You can also adjust the length of the metal strip attached to the sensor as necessary. The sensor and metal strip can also be housed in a plastic pipe (with a diameter of 10 cm [4 inches] or so) for protection, mounted so that the strip is free to move. This reduces the effects of wind and air currents on the sensor.

**Assembly and use**

Although the circuit only consists of a few components, a PCB layout (Figure 3) has been designed for it to make assembly easier. As usual, the layout can be downloaded free of charge [1]. Two pads for the relay contacts are joined together on the PCB. This may appear a bit strange, but it provides more freedom in the choice of relay. Usually the pin for the common contact is in the middle (which is true for the type in the components list) and the pin closer to the relay coil is for the NC contact, but with the stated component values. It can be adjusted by changing the value of C1 (a larger value results in a longer time).

The pulse stretcher output drives an LED via series resistor R5 and a transistor. The transistor in turn drives a mini buzzer and a relay connected in parallel. A larger alarm indicator or the mini transmitter described elsewhere in this edition (see "Wireless Signaling") can be connected to the normally-open relay contact. Although the output stage may appear somewhat overendowed, with an LED indicator (R5/LED1), a buzzer (BZ1) and a relay (RE1), you can easily omit one or two of these options by simply not fitting the components concerned on the board. The entire circuit operates from a supply voltage of just 3 V and the quiescent current consumption is only 50 µA, so a pair of AA alkaline cells is sufficient to power the circuit for many years. A pushbutton switch with a normally closed contact is included in the supply voltage line. This can be used to cancel the alarm indication before the end of the stretched pulse period—a simple but effective solution.

**Web Link**

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

DLP-USB1232H: USB 2.0 UART/FIFO

**HIGH-SPEED**

480Mb/s

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

DLP-IOR4
4-Channel Relay Cable

DLP-TH1b
Temp/Humidity Cable

DLP-RFID1
HF RFID Reader/Writer

DLP-FPGA
USB-to-Xilinx FPGA Module

**www.dlpdesign.com**

SCOPE DEALS

PASSPORT-SIZE PC SCOPES

Great scopes for field use with laptops.
Up to 200MHz bandwidth with 1Gsa/s, high speed data streaming to 1Msa/s, build-in 16Sa/s AVG/wfrm gen. - PS2800A $239+

30MHz SCOPE

Remarkable 30MHz, 2-ch 250Ms/s sample rate oscilloscope, 8-in color TFT-LCD and Autoscale function, includes FREE carry case + 3 yr warranty - SDS032E $289

60MHz SCOPE

Best selling 60MHz 2-ch scope with 500Ms/s rate + huge 10Msa memory & 8" color TFT-LCD. Includes FREE carry case - SDS051 $349

100MHz SCOPE

High-end 100MHz 2-ch 1Gs/s benchscope with 11Msa memory and USB port + FREE scope case. New super low price - Ds1012E $399

NEW!

70-300MHz SCOPES

Fast, versatile 2-ch 250Ms/s scopes with 8" WVGA LCD, integrated generator, 14Mpa memory, very low noise floor. - Ds2000A series $839+

INCREDiBLE LOW PRICES, FREE TECHNICAL SUPPORT

GREAT CUSTOMER SERVICE

Saelig.com

Take out a FREE membership to Elektor.POST

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

Elektor

Register today at www.elektor.com/newsletter

Silicon based life forms of the world, united!

When you receive this Master POST, you are already one of the many fine members of a unique circle of people who, with their
findings, share ideas and enthusiasm, keep our electronic hobby alive. It is through our readers, the people and the
people, that Elektor can do what it does. Through the posts, the photos, the ideas, the hints, the tips and the
members, important people and ideas can be shared.

It is important to me that the reader continues to work with the ideas and "taste" and allows the best people to be
shared in the magazine. I hope you will enjoy the fun we are having in preparing and publishing this selection of
ideas that we hope will be of interest to you and your friends.

The people, the ideas, the photos, the articles, the hints, and all of the content that goes into this magazine, is
the result of the work of many people who are dedicated to bringing you the very best in electronics.

The people, the ideas, the photos, the articles, the hints, and all of the content that goes into this magazine, is
the result of the work of many people who are dedicated to bringing you the very best in electronics.
Although most PCs nowadays have dispensed with the traditional serial interface, it is still widely used in the microcontroller world. This interface was originally developed to transfer serial data over a conductor. The individual bits obediently travel down the wire one after the other. The bit timing is precisely defined so that the receiver can make sense of the serial bit stream. A commonly used data rate is 9600 bits per second, which is also called 9600 baud. In addition to the eight data bits of each byte, there is a start bit and one or two stop bits. Each data bit is present on the wire for an interval of approximately 100 µs (at 9600 baud).

Print output

Every ATmega microcontroller has a serial interface (UART) with RXD (PD0) and TXD (PD1) lines. These signals are TTL compatible, which means that the signal level is 5 V in the quiescent state and 0 V in the active state. By contrast, a COM port on a PC operates with RS232 signal levels. They are -12 V in the quiescent state and +12 V in the active state. For this reason, a peripheral interface IC such as the MAX232 is often necessary to invert and adapt the signal levels.

Instead of this, the Arduino board has a serial to USB converter that sends the data over USB. This means that you need a driver for a virtual serial port (such as COM2) on the PC. For programs running on the PC, that makes the data transported over USB look like it entered the PC through a conventional serial port. This data can therefore be displayed using a standard terminal emulator program.

We recommend that you use the terminal emulator program integrated into Bascom. You can open it under Tools > Terminal Emulator or with the key combination Ctrl-T, but you have to configure the right settings to make it work properly. In particular this means selecting the right COM port (the same one you configured for the boot loader) and setting the right baud rate (in this case 9600). For the rest you can use the default settings: 8 data bits, 1 stop bit and no parity (Figure 1).

Now you have to persuade the Arduino Uno to transmit something. That's easy in Bascom, thanks to the Print command. See Listing 1 for an example of how to use the Print command (note that all code files can be downloaded at [1]). This command can be used to send output to a printer,
which explains its name. However, to avoid wasting paper it’s better to send text and data to the monitor. The result is shown in Figure 2.

There are two Print commands in the program. The second one sends a message in quotation marks: “Uno”. But first a number is sent – more particularly, the number N. If it’s been a while since you had anything to do with math, you may still have a vague memory that unknown quantities in algebra are often designated as x. You could do the same in Bascom, but in situations where numbers are simply incremented (1, 2, 3 and so on), the symbols N, M, I and J are commonly used in source code. However, you are free to choose your own symbols for variables of this sort. The only thing that really matters is that you first tell Bascom what type of number each variable represents. This is because Bascom has to know how many bytes of memory the number needs and how computations should be performed with the number in the code. For this reason, the variables must be declared (dimensioned) before they are used. That’s the purpose of the following instruction: Dim N as Byte. This means that N is a number of type Byte, which clearly indicates that this number consists of eight bits (recall the previous instalment) and has a number range from 0 to 255.

Assignments

Now you can perform calculations with the number variable N. The variable is assigned the value 0 at the start of program execution. Although you might think this is only logical, it is actually a particular feature of Basic. In other languages you have to write ‘N = 0’, but in Bascom this is done automatically. The value of N is subsequently incremented by 1 each time the loop is executed. My old math teacher would have boxed my ears if I wrote ‘N = N + 1’, because it is not valid mathematical equation. However, in many programming languages the equal sign is used as an assignment operator. The intention here is that the variable N is assigned a new value by adding 1 to the existing value.

You can see the results of this ongoing incrementing in the terminal window: the value of N increases by 1 after each output. You can also see something else: the value stops rising after it reaches 255, since we defined N as a single-byte number. This means that 255 + 1 is not 256, but instead 0. This is called number overflow. Here this is not a problem, but in other places it can be a problem – namely when you have inadvertently chosen a number type that is too small and the overflow is not detected.

There’s another detail that should be noted here: why does the display always go to a new line after “Uno”, but not after the number? The answer is
Projects

After all, our original intention here was to display measurements from the A/D converter, and now we’re ready to do so.

The serial bits output by the Atmega328, which are sent to the serial to USB converter on the Uno board, can easily be seen with an oscilloscope. Simply touch the scope probe to the TX pin of the microcontroller, which is routed to the socket header at the top right on the Arduino board. Then you can see the data stream. The RX pin is next to the TX pin, and if you type some characters on the terminal you can see them on the RX line. This shows that the PC can also send data to the microcontroller. At this point the received data does not do anything because the program ignores it.

The A/D converter

The analog to digital converter (ADC) in the microcontroller is a sort of measuring device. As we all know, measuring involves comparing. First you need some sort of unit to serve as a reference quantity. Before you can measure a distance in meters, you have to know how long a meter is. Likewise, you have to know how big a volt is before you can say how many volts are lurking in your wall outlet. The situation here is similar. The microcontroller needs a reference voltage so that it can compare the voltage on a selected analog input (A0 to A5 on the Uno board) to the reference voltage. The reference voltage is connected to the AREF pin. However, you can also use a register to select the reference voltage. For example, you could apply an external reference voltage of 1 V or connect a 5 V reference voltage from inside the microcontroller, but of course you shouldn’t do both at the same time.

Now try measuring the voltage between AREF and GND (see \textbf{Figure 3}). Since the A/D converter hasn’t been configured yet, the voltage is zero. If you load the program in \textbf{Listing 2} into the microcontroller and make this measurement again, you will see a voltage of 5 V. This is due to the following configuration statement:

\begin{verbatim}
Config Adc = Single, Prescaler = 64, Reference = Avcc '5 V
\end{verbatim}

This initializes the ADC, which means it enables the ADC with specific attributes. Here Single mode means that only one measurement is made for each request. The other option is Free (short for free-running mode), in which measurements

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{Using the A/D converter.}
\end{figure}

\begin{verbatim}
Listing 2. Using the A/D converter
'----------------------------------------
'UNO_AD1.BAS  ADC
'----------------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim D As Word

Config Adc = Single, Prescaler = 64, Reference = Avcc '5 V

Do
    D = Getadc(0)
    Print D
    Waitms 200
Loop
\end{verbatim}
are made repeatedly without interruption. The prescaler generates a clock signal for the A/D converter by dividing the processor clock signal, which is 16 MHz on the Uno board. In this case the A/D converter runs at 250 kHz (16 MHz divided by 64). It can also run faster, but then it is not quite as accurate. The setting ‘Reference = Avcc’ connects the Aref pin to Avcc, which the supply voltage of approximately 5 V. Incidentally, it’s worthwhile taking a look at the data sheet for the microcontroller – the section on the A/D converter is practically a data sheet in its own right.

If after that you still don’t have any idea of what settings you can or should use, the Bascom Help is a good source of information. Simply point the mouse to ‘Config’ and press the F1 key, and you will be rewarded with all sorts of useful information and a bit of sample source code. The ‘instruction D = Getadc(0)’ initiates a measurement of the voltage on A0 and copies the result to the variable D. The A/D converter provides readings with a resolution of 10 bits, which corresponds to a number range of 0 to 1023. The variable D must therefore never be dimensioned as a byte variable. You should use either the type Word (range 0 to 65535) or the type Integer (range -32768 to +32767) to ensure that the readings will fit.

The measurements are made using a successive approximation algorithm with ten steps. First the converter compares the input voltage to half the reference voltage. If it is higher than the reference voltage, the next comparison is to three-quarters of the reference voltage. If it is still higher, the next comparison is seven-eighths, and so on. The resolution doubles with each step. At the end you have number that indicates which step on a ladder with 1024 steps between zero and the reference voltage corresponds to the measured voltage. Each step is approximately 5 mV (5 V divided by 1023) if you use a reference voltage of 5 V. For comparison, the resolution of a digital voltmeter with a 3-½ digit display is roughly twice as good (2000 steps). This makes the microcontroller a fairly good measuring device, although you shouldn’t confuse resolution with accuracy – the accuracy can be seriously compromised by an inaccurate reference voltage.

The analog inputs are subject to the same considerations as the digital inputs: an input voltage well below zero or well above the supply voltage can lead to undesirable side effects, extending as far as latchup (as described in the previous part of this series). However, here again a 10-kΩ resistor in series with the analog input can prevent serious problems. With that precaution in place, it’s even okay to touch the input with your finger. If you do so, you will see varying readings in the terminal window (Figure 4), including many limit values (0 and 1023) because the hum voltage exceeds the measuring range. The overall effect is similar to the waveform from a half-wave rectifier as seen on a oscilloscope. If you compare the measurement cycle time, consisting of the programmed wait interval of 200 ms plus the actual measurement time, with the AC line voltage period of 20 ms (at 50 Hz), you can see that here again there is a strobscope effect. This is called undersampling, and it produces a low-frequency alias signal because the measurement samples do not all lie within the same period of the measured signal. In fact, there are several periods of the input signal between each pair of samples.

**A bit of math**

Our first simple measurement program only generates raw data, so you have to calculate the signal voltage yourself. Consider the following example: You connect analog input A0 to the 3.3 V output of the Arduino board. Now you see a reading of 676, or something close to that. You grab your pocket calculator and calculate: 676 ÷ 1023 × 5 V = 3.305 V (okay). Another option is to modify the program and have the microcontroller do the math (Listing 3). For this you need a variable that can hold more than just integer values. You can use a variable U of type Single (which means a single-precision floating point / real number variable), which is fully suf-
off readings of this sort, but for the time being it’s nothing to worry about.

At this point you can see that it makes a difference whether you power the Uno board over USB or from an external power supply. The supply voltage on the USB port can easily differ from 5 V by 10% or more (see Figure 5), and the accuracy of the measurement results will vary accordingly. By contrast, the 5 V regulator on the Uno board has a relatively narrow tolerance, typically around 1%. Try both options for yourself. Also measure the actual voltage on the 3.3 V terminal and the voltage on the AREF pin in each case. Armed with that information, you can significantly improve the accuracy. Suppose you measure a reference voltage of 5.03 V with an external power supply (using a high-precision DVM). Enter this value in the program in place of the nominal value of 5.0 V, and your readings will be nearly spot on.

You should also try other reference settings. If you configure ‘Reference = Aref’, at first you do not have any reference voltage at all. You have to provide your own reference voltage, with a free choice anywhere in the range of 0 to 5 V. For example, you can connect the Aref pin to the 3.3 V terminal of the Uno board. In this case the measuring range extends to 3.3 V. Another option is ‘Reference = Internal’. In that case the A/D converter uses an internal 1.1 V reference voltage, and you can measure this voltage on the Aref pin. The data sheet very modestly states a tolerance of approximately 10%, which means 1.0 V to 1.2 V. In our case we measured 1.08 V, which is only 2% away from the nominal value. However, you should measure the actual value yourself. There are two things worth noting about this relatively low reference voltage. The first is that the accuracy is independent of the supply voltage; and the other is that it gives you a relatively high resolution of approximately 1 mV.

Measuring the microcontroller’s own supply voltage
You might wonder whether the microcontroller can measure its own supply voltage. At first the answer appears to be no, since you would need a sufficiently large reference voltage that is independent of the supply voltage. However, it turns out that this is indeed possible because the data sheet reveals several interesting details about the A/D converter. The input multiplexer (an analog switch similar to the well-known 4051) has

Listing 3. Conversion to volts
```
'----------------------------------------
'UNO_AD2.BAS    0...5 V
'----------------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim D As Word
Dim U As Single

Config Adc = Single , Prescaler = 64 , Reference = Avcc '5 V

Do
  D = Getadc(0)
  U = D * 5.0
  U = U / 1023
  Print U ; " V"
  Waitms 200
Loop
```
External programmer

When you want to use a microcontroller, you usually need some sort of programming device. This is not necessary with the Arduino because it has a built-in boot loader, as described in the first part of this series. However, you do not have a boot loader when you buy a new AVR microcontroller and fit it on your own PCB, so you have to use a programmer. One low-cost option is the Atmel ISP mkII. It can be used with the free AVR Studio 6 development environment. However, there are lots of other programmers that can do the same job.

A six-pin ISP connector is normally used for in-system programming (ISP) of AVR microcontrollers. Many microcontroller boards, including the Arduino, have a suitable 6-pin header for this. The SPI interface, which consists of serial data streams (MOSI to the microcontroller and MISO back to the programmer) and a clock signal (SCK), is often used to transmit the programming data. This interface will be described in more detail in a later instalment. The pinout of the ATmega328 is:

1. MISO, PB4
2. Vcc
3. SCK, PB5
4. MOSI, PB3
5. Reset
6. GND

The Arduino Uno board has two ISP ports for external programmers. The one close to the USB connector is best ignored, since it is used to program the USB to serial converter described in the body of this article. The ISP port for the ATmega328 is located at the edge of the board. It’s nice to know that it’s possible to connect an external programmer, because it means you can always restore everything to the original state if you do something really wrong, such as accidentally deleting the boot loader.

You can also use the programmer to read out the content of the flash memory, for example in order to make a backup copy (including the boot loader) that you can use to restore the Arduino if necessary. A backup copy (UnoBoot.hex) [1] is also available in the Elektor archive in case you need it.

In order to load your own program into the microcontroller, you first have to load the hex file. In this case you must tick the option ‘Erase chip before programming’ to ensure that the boot loader is deleted. This means you have to decide whether or not you want to use the boot loader. You can also opt to use another boot loader, such as the MCS boot loader. We’ll describe how that works in one of the upcoming instalments.

You can also view the microcontroller fuse settings with Bascom or in Atmel Studio 6. By the way, the term ‘fuse’ comes from the early days of microcontroller technology when there was program memory in which links were literally burnt through to program the memory once and for all. The modern fuses in ATmega microcontrollers are actually flash memory cells that must be configured to define specific basic settings. Among other things, they allow you to switch back and forth between the internal clock and an external clock source. The screenshot shows the factory default fuse settings of the Arduino Uno board. You can see that the boot loader is enabled (BOOTRST) and how much memory is reserved for it.
reference voltage. The trick here is to use the imprecise supply voltage as the reference voltage and measure the precise internal reference voltage. In this case the program must calculate in the opposite direction to determine the actual supply voltage (see Listing 4). The first measurement (Figure 5) shows the supply voltage with USB power, which in this case is about 4.5 V. Then an external power supply is connected while the program is running, so the supply voltage is provided by the distinctly more precise 5 V voltage regulator. The corresponding measurement is accordingly close to 5.0 V.

Measuring the temperature

In addition to measuring its own supply voltage, the ATmega328 can measure its own temperature thanks to an internal temperature sensor connected to channel 8 of the A/D converter. The output voltage of this sensor is proportional to the temperature, with a factor of approximately 1 mV per degree. The accuracy is not especially high – the specified tolerance is ten degrees. However, you can always calibrate it yourself. According to the data sheet, the sensor output is 242 mV at –45 °C, 314 mV at 25 °C and 380 mV at 85 °C. To obtain meaningful readings with voltages in this range, you have to use the relatively low internal reference voltage (1.1 V). However, the same program is supposed to measure higher voltages as well, so it automatically switches to the higher reference voltage (Avcc) as necessary. For internal temperature measurement the A/D converter operates with the internal 1.1 V reference voltage and a resolution of about 1 mV, which roughly corresponds to 1 degree at the temperature sensor output. At 25 degrees C the sensor output is approximately 314 mV. We decided to take a quick-and-dirty empirical approach, and we determined that the temperature in degrees C could be found by subtracting 338 from the measured value (see Listing 5). However, even better results could be obtained with a bit more computation effort. In any case, you should adjust the results according to your actual situation to obtain the best possible match with the temperature shown by an ordinary thermometer. Next we held a finger on the microcontroller to see whether the temperature rose. Then we wanted to see whether the temperature of the microcontroller increases when it is driving a load. Figure 6 shows a 220 Ω resistor connected between two port pins. If one output

![Diagram](image_url)

Figure 6. Temperature measurement under load.

**Listing 4. Measuring Vcc.**

```bas
'----------------------------------------
'UNO_AD3.BAS  Vcc / 1.1 V
'----------------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim D As Word
Dim U As Single

Config Adc = Single , Prescaler = 64 , Reference = Avcc

Do
    D = Getadc(14)    'Ref 1.1 V
    U = 1023 / D
    U = U * 1.1
    Print "Vcc = ";
    Print U ; " V"
    Waitms 1000
Loop
```

a total of eight channels, of which six are connected to pins A0 to A5. The other two are only fed out in the SMD version of the ATmega328. However, there are also several ‘hidden’ internal channels, and one of these (channel 14) allows the microcontroller to measure its internal 1.1 V reference voltage.
is set high and the other is set low, the voltage on the load resistor is close to 5 V. This results in a load current of roughly 20 mA. We wanted to see whether this causes the microcontroller to warm up. The program also measures the voltage drops on the port pins. For this purpose, the reference voltage must be briefly switched to 5 V. Although it is somewhat unusual for a program to work with two different reference voltages, switching voltages while the program was running does not cause any problems.

The results: with an output current of 20 mA, the voltage drops are just 0.4 V at the lower output and 0.5 V at the upper output. This leaves 4.1 V

Listing 5. Temperature and voltage drops

```
’-------------------------------------------
’UNO_AD4.BAS  Temp
’-------------------------------------------
$regfile = “m328pdef.dat”
$crystal = 16000000
$baud = 9600

Dim D As Word
Dim N As Word
Dim U As Single

Config Portb = Output
Portb.5 = 1
Portb.4 = 0

Do
    Config Adc = Single , Prescaler = Auto , Reference = Internal ‘1,1 V
    Waitms 200
    D = Getadc(8)                                             ‘Temperature
    D = D - 338
    Print D ; “ deg”
    Waitms 500
    Config Adc = Single , Prescaler = Auto , Reference = Avcc ‘5 V
    Waitms 200
    D = Getadc(0)
    U = D * 5.0
    U = U / 1023
    Print U ; “ V”
    D = Getadc(1)
    U = D * 5.0
    U = U / 1023
    Print U ; “ V”
Loop
```

Figure 7. Temperatures and voltage drop
across the load resistor, yielding a current of 19 mA. The additional power dissipation in the microcontroller is 17 mW (0.9 V x 19 mA). Significant warming cannot be expected from this amount of power, and that’s exactly what we saw from the temperature measurement (Figure 7). According to the data sheet, the maximum current at any port pin should not exceed 40 mA and the total current should not exceed 200 mA. In our experiments we found that the ports could handle currents up to 100 mA without damage, which means that even small DC motors could be operated directly from the ports without driver ICs. However, if you want to be on the safe side the rule is to never exceed 40 mA.

**Listing 6. Measuring the input hysteresis**

```bas
'---------------------------------------------------
'UNO_AD5.BAS   Hi/Lo Input Threshold
'---------------------------------------------------
regfile = "m328pdef.dat"
crystal = 16000000
baud = 9600

Dim D As Word
Dim Dmin As Word
Dim Dmax As Word
Dim N As Word
Dim U As Single
Config Portb = Output
Config Adc = Single, Prescaler = 64, Reference = Avcc '5 V

For N = 1 To 10000
    Portb.5 = Not Pinc.5
Next N

Dmin = 1023
Dmax = 0

Do
    For N = 1 To 10000
        Portb.5 = Not Pinc.5     '13 100k A5
        D = Getadc(5)           'A5 100µF GND
        If D < Dmin Then Dmin = D
        If D > Dmax Then Dmax = D
    Next N
    U = Dmin * 5.0
    U = U / 1023
    Print "Min = ", U, " V"
    U = Dmax * 5.0
    U = U / 1023
    Print "Max = ", U, " V"
Loop
```

Figure 8. Measuring the input hysteresis.

Figure 9. The upper and lower switching thresholds of an input.
Measuring the input hysteresis

In the previous instalment of this series we used external test equipment to determine the switching thresholds of a digital input. Now we want to do the same thing entirely automatically with the built-in A/D converter (see Figure 8). For this we use two new variables (Dmin and Dmax) to hold the values of the lower and upper switching thresholds (see Listing 6). The measurement loop is preceded by a control loop which ensures that the input voltage is already within the desired range before the actual measurement starts. At first the maximum possible value is stored in the Dmin variable and zero is stored in the Dmax variable. The program then looks for the actual limit values. This is done using a random-sample approach. The longer the measurement session lasts, the more accurate the determined limit values are.

This program shows how to use loop counters. The For-Next loop counts down from N = 1000 to 0 and executes the instructions inside the loop exactly 1000 times. Although one thousand iterations may sound like a lot, the entire process takes only a fraction of a second. The large number of measurements increases the probability that the measurement results include the true limit values. This means that you have a good chance of finding the correct values after just one round. This can be seen from the fact that the subsequent rounds do not yield any significant changes.

For N = 1 To 1000
  ... instructions ...
Next N

Figure 9 shows the result of this experiment. The switching thresholds are 2.23 V and 2.62 V. These are approximately the same as the values determined using the external test equipment.

Web Link
XXL LED VU Meter
With a wide range and high resolution

An LED VU meter is an excellent tool for keeping a close eye on the magnitude of audio voltages and so help prevent an input from being overloaded, for example when used with a mixing desk or a guitar amp. The circuit described here has an exceptionally wide input range of 60 dB and a step size of 1 dB thanks to the use of 6 LM3915 driver ICs, which control a total of 60 LEDs.

The author was looking for an LED VU meter for his DIY guitar tube amplifier. After some research an old veteran soon presented itself: the LM3915 dot/bar display driver. This IC spans 30 dB with a step size of 3 dB. It can be used to drive a total of 10 LEDs. Various LED VU kits using this IC are available and are usually satisfactory.

Further research resulted (of course) in an Elektor article by Rikard Lalic. This article from the 2002 Summer Circuits edition [1] describes an LED VU meter, again using the LM3915, but now with a range of 60 dB, which is achieved by using 2 LM3915 ICs, where the second IC receives a voltage that is 30 dB smaller.

Rising to the challenge to make an LED VU meter that has this wide range, as well as a step size of 1 dB, this article describes the design, construction and test of such an XXL LED VU meter, with a range of 60 dB and a resolution of 1 dB, meaning there’s a grand total of 60 LEDs to be controlled!

An interesting aspect of this design is that it is entirely analog (there is no microcontroller to be found) and that is has a didactic purpose. The author uses it to explain to his students that mathematics and the design of electronics go hand in hand. This article also contains a few sidebars where readers can refresh their own theoretical knowledge somewhat!

Design and calculations

Lalic, in his original design, assumed a fixed set-point of 5 V for the first LM3915. To allow this design to measure higher powers as well, he...
supplied a table where the input signal to the LM3915 could be adjusted using a voltage divider. Because the LM3915 has a default step size of 3 dB, having a step size of 1 dB has to be solved by using multiple LM3915 ICs. In the original design the second LM3915 was adjusted 30 dB by using multiple LM3915 ICs. In the original design the second LM3915 was adjusted 30 dB lower than the first LM3915. This can be further refined by adding two more ICs, which are adjusted to be 1 dB and 2 dB lower respectively, relative to both of the existing ICs. All this together will now create a total range of 60 dB in steps of 1 dB. Therefore we have a total of six LM3915 ICs which are adjusted according to Table 1.

As an example we will assume that full power drive is reached at an RMS power of 100 watts into 8 Ω. This corresponds to an 28.28 Vrms and 40 Vpeak (see sidebar ‘Power and voltage: RMS and peak?’). Since the LM3915 responds to the peak value (because it is, in fact, a comparator), the voltage divider has to be scaled such that this 40 V corresponds to the reference voltage of 5 V at the input of the LM3915. Supposing that R2 has a fixed value (10 kΩ), then the correct voltage divider ratio can be set with the aid of R1 for various (RMS) powers and loudspeaker impedances RL.

In the sidebar we already observed that

\[ P_{\text{rms}} = \frac{\hat{V}^2}{2R_L} \]

Rearranging this becomes

\[ \hat{V} = \sqrt{2P_{\text{rms}}} R_L \]

This peak voltage \( \hat{V} \) is reduced by the voltage divider \( R_1/R_2 \) to 5 V, so that we obtain the following:

\[ \hat{V} = \frac{10k}{R_1+10k} = 5 \iff \hat{V} \times 10k = 5 \times (R_1+10k) \iff \hat{V} \times 10k - 50k = 5 \times R_1 \]

Fig. 1. Because six ICs type LM3915 have been used, a total of 60 LEDs can be controlled, which results in a range of 60 dB and a step size of 1 dB.

### Table 1: Reference voltage for each IC

<table>
<thead>
<tr>
<th>IC-nO.</th>
<th>Ratio [dB]</th>
<th>Ratio</th>
<th>U_{REF} [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1</td>
<td>0 dB</td>
<td>1.000</td>
<td>5.000</td>
</tr>
<tr>
<td>IC2</td>
<td>−1 dB</td>
<td>0.8913</td>
<td>4.456</td>
</tr>
<tr>
<td>IC3</td>
<td>−2 dB</td>
<td>0.7943</td>
<td>3.972</td>
</tr>
<tr>
<td>IC4</td>
<td>−30 dB</td>
<td>0.0316</td>
<td>0.158</td>
</tr>
<tr>
<td>IC5</td>
<td>−31 dB</td>
<td>0.0282</td>
<td>0.141</td>
</tr>
<tr>
<td>IC6</td>
<td>−32 dB</td>
<td>0.0251</td>
<td>0.126</td>
</tr>
</tbody>
</table>

www.elektor-magazine.com | June 2014 | 41
Component List

Resistors
Default: 0.25 W 1%
R1 = see text and Table 2
R2 = 10kΩ
R3,R6,R9 = 820Ω
R4a = 36kΩ
R4b,R7b = 2.4kΩ
R5,R8,R11 = 47Ω
R7a = 9.1kΩ
R10 = 1.6kΩ
R12,R13,R14 = 680Ω
P1–P6 = 100Ω trimpot, horizontal

Capacitors
C1 = 1000µF 35V, low ESR, 5mm pitch, 13mm diam.
C2 = 10µF 16V, low ESR, 2mm pitch, 5mm diam.
C3 = 1µF 100V, MKT, 5mm pitch
C4 = 1nF, 5mm pitch
C5–C10 = 100nF, 2.5mm pitch
C11,C12,C13 = 22nF, 2.5mm pitch

Semiconductors
B1 = DB101
D1–D50 = 10-element LED bargraph display, green, 10.16 x 25.40 mm
D51–D60 = 10-element LED bargraph display, red, 10.16 x 25.40 mm
IC1–IC6 = LM3915N
IC7 = LM7812 (TO220)

Miscellaneous
J1,J2,J3 = 2-pin pinheader, 0.1” pitch
6 pcs. DIP-20 IC socket for LED bargraph displays
6 pcs DIP-18 IC sockets for LM3915 ICs
PCB # 120698-1, see [2]

Figure 2. The printed circuit board layout designed by the author. The LEDs can be fitted at either the component side or the solder side, whichever is preferred (but note the correct polarity).
And finally we can solve this for $R_1$:

$$R_1 = \hat{U} \times 2k - 10k$$

Using the formulas for $\hat{U}$ and $R_1$, **Table 2** can be populated with various values of $P_{\text{RMS}}$ and $R_L$. The purpose of capacitor $C4$ at the input is to prevent undesirable RF interference and together with $R1$ forms a low-pass filter. A 12-V voltage regulator provides a stable power supply voltage.

### Design of the reference voltages

The generation of the reference voltages requires a little extra attention. For one thing, since the range of 60 dB is reasonably large this has the consequence that the reference voltages can become quite small. A voltage ratio of 60 dB does, after all, mean a ratio of 1000 to 1. The bottom-most LED will therefore already turn on at only 5 mV! A good printed circuit board design with ground and power planes is essential for a successful implementation!

The LM3915 generates a constant voltage of 1.25 V between pin 7 (Ref Out) and pin 8 (Ref Adj). The current that is required to generate this voltage also determines the current that will flow through each LED. Using a resistance value of 820 $\Omega$ there will be a current of about 15 mA running through each LED. This is sufficient to turn the LEDs on with plenty of brightness.

As the first step, the reference voltage at IC1 can be set to 5 V using potentiometer P1. This voltage is measured at test point TP1. Once this is done, the voltage at IC4 can be set to 158 mV (see **Table 1**) using potentiometer P4. This voltage is measured at test point TP4. After this proceed to adjust the center sections. Using potentiometer P2 adjust the reference voltage for IC2 to 4.456 V. This voltage is measured at test point TP2. This is then followed by setting the voltage for IC5 to 141 mV using potentiometer P5. This voltage can be measured at test point TP5.

Next use potentiometer P3 to set the reference voltage for IC3 to 3.972 V. This voltage is measured at test point TP3. Finally use potentiometer P6 to set the voltage for IC6 to 126 mV. This voltage is measured at test point TP6.

### Circuit board assembly

Assembling the circuit board should present few difficulties. For his prototype the author chose to mount the LEDs on the solder side of the circuit board. These, however, could also be fitted without any problems on the component side, if preferred. Just make sure that the LEDs are mounted with their polarity the correct way around. One of the four corners of the LED has a bevel. This side must face the long edge of the board.

With some resistance values (for example R1, see **Table 2**) it may be necessary to solder two resistors in parallel in order to obtain the correct value. You can, of course, also use 1%-resistors, but these are not always all available in the assortment of resistors in the (hobby) lab. After all the parts have been mounted (but do **not yet** fit the LM3915 ICs in their sockets) you can begin with testing the power supply section. There are two options here:

- Connect a transformer with an output of 12 V$_{\text{AC}}$ to connector J1 and mount B1, C1 and IC7.
- Use an external 12 V$_{\text{DC}}$ power supply and connect this to connector J2. In this case IC7 must **not** be fitted.

Measure that there is a stable 12-V signal everywhere (pin 3 and pin 9 of each IC socket). If

---

**Table 2: Value of R1 for different powers**

<table>
<thead>
<tr>
<th>Loudspeaker Impedance $R_L$</th>
<th>4 $\Omega$</th>
<th>4 $\Omega$</th>
<th>4 $\Omega$</th>
<th>8 $\Omega$</th>
<th>8 $\Omega$</th>
<th>8 $\Omega$</th>
<th>16 $\Omega$</th>
<th>16 $\Omega$</th>
<th>16 $\Omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS Power $P_{\text{RMS}}$</td>
<td>10 W</td>
<td>50 W</td>
<td>100 W</td>
<td>10 W</td>
<td>50 W</td>
<td>100 W</td>
<td>10 W</td>
<td>50 W</td>
<td>100 W</td>
</tr>
<tr>
<td>Peak Voltage $\hat{U}$</td>
<td>8.9 V</td>
<td>20.0 V</td>
<td>28.3 V</td>
<td>12.6 V</td>
<td>28.3 V</td>
<td>40.0 V</td>
<td>17.9 V</td>
<td>40.0 V</td>
<td>56.6 V</td>
</tr>
<tr>
<td>$R1$ ($\Omega$)</td>
<td>7889</td>
<td>30000</td>
<td>46569</td>
<td>15298</td>
<td>46569</td>
<td>70000</td>
<td>25777</td>
<td>70000</td>
<td>103137</td>
</tr>
<tr>
<td>$R1$ – implementation</td>
<td>22k//12k</td>
<td>47k//82k</td>
<td>47k</td>
<td>15k</td>
<td>47k</td>
<td>220k//100k</td>
<td>47k//56k</td>
<td>220k//100k</td>
<td>330k//150k</td>
</tr>
<tr>
<td></td>
<td>(-1.58%)</td>
<td>(-0.41%)</td>
<td>(+0.9%)</td>
<td>(-2.0%)</td>
<td>(+0.9%)</td>
<td>(-1.8%)</td>
<td>(-0.87%)</td>
<td>(-1.8%)</td>
<td>(0.0%)</td>
</tr>
</tbody>
</table>
not, switch off the power supply and track down the fault. Once all is well, you can fit (with the power supply off) all the LM3915 ICs carefully in their sockets. Turn the power supply back on and adjust all the reference voltages to their correct values according to Table 3.

After this the board is ready for use. The loudspeaker signal is connected to connector J3.

Web Links

---

### Table 3: Adjustment of the reference voltages

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP1</td>
<td>5.000 V</td>
<td>4.456 V</td>
<td>3.972 V</td>
<td>158 mV</td>
<td>141 mV</td>
<td>126 mV</td>
</tr>
</tbody>
</table>

---

**The Author**

Emile van de Logt works at Hogeschool Rotterdam, Netherlands, as Head of Department for the Electric Engineering and Health Care Technology departments. He studied Electric Engineering at Eindhoven Technical University, and Management Studies at Open University. An enthusiastic electronics hobbyist, Emile looked after Elektor’s *Embedded C Programming* and *FPGA-VHDL* workshops. He builds tube amplifiers, guitar effects equipment and is an amateur beer brewer with an entirely automated brewing system, the hardware and software of which he designed himself.

---

### The Logarithm

The logarithm is often used in order to allow large ratios between numbers to be expressed clearly. Normally the logarithm with base 10 is used. There are a few rules that are important when calculating with logarithms:

\[ \log_{10}(b) = x \] can also be written as a power: \[ a^x = b. \]

The knack of working with logarithms is to transform one of these forms into the other.

For example: how big is \( \log_{10}(1000) \)? If we call the answer \( x \), then we can also write: \( \log_{10}(1000) = x \), that is \( 10^x = 1000 \), that is \( 10^x = 10^3 \), therefore \( x \) (the answer) is 3. Consequently \( \log_{10}(1000) \) equals 3. Similarly, \( \log_{10}(1000) = 3. \)

---

### The Decibel

is named after Alexander Graham Bell. The Bel (B) was originally used to indicate the power loss in cables. An unwieldy unit it was soon redefined to deci-bel (0.1 bel).

When working with decibels it is important to make a distinction between power and voltage ratios. The definitions of these are as follows: number of dB’s = \( 10 \log_{10} \left( \frac{P_1}{P_0} \right) \). When \( P_1 \), for example, is 100 times bigger than \( P_0 \), then this ratio corresponds to 20 dB.

With voltage ratios the definition is as follows: number of dB’s = \( 20 \log_{10} \left( \frac{U_1}{U_0} \right) \). When \( U_1 \), for example, is 100 times bigger than \( U_0 \), then this ratio corresponds to 40 dB!

Assuming we want the difference between two voltages \( U_1 \) and \( U_0 \) to be 30 dB we can write: \( 20 \log_{10} \left( \frac{U_1}{U_0} \right) = 30 \) that is, \( \log_{10} \left( \frac{U_1}{U_0} \right) = 1.5 \). According to the notations defined for the logarithm this can also be written as: \( 10^{1.5} = \left( \frac{U_1}{U_0} \right) \), that is, \( U_1/U_0 = 31.62 \). Consequently \( U_1 \) has to be 31.62 times bigger than \( U_0 \).

---

### Power and voltage: RMS and peak?

For a pure sine wave signal it holds that the peak value (indicated with \( \hat{U} \)) is bigger by a factor of \( \sqrt{2} \) compared to the RMS (root-mean-square) value (indicated with \( U_{\text{rms}} \)).

We know that \( P \) (power) = \( U \) (voltage) \times I \) (current). Ohm’s Law tells us that \( U = I \times R \), therefore

\[ P = U \times I = U \times \frac{U}{R} = \frac{U^2}{R}. \]

It is important to indicate which power is used: RMS power \( (P_{\text{rms}}) \) or peak power? You only get RMS power if you use the RMS value of the voltage: \( P_{\text{rms}} = U_{\text{rms}}^2/R = (\hat{U}/\sqrt{2})^2/R = \hat{U}^2/2R \).

The RMS power is therefore half of the peak power!

An RMS power of 100 watts into a loudspeaker of 8 ohms is thus realized by a sine wave signal with an RMS voltage \( U_{\text{rms}} = 28.28 \) V. This signal has a peak voltage \( \hat{U} \) of 40 V.
Further Information and Ordering at www.elektor.com/retronics

Retronics
80 tales of electronics bygones
This book is a compilation of about 80 Retronics installments published in Elektor magazine 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.

£26.95 • €29.95 • US $41.00

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

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

Easy As, with Cleverscope.

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

100uH toroid

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

CS328A-FRA
14 Bit MSO

www.cleverscope.com

EAGLE V6 Getting Started Guide
Learning to fly with Eagle
This book is intended for anyone who wants an introduction to the capabilities of the CadSoft's EAGLE PCB design software package. After reading this book while practicing some of the examples, and completing the projects, you should feel confident about taking on more challenging endeavors.

The book will quickly allow you to:
• obtain an overview of the main modules of EAGLE: the schematic editor; layout editor and autorouter in one single interface;
• learn to use some of the basic commands in the schematic and layout editor modules of EAGLE;
• apply your knowledge of EAGLE commands to a small project;
• learn more about some of the advanced concepts of EAGLE and its capabilities;
• understand how EAGLE relates to the stages of PCB manufacture;
• create a complete project (a proven design from the engineering team at Elektor), from design through to PCB fabrication.

Incl. CD-ROM containing EAGLE 6.4.0 for MS Windows, Linux and Mac
£30.95 • €34.50 • US $47.00

Further Information and Ordering at www.elektor.com/eagle
DIY $I^2C$ for Raspberry Pi

Using a PIC16F88 microcontroller

By Bert van Dam (Netherlands)

This small article describes how to build your own $I^2C$ slave device using a PIC microcontroller, and how to control it with the Raspberry Pi. This approach combines the power of the microcontroller with the flexibility of the Raspberry Pi. There are many things that you can implement with an $I^2C$ device, such as a light meter, remote sensor, relay control or motor controller.

What do you need for this? Actually not that much: a Raspberry Pi board, an SD card with the Raspberry Pi operating system and $I^2C$ drivers (which you can obtain from the author’s Raspberry Pi book, among other sources), a PIC 16F88 microcontroller, a PIC programmer, and a breadboard with an LED, an LDR and a few bits of wire.

**What can you do with an $I^2C$ slave device?**

The Raspberry Pi has a GPIO header for connecting external circuitry, such as $I^2C$ slave devices. The advantage of these $I^2C$ slave devices is that you can use them to connect components that are not suitable for direct connection to the I/O header. You can buy a wide range of $I^2C$ slave devices and connect them to the Raspberry Pi without further ado, including temperature sensors and I/O extenders. However, you can’t buy devices for every possible application, and sometimes the available devices don’t do what you want. If you build your own $I^2C$ slave using a microcontroller, the potential uses are virtually unlimited. For example, the 16F88 used in this article has several analog inputs, digital I/O, high-resolution PWM, a TTL RS-232 port, frequency counters, SPI and $I^2C$ ports and EEPROM all integrated into the device. The possibilities are only limited by your imagination.
Basic I²C bus structure

An I²C data bus consists of one master and one or more slaves. They are connected to each other by data and clock lines as shown in Figure 1. These two lines are held High by a pair of pull-up resistors. Consequently the voltage on the I²C bus depends on the voltage rail to which the pull-up resistors are connected. It also means that it’s entirely possible for the master and slave devices to operate from different supply voltages. In the case of the Raspberry Pi, the pull-up resistors are built in and connected to 3.3 V, so the slaves must be able to handle 3.3 V and see this voltage as a logic high level. The PIC sees any voltage above 2 V as a logic high level, so it has no problem communicating with a Raspberry Pi operating at 3.3 V.

The bus master places a clock signal on the clock line by pulling it low at a specific frequency. The line automatically returns to the high level due to the pull-up resistor. The bus master also puts data pulses on the data line in the same manner, coordinated with the clock pulses. This means that the master controls the communication process and the slaves have to follow.

When a request is sent to a slave, it must respond. However, the slave may not be able to respond before the next clock pulse—for example, if it first has to compute something. In this case the slave can pull the clock line low. Now the master cannot issue another clock pulse because the slave has already pulled the clock line low, and releasing the line has no effect as long as the slave holds it low. This gives the slave the time it needs to generate the required response. When it is ready, the slave releases the clock line so the master can again supply the clock signal. Of course, the clock signal timing is skewed at this point because the pulse was too long. This procedure is called clock stretching. We also use this procedure in our project.

Setting up the I²C bus between RPi and PIC

In the schematic in Figure 2 you can see the Raspberry Pi header on the right and the 16F88 on the left. The microcontroller pins that are not connected to anything in the schematic are left open. The two pull-up resistors required on the I²C data bus are built into the Raspberry Pi and are therefore not shown on the schematic.

The LDR and the LED with the associated resistors are included to show how the circuit works. You probably won’t need them for your own projects. We use the LED in our first demo project and the LDR in the second one.

If you build this circuit on a breadboard as shown in Figure 3, you must fit a 100 nF capacitor in each corner to suppress noise on the supply lines. These capacitors are not shown on the schematic. Note that if you previously used the microcontroller in other projects, pin 7 and/or pin 10 may already be configured as outputs. In that case they will have an output voltage of 5 V, which can damage the Raspberry Pi. To avoid this, load the I²C program into the microcontroller before you connect it to the Raspberry Pi. After this you can reprogram the microcontroller without any risk. One option for programming the microcontroller is to use the Wisp648 in-circuit programmer. In that case you can simply leave the microcon-
controller plugged into the breadboard during programming. This is particularly handy when you are experimenting with the circuit and constantly making changes to the software.

However, when you use an in-circuit programmer it is a good idea to use an external power source for the 5-V portion instead of the Raspberry Pi, to ensure that you always have enough power available. In that case the 5 V pin of the Raspberry Pi must be left open, since you can’t power the circuit from two sources at the same time. However, the rest of the pins (including ground) should remain connected.

Software structure
For the sake of simplicity, we omit any description of the settings (you can read about them at your convenience in the source code) and limit ourselves to a number of basic instructions. Here we use Python, the standard free programming language for the Raspberry Pi.

We use the following Raspberry Pi instructions:

```python
bus.write_byte(address, command)
```
This sends the content of the variable ‘command’ to the I2C slave device with the given address. This means that the slave receives an address and a command.

```python
data = bus.read_byte_data(address, command)
```
This sends the content of the variable ‘command’ to the I2C slave device with the given address. After this the master waits for a response from the slave, which is copied to the variable ‘data’. In this case the slave receives an address, a command, and then another address.

```python
data = bus.read_byte(address)
```
Sometimes the slave does not need any command. For example, it may only be able to measure one thing, so that is the only thing the master can request. In that case we can omit the command and simply send the address with the Read bit set. Of course, the software in the slave device must know in advance what it has to send back.

```bash
sudo i2cdetect -y 1
```
You can issue this instruction when you want to know which I2C slaves are present on the bus. This is not a Python instruction, so you must enter it in a command line or use LXterminal if you are working with Xwindows (LXDE). In return you will receive the following reply:

```bash
pi@raspberrypi $ sudo i2cdetect -y 1

0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```
Here you can see that a single I2C slave has replied and its address is 0x20 in hexadecimal notation.

On the microcontroller side we also skip all the settings. Here we use JAL, a widely used free programming language for microcontrollers.

We use the following PIC instructions:

```c
i2c_slave_enable (address)
```
You can use this instruction to configure the communication unit of the 16F88 for I2C and set the slave address. The maximum address value is 127 (7 bits).

```c
data = i2c_slave_read_byte_timeout(error)
```
This function reads a bytes and uses a timeout counter. The bytes normally arrive one after the other, but if it takes too long a timeout occurs. When this happens, the function stops waiting and sets the variable ‘error’ to true.

```c
data = i2c_slave_read_byte
```
This function also reads a byte, but without a timeout. You can use this to wait for the first address. There is no timeout here because it does not matter how long you have to wait, since it is entirely possible that the master never tries to contact the slave.

```c
data = i2c_slave_write_byte(data, error)
```
This function writes a byte of data. The clock line is held low until this function is ready, which means that the master must wait for the slave. If the write operation fails, the variable ‘error’ is set to true.
Demo project 1: LED switching
The first project involves controlling a LED over the I2C bus. This requires the Raspberry Pi to send a command to the microcontroller, with no response expected. We use the following instruction at the Raspberry Pi end:

```c
bus.write_byte(address, command)
```

This means that the microcontroller receives the address and a command.

```c
-- wait for address (and ignore)
dummy = i2c_slave_read_byte

-- wait for command
command = i2c_slave_read_byte_timeout(error)
```

The LED must be switched on or off, depending on the content of the command. A ‘1’ causes the LED to light up, while a ‘0’ makes it go dark. A timeout during reading or a hung bus during writing is a serious error. Normally these errors do not happen, but users may send incorrect commands or there can be a technical problem with the bus. I2C slaves usually do not have users, so there is nobody who can do anything to clear these errors. The best solution in such cases is to perform a soft reset on the microcontroller. This takes practically no time to execute, and it automatically restores all registers and flags to good states without requiring the user (that means you) to know exactly what was wrong.

```c
if error then
  -- unexpected I2C event, soft reset
  asm CLRF PCLATH
  asm GOTO 0x00
end if
```

Demo project 2: Light Meter
The second project is a light meter using an LDR. The microcontroller is blessed with five unused analog channels. Here we use just one of them to connect the LDR. In this case it is not necessary to send a command to the microcontroller, since there’s only one thing it can measure. We simply send the address with the Read bit set:

```c
data = bus.read_byte(address)
```

The microcontroller receives this address with the Read bit set and knows that it has to return a response. It prepares the response and sends it back:

```c
-- wait for address (and ignore)
dummy = i2c_slave_read_byte

-- prepare reply and send it
mydata = ADC_read_low_res(0)
i2c_slave_write_byte(mydata, error)
```

This is followed in the code by the usual error handling routine, which you can read at your leisure in the downloadable source code for this article. With a few minor changes, you can also use the code to read a temperature sensor or operate an ultrasonic distance sensor, such as the SRF05.

The software necessary for this article is available for download at [1]. For more projects of a similar nature, see the books Raspberry Pi – Explore the RPi in 45 Projects, 50 PIC Microcontroller Projects for Beginners and Experts, and PIC Microcontrollers – 50 Projects for Beginners and Experts, all available in the Elektor Store.

Web links

About the author
Bert van Dam is a freelance author of books, course material and articles about PIC and ARM microcontrollers, Arduino, Raspberry Pi, Artificial Intelligence, and the programming languages JAL, C, assembler, Python and Flowcode.
Many techniques are still at the research lab stage. Others, like those using gallium arsenide (GaAs), which currently offer the greatest efficiency (20–25%), are used for powering satellites. Their cost puts them out of reach for private individuals. Here are the commonest techniques.

**Monocrystalline silicon**
Monocrystalline silicon cells, very widespread, offer good efficiency: 15% (under standard test conditions). These bluish colored cells consist of a single crystal and have a uniform appearance. The manufacturing process involves melting bars of silicon then cooling them slowly to achieve a homogeneous crystal.

**Polycrystalline silicon**
As their name indicates, polycrystalline silicon cells are made up of multiple crystals, with an appearance rather like a mosaic; their efficiency is 12–14% below that of monocrystalline cells; they are also less expensive.

**Amorphous silicon**
Amorphous silicon cells have only low efficiency (5–7%), but they work just as well indoors as out, which means they are used primarily in consumer applications (watches, calculators, etc.)

**CIS cells**
CIS (copper/indium/selenium) cells represent the new generation of thin-film solar cells. The
materials used in making are not toxic and are easy to obtain. This type of cell can be made on flexible modules, but the manufacturing process for them is quite complex.

They offer good efficiency (10–14 % or even higher) and are starting to be produced, in particular by Saint-Gobain in the author’s home country, for industrial installations.

**Definition of solar energy**

It’s important to make it clear that the function of photovoltaic solar panels is to convert light radiation into electric current; if their operation was associated with direct solar radiation, the slightest cloud would compromise their reliability in use. The efficiency of a photovoltaic panel depends first and foremost on the exposure: in strong sunshine out of a cloudless sky, this can reach 1000 W/m². In cloudy weather, it is still around 500 W/m², while in overcast, rainy weather, it remains at best between 100 and 50 W/m²! If it doesn’t drop to 0 W/m²...

The watt-per-square-meter is the unit used to quantify solar irradiation. Solar panels are characterized by the electrical power they produce under solar radiation of 1000 W/m² at a temperature of 25 °C.

In concrete terms, a so-called 20-watt solar panel produces these 20 watts under light radiation of 1000 W/m² (strong sunshine in cloudless sky); in overcast weather, only around half of this power
Energy is an element that is integrated over a period of time. Electrical power is measured in watts (W), kilowatts (kW), or megawatts (MW). Power is the transfer of energy per unit time.

\[
\text{energy} = \text{power} \times \text{time}
\]

Energy is expressed in watt-hours (Wh). If the battery voltage is known, the unit of consumption normally used in calculations is the ampère-hour (Ah).

**Evaluating consumption**

Photovoltaic panels only supply energy during the day, but the unit of time used is the whole 24 hr period, which means we’ll be working in Ah per day.

**Sizing the battery**

As the photovoltaic panel supplies energy only when it is receiving light radiation, a battery is needed to store the energy. The battery life expectancy is the time for which the battery can keep the load operating without being recharged by the photovoltaic panel. We generally allow for a battery life of five days, to cover severe winter weather conditions with extremely low light radiation.

In order to calculate the battery (accumulator) capacity, we multiply the electrical consumption (Ah) by the battery life required, and then apply a safety factor (0.7) corresponding to the battery’s loss of charge at the lowest operating temperature considered and to the additional losses (battery charging efficiency, regulator conversion efficiency, battery self-discharge). The calculation indicates the capacity actually available at any moment. In the example below, this is equal to 70 % of the nominal capacity.

**Calculations and implementation**

In order to size and then implement a photovoltaic system, we have to proceed in stages.

**Evaluating requirements**

Before starting to calculate, let’s remember the difference between energy and power. Power is an instantaneous dimension (rate of flow).

Example: a lamp consumes 20 W; a solar panel produces 40 W at instant \( t \).

<table>
<thead>
<tr>
<th>device(s)</th>
<th>number of devices</th>
<th>current drawn (A)</th>
<th>duration of use in 24 hrs</th>
<th>energy = current ( \times ) duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>photovoltaic regulator</td>
<td>1</td>
<td>0.020</td>
<td>24 h</td>
<td>0.020 ( \times ) 24 = 480 mAh</td>
</tr>
<tr>
<td>LED lamps</td>
<td>3</td>
<td>0.150</td>
<td>4 h</td>
<td>0.150 ( \times ) 3 ( \times ) 4 = 1.8 Ah</td>
</tr>
<tr>
<td>total energy consumed over 24 hrs:</td>
<td></td>
<td></td>
<td></td>
<td>2.28 Ah</td>
</tr>
<tr>
<td>power consumed per day:</td>
<td></td>
<td></td>
<td></td>
<td>2.28 ( \times ) 12 V = 27.36 Wh/day</td>
</tr>
</tbody>
</table>

will remain. Not forgetting the junction temperature of the photovoltaic cells, which also influences the panel’s efficiency: the modules lose 0.4 % per degree. Solar panel operating temperature depends on the incident radiation, the ambient temperature, the color of the materials, and also the wind speed (5–14% cumulative loss). The second determining factor is the exposure duration, as even on a fine Summer’s day, the lighting varies throughout the day in a not-in-significant way. The time window during which the solar panel receives its maximum energy is only a few hours. In Winter, the light radiation diminishes, and can even disappear for several days in the event of snow or fog.
Example of measuring exposure

These readings taken in December show that the exposure level was low (50 W/m² on 12/18 and 200 W/m² on 12/19) in the French city of Tours. The measurements were taken using a solarimeter, which gives an instantaneous reading of the solar irradiation in W/m².

Using photovoltaic panels

The position and orientation of the photovoltaic panels are determining factors for the efficiency of the installation.

Orientation: the cardinal point is located towards the sun’s highest point; in the northern hemisphere, that’s the south, and towards the north if you are in the southern hemisphere.

Inclination: the optimum inclination with respect to the horizon is usually taken as the latitude of the location plus 15°.

The “L+15°” inclination rule of thumb optimizes the production of energy in the least favorable months (Winter); for Summer use only, the optimum angle would be much smaller. You can refer to the website [2] which gives the optimum inclination for each month of the year (Table 2).

Calculating recoverable solar energy

The solar irradiation (in W/m²) falling on the photovoltaic panel varies during the day, depending on the time and weather conditions. Various websites give the amount of sunshine in Wh/m² according to the time of day and geographical location. The average values are spread over the year. For sizing the installation, we’ll take the least favorable value (December).

A few average solar irradiation values:

- Paris region: 1.12 kWh/m²/d
- South of France: 3 kWh/m²/d
- North Africa: 5 kWh/m²/d

panel power =

\[
\frac{\text{consumption of system over 24 hr (Wh/d)}}{\text{daily exposure (kWh/m²/d)} \times \text{loss factor}} = \frac{32.8}{1.120 \times 0.7}
\]

The panel power again has been rounded. Which gives requirement for a solar panel with a power of 50 W.

Note: Settling for a 20-watt panel to power a load of the same power is a (frequent) error to be avoided!
Consumers and start heating up. It’s sometimes called the hot spot effect, and can have undesirable consequences.

Maintenance

With an inclination of L+15°, rain does a good job keeping photovoltaic panels clean. You’ll need to periodically check the connections are in good condition (tightening, oxidation).

Charging regulator operating principle

The charging regulator [1] checks and protects the battery from over-charging and also deep discharge, which would damage it irreversibly.

A photovoltaic installation is made up as follows.

Reverse blocking diode

Generally inserted in the circuit at the solar panel connecting box, this avoids the battery’s discharging through the solar panel at night or when the battery is fully charged and its voltage is higher than that of the panel under low illumination.

Schottky diodes are preferred for their low forward voltage (0.4–0.6 V). On bottom-end panels, we find conventional diodes, which entail a voltage drop of 0.8 V. This higher threshold is not negligible in our calculations, as on a 12 V solar panel it entails a 7% loss.

Charging regulator

Its function is above all to preserve battery life. When the battery voltage reaches 14.4 V (nominal threshold), the regulator interrupts the charging of the battery, starting it again when the voltage drops below 12.6 V again.

Most modern chargers use a battery charge /

---

**Table 2. Solar Irradiation for a site in central France, drawn up by PVGIS (Photovoltaic Geographical Information System) [2]**

Location: 47°13’10” North, 2°11’50” East, Elevation: 113 m ASL.
Solar radiation database used: PVGIS-CMSAF.
Optimal inclination angle is: 36 degrees from vertical (= 54 degrees elevation).
Annual irradiation deficit due to shadowing (horizontal): 0.0 %.

<table>
<thead>
<tr>
<th>Month</th>
<th>$H_h$</th>
<th>$H_{opt}$</th>
<th>$H(36)$</th>
<th>$T_{opt}$</th>
<th>$T_{24h}$</th>
<th>$N_{DD}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>1070</td>
<td>1710</td>
<td>1710</td>
<td>63</td>
<td>4.5</td>
<td>400</td>
</tr>
<tr>
<td>Feb</td>
<td>1780</td>
<td>2580</td>
<td>2580</td>
<td>57</td>
<td>5.4</td>
<td>341</td>
</tr>
<tr>
<td>Mar</td>
<td>3160</td>
<td>4120</td>
<td>4120</td>
<td>47</td>
<td>8.2</td>
<td>273</td>
</tr>
<tr>
<td>Apr</td>
<td>4590</td>
<td>5180</td>
<td>5180</td>
<td>33</td>
<td>10.4</td>
<td>173</td>
</tr>
<tr>
<td>Mei</td>
<td>5370</td>
<td>5390</td>
<td>5390</td>
<td>19</td>
<td>14.6</td>
<td>65</td>
</tr>
<tr>
<td>Jun</td>
<td>6050</td>
<td>5820</td>
<td>5820</td>
<td>13</td>
<td>18.3</td>
<td>16</td>
</tr>
<tr>
<td>Jul</td>
<td>5860</td>
<td>5780</td>
<td>5780</td>
<td>17</td>
<td>20.1</td>
<td>2</td>
</tr>
<tr>
<td>Aug</td>
<td>4960</td>
<td>5360</td>
<td>5360</td>
<td>28</td>
<td>20.2</td>
<td>15</td>
</tr>
<tr>
<td>Sep</td>
<td>3920</td>
<td>4920</td>
<td>4920</td>
<td>43</td>
<td>16.4</td>
<td>97</td>
</tr>
<tr>
<td>Okt</td>
<td>2290</td>
<td>3250</td>
<td>3250</td>
<td>54</td>
<td>13.3</td>
<td>214</td>
</tr>
<tr>
<td>Nov</td>
<td>1220</td>
<td>1910</td>
<td>1910</td>
<td>62</td>
<td>7.4</td>
<td>368</td>
</tr>
<tr>
<td>Dec</td>
<td>971</td>
<td>1530</td>
<td>1530</td>
<td>63</td>
<td>4.7</td>
<td>414</td>
</tr>
<tr>
<td>Year</td>
<td>3440</td>
<td>3970</td>
<td>3970</td>
<td>36</td>
<td>11.9</td>
<td>2378</td>
</tr>
</tbody>
</table>

$H_h$: Irradiation on horizontal plane (Wh/m²/day)

$H_{opt}$: Irradiation on optimally inclined plane (Wh/m²/day)

$H(36)$: Irradiation on plane at angle: 36deg. (Wh/m²/day)

$I_{opt}$: Optimal inclination (deg.)

$T_{24h}$: 24 hour average of temperature (°C)

$N_{DD}$: Number of heating degree-days (–)

PVGIS © European Communities, 2001-2012
discharge calculation algorithm so as to make the best possible use of it.

**Discharge limiter**

Here, the aim is to protect the battery against deep discharge. When the battery voltage drops below 11.6 V, the limiter disconnects the load being powered by the photovoltaic installation in order to relieve the battery; the load is connected again when the battery voltage reaches 12.6 V. Batteries do exist that can withstand deep discharge, but they are markedly more expensive than a standard battery. A graph provided by the battery manufacturer makes it possible to calculate its life in accordance with the level of discharge accepted. The heavier and more frequent the discharges, the shorter the overall battery life.

**Modes**

The drawings opposite show the system’s three different modes of operation. At the top, the battery is fully charged and powering the installation. In the center, the battery is deeply discharged. Consequently the load is disconnected and the battery is being charged by the PV panel. The lower drawing shows the battery partly charged, and connected to the solar panel as well as the load.

To complement all these notions and put them into practice under the best conditions, I’ll conclude by proposing a spreadsheet for calculating the battery life for your photovoltaic installation. Download it from the page for this article on our website [3]; it will enable you to find the optimal size for your panels, regulator, and battery.

**Web Links**


**Sources**

http://en.wikipedia.org/wiki/Photovoltaic_panel
Although the name doesn’t actually say so, a reflow oven is used to solder surface-mounted devices or SMDs. Now certain integrated circuits exist only in an SMD version, it’s more difficult for an electronics enthusiast to get by without these omnipresent SMDs. They are often cheaper than conventional through-hole components. Thanks to SMDs, printed circuits are getting smaller—and thus the cost of projects. The fact remains that for soldering these miniature components, you need little fairy fingers... or a reflow oven! You’re hesitating? Well, I’m not hesitating any more.

Before getting into the reflow oven proper, it’s worth taking a quick look at SMD technology (see inset) if this is new to you. This is also the moment to remind ourselves of the steps in producing a PCB using SMDs:

- If you use a program like ISIS/ARES, Eagle, etc. for designing the circuit, define right at the design stage the package to be used, allowing for the power to be dissipated.
- The layout itself isn’t a problem, unless from the outset you are looking for a very high density.
- The solder paste can be applied either using a mask, or by syringe. The latter lets you reduce paste consumption. Using the mask, you spread a (larger) amount of paste, then remove the mask carefully to avoid overspill.

**The Author**

Passionate about his profession, electronics, through measurement and now particularly measuring time, Jean-Pierre Duval spent most of my career in electronics and robotics in the field of medical, biological, and agro-food laboratories. With the CECOS laboratory at the Cochin hospital in Paris, he perfected the first morphological sperm counters, which are still on the market today.
After use, the paste has to be kept in the ice-box and must be heated up at least 2 or 3 hours before you want to use it.

- Place the components on the paste pads (a delicate stage!)
- Baking or reflowing.
- Cleaning the circuit using a suitable chemical.
- Magnifier inspection to get rid of the little blobs of paste that are liable to create shorts.

Making the move to SMD involves a certain amount of investment, which you can minimize by making a sum of the tools yourself—including the reflow oven itself and the squeegee for applying the solder paste. But I will just point out that the project described here is not intended to be copied exactly. It should simply provide you with the initial inspiration, just as I myself drew inspiration from earlier projects [3].

**Why fan-assisted?**

Having worked for a long time inspecting ovens, incubators and autoclaves and measuring their characteristics, I know that an oven is only usable if its temperature is uniform at all points—whence the use of a fan-assisted oven. Imagine a cell culture to be carried out in an oven at 37 °C (98.6 °F). At 40 °C (104 °F) it will be cooked; at 35 °C (95 °F) it won’t grow at all. So we need the same temperature everywhere. The same

---

### SMD technology

For easier handling, it’s best to choose components that are not too small. For resistors, just as for conventional components, the size of the device (**Figure 1a and 1b**) varies according to the power to be dissipated. The .25-watt resistor size 1206, referring to its dimensions expressed in .01 inches, is the commonest. The metric designation is 3216 (3.2 mm by 1.6 mm).

**Packages:** there is a huge variety of packages, they are usually well described in the datasheets. To use certain devices like BGAs (Ball Grid Array ICs), which have ball-shaped soldering points hidden beneath the device, you need industrial fitting tools that amateurs and even smaller companies can’t afford. Many SMDs exist in various packages, but some are only available in a single configuration.

**Examples:** SOT-23 (Small Outline Transistor) packages are often used for diodes or transistors, e.g. the BAS16 diode (**Figure 2**). For microcontrollers and other integrated circuits, some packages are easier to use than others; you can still just about handle a TQFP (Thin Quad Flat Package) with 64 pins on 0.8 mm pitch, but it becomes a real headache when you get down to a TQFP with 100 pins and only 0.5 mm pitch.

Certain components are not marked at all. The only solution: meticulous storage!
goes for an autoclave sterilizer: 121 °C (250 °F) means 121 °C (250 °F) everywhere. By stirring up the air in an enclosed space, the fan-assist avoids points of inertia and zones that are too cold or too hot. I chose an SEB Délice 18-liter oven (Figure 3). To be quite truthful, I made a mistake when I bought my oven. Or rather, I was misled: the blurb said 1500 W heater element + 750 W grill. In fact, I have 750 W + 750 W. I should point out that the fault lies not with the manufacturer, SEB, but with the store.

Is 1500 W enough? Normally no—you need at least 2000 W; but with the fan assist and by adding heat reflectors, we can obtain a temperature and ramp-up times compatible with the specifications of the solder used. As a reminder, PCB-Pool offers a 1500 W convection oven. As for the high-quality oven offered by elektorPCBServico [1], that consumes 3000 W; its dimensions, in particular its height, allow the heat to be confined towards the PCB; its curves are remarkable.

Oven geometry

Before using or even modifying the oven, I wanted to check how even the heat was, taking four points in a plane within the space used for reflow: the center of the oven and three corners of a 10 × 12 cm rectangle.

Though the ripple on the curves, caused by the hysteresis of the bimetal thermostat, may seem surprising, they are all very close, irrespective of the measurement point, and confirm the assumption of the need to stir up the air. It will be noted that when set to 240 °C, the temperature measured at the center of the oven is over 270 °C—a difference doubtless explained by the off-center position of the bimetal strip, on the oven side wall. I recorded four cycles of 15 mins corresponding to the four measurement points, and took the average of the last three. Between each cycle, I let the oven cool down for around one hour (Figure 4). I discarded the first cycle as the thermocouple had moved.

The measurements were made using a K thermocouple, calibrated against a mercury thermometer that I used to use for my oven checking.

Thoughts on the reflow curves

Before starting this project, I read up on var-
In electronics, not moving forward equals going backward!

Measuring the temperature—a difficult one!
Thermocouple, PT100, Anyone? I used a K-type thermocouple. As the oven is fan-assisted, it can be positioned under the board to be soldered. Everyone knows drafts cool down bodies. The thermocouple will indicate the temperature of the air and not the temperature of the bodies. As it’s difficult to apply an inertia factor, I added an arbitrary 20 s of time for taking the temperature between pre-heat and reflow.

Microcontroller and programming language:
I’m in the habit of using AVR microcontrollers from Atmel, programmed in Bascom BASIC; it’s reliable and perennial. I only chose the Mega644 because I had one in stock and not the Mega32 to hand. They are pin compatible, but for this application the M32 would be more than adequate.

Choice of regulation mode: Pulsewidth modulation (PWM) or not? I asked myself that question at length. After a good many tests and experiments, I decided to go for a kind of slow PWM based on the slope of the measured temperature. I do not use the PWM timer. Instead, I measure the temperature as 0% at 0 seconds and 100% at 1000 ms. Every second, the temperature is checked and the system reacts accordingly. Everything is done in the Phasefour() routine of my program, which I am offering for you to download [2].

Choice of display: While I was perfecting the program, a graphics display (128 × 64) that I had was a great help for viewing the tempera-
ture/time curve; so I kept it in. It’s a bit of a luxury—you could make do with an LCD.

**Schematic:** Only a few active components: the microcontroller, the graphics display, the Max6675 thermocouple converter, the solid-state relay, represented on the circuit by the terminal strip. Mine is a big relay that I had salvaged; this sort of solid-state relay, with no moving parts, has the advantage of switching at the zero-point of the phase voltage, meaning a total absence of interference.

I added a clock crystal so the durations could be controlled by a timed interrupt (Figure 7), two LEDs—one red (5 V supply indicator) and the other green (switching of the solid-state relay)—and filtering capacitors on the 5 V supply. There is no 230 V AC line voltage in the electronic regulation part. I used a commercially available switch-mode power supply (5 V, 500 mA). The 230-V AC line voltage is applied via the original power cord; one wire though is connected to the large solid-state relay mounted on the oven car-cass, the other goes via a fuse (8 A). For 120-V ovens a larger fuse is applicable.

**Modifications**
The modifications to the oven wiring involve removing the fan-assist/convection switch and the bimetal thermostat, and lastly, replacing the original timer with a switch (Figure 8a). I also

Figure 7.
Outline diagram of my control board.
fitted the solid-state relay (40 A, although 10 A would have been quite enough!) to the inside of the rear cover, a connector for the 5 V rail, along with a fuse holder and an 8-A fuse (Figures 8b and 8c).

For improved efficiency, I modified the oven door. As manufactured, this consists of two glass panes spaced about 1 cm apart to avoid users burning themselves on the outside. To recover as much of the radiated energy as possible, I lined the inner wall with a reflector made from household aluminum foil, with the shiny side inwards. As the back of the oven includes a cavity to allow you to use round dishes, here too, in order to recover the maximum radiated heat, I inserted vertically a disposable aluminum freezer tray.

**Operation**

At start-up, the oven is in standby. Pressing the menu button takes us to the first choice: Launch corresponds to an immediate launch using the parameters in memory. Parameters gives you the choice to modify or view. Then depending on the choice, the user will be able to view or modify, for each phase, the temperature and the slope in % (Figure 9a) and (Figure 9b). By experiment, and paradoxically, the Pre-heat slope is 100 %—we need to overcome the oven’s inertia.

Reflow sequence: For each phase, there are two parameters: the temperature and the slope, which can be changed depending on the solder paste, and which are saved in Flash memory, where they remain available even after a loss of power. The temperature settings copy the indications given with the paste used (see above). The slope corresponds to the speed at which the temperature rises.

A few screenshots of the reflow sequence (Figure 10): in these screens, Ts = time in seconds, t.d = temperature in degrees. At the end of the cycle, the oven stops heats and an audible signal indicates it’s time to open the door.

**Results:** The soldered SMDs are then examined under a magnifier: No micro-blobs? Are the solder joints shiny? Then everything’s OK.

Figure 11a shows the board that drives my oven while it was being assembled.
My first PCB soldered using the oven is a configurable resolution I²C thermometer from Maxim-IC (DS75) (Figure 11b).

**The program**

Apart from the ATMEGA 644, I’ve used the MAX6675, which compensates the reading of the thermocouple junction and converts it into a 16-bit digital value, of which 12 bits can be used for the temperature. One bit signals a thermocouple error: absence of thermocouple, for example. It uses the SPI protocol.

The program written in Bascom BASIC (thanks to EX4 on the Bascom forum who sent me the MAX6675 section) consists of a main module, fourRefusion5.bas, and two sub-programs, SubandFunction.Bas and FonctionFour.bas.

Interesting, isn’t it? Are you still hesitating? Allow yourself to be tempted by the experiment, you won’t regret it—once you start doing reflow, electronics takes on a whole new dimension.

(140031)

**Web Links**

[1] eC-mate reflow oven from elektorPCBservice: http://goo.gl/3gIZrn
[3] Elektor Reflow Soldering Oven
   Elektor, January 2006: www.elektor-magazine.com/050319

NEW!

A Must-Have for All Elektor Fans!

£77.95 • € 89.00 • US $121.00

Order Today at www.elektor.com/zeroes
Today let’s try a larger, more complex design in DesignSpark PCB that uses multiple schematic sheets.

So far I’ve been using small design examples that fit on a single schematic sheet. Today let’s try using multiple schematic sheets in DesignSpark PCB like you would for a larger design.

Creating multipage schematics
I usually like to partition larger designs into multiple schematic pages to help break it up into multiple logical blocks. Each page has its own unique net list and connections between sheets are done with “ports” or “sheet connectors” depending on the PCB package. DesignSpark PCB can do all of this with a little effort.

Once a schematic has been added to a project file, DesignSpark will make sure that it has unique reference designators across the entire design and will warn you if it finds any duplicates. Any added components will also get unique designators. Net connections on individual pages are also assigned unique names but you can create net connections between sheets by giving it a name which makes it global across the project. The name can be anything but I recommend that it doesn’t start with “N” followed by numbers which is the default DesignSpark naming convention. DesignSpark already includes power and ground components which automatically create global power nets. Although you can use custom net names to join nets across pages I find it makes a schematic very difficult to follow and understand. So instead we’ll create “port” symbols like in Figure 1 which shows an example of an input port, bidirectional port and an output port. They are just a simple one-pin component with an arrow to indicate the signal direction.

You use the ports by placing them on a schematic page and wiring them appropriately. DesignSpark doesn’t allow library components to show net names automatically at a fixed position so you will need to do that manually. When you right click on a wire, you can select the “Display Net Name” to show the net name and then manually move it how you would like it. Don’t forget to change the net name to a custom one by right clicking on the net name and selecting the “Change Net” option because...
DesignSpark will automatically hide all of the default net names in the “Change Net” window making them impossible to select.

Let’s use the two-page schematic in Figure 2 for the rest of our discussion. I’ve kept it simple so it’s easy to see how net connectivity works in DesignSpark.

Creating the PCB
Now let’s import our design into a PCB to make sure that the PORTA and PORTB nets are connected properly. First you create the PCB file using the File->New command and checking the “Add to Open Project” box which will give you a blank PCB. Now open one of the schematic pages and use the “Tools->Forward Design Changes” menu to add the components and nets to the PCB. After a little editing I got a PCB like in Figure 3. I labeled all of the net connections to show that the netlist was properly imported, including the PORTA and PORTB nets.

Cross probing nets
Our example multiple page schematic is pretty simple so it’s easy to remember the entire schematic while working on it. But this becomes an issue for large complex designs when you are trying to understand a net connection and you want to do something like match a net connection in a PCB to a schematic. This is called cross probing the nets and it’s one of the new features in DesignSpark PCB version 6.0.

So how do you cross probe a net in the new version? Use the “Edit->Cross Probe” menu to go into cross probe mode. Now every net connection you click on a net in the PCB will automatically be selected and shown in the schematic. If you click on a net in the schematic then it will be selected and shown in the PCB as well. Once you’re done probing nets the “Edit->Cross Probe” menu will exit cross probe mode.

The PORTA and PORTB sheet to sheet connections in our example project are a special case for cross probing. When you select one of those nets in the PCB, DesignSpark will highlight all of the PORTA/B connections but it will only open one of the schematic pages for viewing. The easy way to see all of the PORTA/B connections is to go into cross probe mode in the schematic sheet and select the net. This makes DesignSpark open the reset of the schematic pages that contain that net and display them all simultaneously. For example, Figure 4 shows how DesignSpark highlights the PORTB net when I clicked on it from the PCB with this procedure.

Web Link

Figure 3. Imported PCB.

Figure 4. Cross probing nets.
Varactor Diodes

Weird Component #6

By Neil Gruending
(Canada)

If you’ve ever touched any RF circuitry there’s a very good chance it contained at least one variable capacitor in its voltage controlled oscillators (VCOs) and filter circuits. They used to use mechanical variable capacitors but now varactor diodes, also known as varicaps, are preferred because of their small size, high performance and excellent tracking ability. Figure 1 shows the schematic symbol for them.

![Figure 1. Varactor (varicap) schematic symbol.](image1)

Figure 1.
Varactor circuit examples.

Figure 2.
Varactor diode circuit examples.

All semiconductor PN diode junctions have a small amount of variable capacitance when they’re reversed biased. The reverse bias on the diode PN junction creates an insulator region at the junction that creates a small parallel plate capacitor. The insulator width will increase with the bias voltage which decreases the effective capacitance. This is what gives diodes their variable capacitance. Varactor diodes, however, are specifically optimized for their capacitance properties and have two distinct types based on their adjustable range. Abrupt varactor diodes change a small amount of capacitance for a change in bias voltage which gives them high $Q$ (quality factor) and low distortion properties. Hyperabrupt diodes on the other hand have a much larger change in capacitance over the same bias voltage change which can increase distortion. Abrupt diodes can have a capacitance range of 2:1 whereas hyperabrupt diodes typically have a capacitance range of 10:1. In the past, power varactors of the hyperabrupt variety were used as frequency multipliers (mostly triplers), providing a cheap way for radio hams to produce a few watts of RF power on UHF like 70 cms (144 MHz × 3) or SHF like 23 cms (430 MHz × 3). The method is suitable for non-linear modes only, i.e. FM and CW.

![Figure 2. Varactor diode circuit examples.](image2)

Figure 2 shows two varactor diode circuit examples. Both circuits are using the varactor diode to adjust an LC tank circuit by changing the diode reverse bias voltage $V_{\text{tuning}}$. Figure 2a is a simpler design and works well with low amplitude signals. However, when the signal gets too large it will affect the diode bias voltage and cause distortion. Figure 2b uses two diodes to counteract the distortion at higher signal levels. This is because as the signal causes one diode to decrease its capacitance the other one will increase its capacitance by the same amount. The downside is that the two diodes in series will have half the capacitance which is significant when we’re talking about capacitors in the picofarad range.

The varactor diodes don’t need a lot of bias current so the value for $R$ isn’t critical as long as it provides enough isolation between the bias voltage and the tank circuit. Sometimes you will see an inductor used instead, but its function is the same. It’s also important to keep the diodes reverse biased because they will create a lot of spurious noise if they start to forward conduct. If you want to experiment with varactor diodes but don’t have any, try a regular diode from your parts box. Lots of people have had success using 1N400x diodes, power rectifier diodes, zener diodes and even LEDs. Just don’t give up—it may take several tries to find a diode that works!

(140026)
Elektor PCB Service at a glance:

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

Delivery from 2 working days

Further information and ordering at www.elektorpcbservice.com
3D printing is all the craze. Even national TV networks covered the subject and today you can buy a 3D printer on almost every street corner. And yet, with all these printers around, with all those enthusiasts busy printing in 3D, I still have to see the first really useful 3D-printed output. Many 3D printers are used to print parts to build another 3D printer. It is almost a self-replicating machine, and it is probably fair to say that 3D printers have brought mankind one step closer to this self-imposed goal. Printing such parts is nice, but why would you do that if you already have a 3D printer? “So that someone else can build one too!” is the habitual answer. I see. Another popular job for 3D printers is to print “things” — all kinds of small objects, artful or not, that serve no other purpose than showing off the 3D printer’s capabilities.

Technology forecasters predict that in a few years’ time most households will own a 3D printer and they may very well be right. I do find this a bit worrying though, because what are all these people going to print? Exactly! Lots of objects and things that have no other purpose than to keep their 3D printer busy.

The “ink” used by most 3D printers is a kind of plastic wire—actually, a polymer filament, which implies that their output consists of polymer objects. According to Wikipedia, common polymers used in 3D printing are acrylonitrile butadiene styrene (ABS), polycarbonate (PC), polylactic acid (PLA), high density polyethylene (HDPE), PC/ABS, and polyphenylsulfone (PPSU)—available in brilliant colors, of course.

Probably, most of the polymer objects we are going to print in huge quantities will sooner or later end up as 3D printed waste, resulting in tons of high-tech waste besides our obsolete cell-phones. Already, projects have been launched to find ways to create the polymer filament from waste products, so we will not run out of “ink” soon. But wait a minute—wouldn’t it be much simpler and cheaper to print the polymer filament with our 3D printer? Then we can use the printer to not only print itself, but also its own ink and we have a truly self-sustaining system. All we have to do is to put energy in it to keep it running. Lots of energy, I admit, but well spent, because the 3D printing circus has a high amusement factor.

After this rant I dare you to 3D print something useful and I don’t mean a coat hanger or some other trinket or gizmo you can buy of better quality and for less money than you can print it. Please post photographs of your useful 3D printed objects on Elektor.Labs (or send them to labs@elektor.com and we will put them on the web for you). The photograph of the most useful object (subjective, I know!) will win something nice, I promise. Meanwhile here is a photo of a useful object we printed for one of our upcoming projects. The other photo shows other products of a less useful variety.

By Clemens Valens (Elektor.Labs)

3D-Print Your Own Ink

A useful (to us!) 3D-printed object to keep two PCBs together.

Officially, none of these objects were 3D-printed at Elektor Labs.

Many 3D printers are used to print parts to build another 3D printer. It is almost a self-replicating machine, and it is probably fair to say that 3D printers have brought mankind one step closer to this self-imposed goal. Printing such parts is nice, but why would you do that if you already have a 3D printer? “So that someone else can build one too!” is the habitual answer. I see.

Another popular job for 3D printers is to print “things” — all kinds of small objects, artful or not, that serve no other purpose than showing off the 3D printer’s capabilities.

Technology forecasters predict that in a few years’ time most households will own a 3D printer and

The “ink” used by most 3D printers is a kind of plastic wire—actually, a polymer filament, which implies that their output consists of polymer objects. According to Wikipedia, common polymers used in 3D printing are acrylonitrile butadiene styrene (ABS), polycarbonate (PC), polylactic acid (PLA), high density polyethylene (HDPE), PC/ABS, and polyphenylsulfone (PPSU)—available in brilliant colors, of course.

Probably, most of the polymer objects we are going to print in huge quantities will sooner or later end up as 3D printed waste, resulting in tons of high-tech waste besides our obsolete cell-phones. Already, projects have been launched to find ways to create the polymer filament from waste products, so we will not run out of “ink” soon. But wait a minute—wouldn’t it be much simpler and cheaper to print the polymer filament with our 3D printer? Then we can use the printer to not only print itself, but also its own ink and we have a truly self-sustaining system. All we have to do is to put energy in it to keep it running. Lots of energy, I admit, but well spent, because the 3D printing circus has a high amusement factor.

After this rant I dare you to 3D print something useful and I don’t mean a coat hanger or some other trinket or gizmo you can buy of better quality and for less money than you can print it. Please post photographs of your useful 3D printed objects on Elektor.Labs (or send them to labs@elektor.com and we will put them on the web for you). The photograph of the most useful object (subjective, I know!) will win something nice, I promise. Meanwhile here is a photo of a useful object we printed for one of our upcoming projects. The other photo shows other products of a less useful variety.

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

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

Further Information and Ordering at www.elektor.com/flowcode
We suspect Labs trainee Arne Hinz, student of Electrical Engineering at RWTH Aachen, Germany (photo) has designed what is believed to be the largest PCB ever to originate from Elektor Labs. This first prototype of the board measures an amazing 37.5 × 19 cms (14.8 × 7.5 inches). After fitting it into a case, thoughts arose to make the board even bigger to facilitate easy mounting onto the available screw mounts.

What does it do, you ask? Well, it’s a beefy single-phase to three-phase AC converter with adjustable frequency. Final specifications are subject to change, but the initial goals were:

- frequency adjustable from 0 to 400 Hz;
- output power 3 kW.

With this first prototype, the input voltage is limited to 230 VAC, but other line voltage options like 120 VAC are not ruled out and can be implemented at a later stage of the design cycle.

Though this project will not be ready tomorrow, we are planning on releasing the plans later this year rather than next year, so keep an eye on upcoming magazines if you are interested in this design!
We’re celebrating a new look — and the next 25 years.

25% OFF CIRCUIT CELLAR

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

JOIN TODAY!

www.circuitcellar.com/sepN13
Chip Tips: The MAS6510 Capacitance-to-Digital-Converter

Many modern sensors express their measurements in the form of a change of the sensors capacitance or inductance. Converting these changes into digital values usually requires a special interface and complex electronics. In-house solutions using a microcontroller require not just programming skills but also analog know-how and usually, a bit of tinkering. It’s good that there are also some tailor made ICs available to do the job. The chip used here is a Capacitance to Digital Converter (CDC); it needs nothing more than connection to the sensor. There are a few features to consider for reliable and stable operation.

The IC MAS6510 is supplied by the ‘fabless’ semiconductor manufacturing company Micro Analog Systems Oy based in Finland. This particular chip is only supplied in a QFN16 package outline measuring 4 x 4 x 0.75 mm. It features a Delta-Sigma-converter with 24-bit resolution suitable for interfacing with capacitive pressure and humidity sensors in MEMS technology (see Figure 1). Sensors using the change of capacitance of a single capacitor connect to pins CS and CC (see Figure 2) of the MAS6510. The chip also has a CR input for dual capacitance sensors. More details of this configuration can be found on the data sheet [1]. MAS also produce a simplified variant of this chip with ‘only’ 16-bit resolution and a version specifically for piezoresistive sensors. All variants use the familiar SPI and I²C bus interfaces.

The MAS6510 block diagram in Figure 2 shows two internal trimmer capacitors at the sensor inputs CS and CR. These capacitor matrices are used for sensor-offset calibration and their values are software programmable. These parameters along with parameters defining the clock division ratio and operating mode are stored in an internal 32 Byte EEPROM. The chip’s I²C address is also stored here, leaving 27 bytes available for storage of any other application specific data such as sensor trim and calibration values. The data sheet gives detailed information about memory usage and register description. The principle mode of operation is governed by the type of sensor in use whether it has one or two capacitors at its output: it can measure the capacitance of a single capacitor or the ratio of two capacitors (CS - CR)/CS or the difference between the two capacitor values CS – CR. The measurement range extends to 20 pF (see Table 1; view original specs at [2]). Use of the internal clock divider allows this base capacitance range to be increased. A capacitance difference of 20 pF is allowed in differential mode which can also be extended by using a lower clock frequency. Some sensors that use the ratio of two capacitors employ linearization to correct the otherwise non-linear 1/C characteristic. The internal divider can divide the clock by a maximum of 1/8. The clock can optionally be derived from an external source.

Supply voltage to the chip is in the range from 1.8 to 3.6 V. The operating current is dependant on supply voltage and the sample rate ranging from a minimum of 3.3 to 48 µA at one sam-

Figure 1.
A miniature capacitive pressure sensor from VTI technologies.
ple per second. Between measurements most of the power-hungry chip functions are switched to sleep mode. The maximum conversion rate is dependent on the clock rate between 12 and 173 Hz. The actual usable resolution will always be 17.5 bit. The MAS6510 can also measure chip temperature and VDD supply voltage for use as measurement compensation to improve accuracy.

Web Links

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conditions</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Voltage $V_{DD}$</td>
<td></td>
<td>1.7 V</td>
<td>2.7 V</td>
<td>3.6 V</td>
</tr>
<tr>
<td>Average current Consumption $I_{DD, AVG}$ (resolution dependent)</td>
<td>1 conversion/s; Cap. Dif. Reg OFF</td>
<td>3.3 µA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conversion time $t_{CONV}$</td>
<td>Normal clock (SOSC=00); OSR=256</td>
<td>5.8 ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal system clock oscillator frequency $f_{SYS, CLK}$</td>
<td>Normal clock (SOSC=00)</td>
<td>200 kHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor excitation frequency $M_{CLK}$</td>
<td>Normal clock (SOSC=00)</td>
<td>50 kHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serial Bus Clock Frequency $f_{SCL}$</td>
<td>I2C bus; SPI bus</td>
<td>400 kHz</td>
<td>2 MHz</td>
<td></td>
</tr>
<tr>
<td>Changing capacitance range in capacitance difference mode $\Delta C_{DIFF}$</td>
<td>Division by 2 (SOSC=01)</td>
<td>2 pF</td>
<td>30 pF</td>
<td></td>
</tr>
<tr>
<td>Maximum sensor capacitance in capacitance difference mode $C_{S, MAX}$</td>
<td>Division by 2 (SOSC=01); int. oscillator</td>
<td>40 pF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changing capacitance range in capacitance ratio mode $\Delta C_{RATIO}$</td>
<td>depending on sensor characteristics</td>
<td>2 pF</td>
<td>20 pF</td>
<td>(&gt;20 pF)</td>
</tr>
<tr>
<td>Operating Temperature $T_A$</td>
<td></td>
<td>-40 °C</td>
<td>+25 °C</td>
<td>+85 °C</td>
</tr>
</tbody>
</table>
Complex electrical wiring is required if a single load like hallway or corridor lighting is to be switched on and off by any one of a large number of switches. Here we propose a workaround using a DIY thyristor.

Looking at the schematic in Figure 1, imagine the AC grid voltage to the circuit arriving on K1 via a chain of pushbutton switches of the normally-closed type. The switches can be any number from two in a small hallway to tens distributed along endless hotel corridor walls like we saw in *The Shining*. Electrically the switches are in series in the Neutral or Live line. LA is the load, it is powered through relay contact RE1B and connector LA1.

Initially the circuit’s AC supply voltage arrives on rectifier B1 through series capacitor C3 (or C3||C4 for 115 VAC grids) shunted by R6/R7, and R5. Zener diode D3 at the rectifier output limits the voltage on the control circuit to about 15 V. Resistor R4 and capacitor C2 (which has a relatively low capacitance) shunt the zener diode. Capacitor C1 (with a high capacitance—1000 μF) is connected via diode D2. The quasi thyristor formed by transistors T1, T2 and relay coil RE1A is effectively in parallel with C1. The ‘control electrode’ of the thyristor-analogy is connected to the cathode of D2.

When any of the pushbuttons (not drawn) in the AC supply voltage chain is pressed, capacitor C2 discharges rapidly and the control electrode of T1-T2 is effectively connected to C1. The DIY thyristor ‘fires’ thanks to capacitor C3 discharging. Consequently the relay contact is energized, switching on (lamp) load LA.
When the pushbutton in the chain is released, the AC grid voltage is applied again to the circuit, causing a charge voltage to C1 to be upheld and the ersatz thyristor to keep conducting. To turn off the lights, users press any of the pushbuttons in the corridor and hold it down for about a second. This causes capacitor C1 to discharge completely, T1-T2 to be reset, and the relay and consequently the load to be switched off.

**Construction**

The PCB designed for the project appears in Figure 2. It is compact at a size of 54.4×45.5 mm. It might just fit inside a flush-mount wall electrical box but certainly inside a junction box. If necessary the PCB corners can be filed off. The relay is a type with low power consumption and a small footprint (approximately 5×20 mm). The high sensitivity coil type of the 12-V version device from Omron’s G6DS series from only requires 10 mA nominal coil current to activate—it has one normally-open (NO) contact. In practice, this relay should actuate at about 70% of the nominal value. The single, normally-open, relay contact can handle up to 5 A with a resistive load, or 2 A with an inductive load.

The lead spacing of the capacitors in the AC grid voltage divider (C3 and C4) is either 15 mm or 22.5 mm. Their values are not the most common for X2 class capacitors. C4 should only be mounted if the AC grid voltage is 115 VAC.

![Schematic of the Multi-switch Corridor Lights Control.](image)

**Component List**

<table>
<thead>
<tr>
<th>Resistors</th>
<th>Value</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1, R2</td>
<td>510 kΩ 1%, 0.25 W</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>4.7 kΩ 5%, 0.25 W</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>2.0 kΩ 1%, 0.25 W</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>620 Ω 1%, 0.5 W</td>
<td></td>
</tr>
<tr>
<td>R6, R7</td>
<td>470 kΩ 5% 0.25 W</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacitors</th>
<th>Value</th>
<th>Pitch</th>
<th>Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1000 µF 20% 25 V, 0.2” pitch, 10 mm max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>3.3 µF 20% 100 V, 0.1” pitch, 6.3 mm max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>330 nF 20% 275 VAC, Class X2, 15 or 22.5 mm pitch, 11 mm max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>220 nF 20% 275 VAC, Class X2, 15 or 22.5 mm pitch, 8 mm max</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semiconductors</th>
<th>Value</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>W06MG</td>
<td>WOBM</td>
</tr>
<tr>
<td>D1, D2</td>
<td>1N4148</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>BZK55C15</td>
<td>15 V zener diode, 0.5 W, DO-35</td>
</tr>
<tr>
<td>T1</td>
<td>BC547B</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td>BC557B</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Miscellaneous</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1, LA1</td>
<td>2-way PCB mount screw terminal block, 7.5 mm pitch</td>
</tr>
<tr>
<td>RE1</td>
<td>relay, G6DS-1AH 12DC, PCB mount, SPNO, coil 12 DC / 10 mA, contact 250 V AC / 5 A</td>
</tr>
<tr>
<td>PCB #</td>
<td>130179-1 v1.1</td>
</tr>
</tbody>
</table>

* use with 115 VAC only

---

**Caution.** Parts, connections, wires and copper tracks in this circuit are at live potential. The circuit must be enclosed in an approved case observing all relevant precautions in terms of electrical safety. Never work on the circuit with AC power connected to K1.
HABEY USA, a leading manufacturer of embedded computer and a Freescale Connect Partner, announces The HIO Project, an open hardware project to enable developers and users to have access to an affordable Freescale i.MX6 ARM Cortex A9 Processor based open architecture platform. The core component of the HIO Project is the HIO Mainboard, model HIO-EMB-1200. It is a Post-it sized, Lego-like ARM based open modular platform for rapid prototyping, quick application module development and fast time to market. Unlike the traditional Computer-on-Module (COM) or PC/104 standard, the HIO Mainboard is a fully functional ARM computer board with on-board Freescale i.MX6 ARM processor, RAM, iNAND flash, HDMI, USB and power input from mainboard or add-on boards. Measuring just 72mm x 80mm, the HIO-EMB-1200 is based on HIoTX, a newly developed form factor by HABEY USA for the HIO Project. It is 80% smaller than the mini-ITX form factor and 20% smaller than the tiny PICO-ITX standard. The HIoTX form factor leverages the Freescale i.MX application processor’s rich I/O feature and flexibility. The HIO Mainboard’s unique low-profile modular design allows it to fit into space-restrictive applications while still allowing 3-dimensional expansion utilizing its 200-pin female header on the top and bottom of the board. It provides an fresh platform for the latest generation of small, smart, interactive and connected devices in the new Internet of Things.

The Ethos lite is a tiny unmanaged Ethernet switch that has 5 ports, supports 10/100 Mb data rates on a board measuring just 1.75” x 1.75” and weighing only 15 grams. One primary area where the Ethos lite stands out is in situations where weight and power are high priorities such as Unmanned Aerial Vehicles or robotics. All 5 ports are fully shielded with magnetic couplers for common mode rejection meeting Ethernet wiring specifications. This allows designers the ability to skip the large RJ45 connectors and wire directly onto connected peripherals or place the connectors in a more flexible location. The Ethos board can accept a wide range of voltage input from 5-15V making it an ideal fit for Unmanned Systems that provide battery power. The Ethos board can also double as a 3.3 V regulator providing up to 1 amp of current. Under full load (all 5 ports actively sending data) the Ethos consumes less than 2 watts. Connections to the Ethos lite are provided by Molex PicoBlade headers with one for each port and an additional connector for power. As an added feature the MDIO lines to the Ethernet switch IC have been exported allowing developers access to management data registers. Included with the Ethos lite is the main board along with 6 Molex PicoBlade cables (5 with RJ45 connectors, 1 with bare wires for power). The Ethos lite costs $250 with the ability to purchase extra Molex cables in a variety of configurations.

Triangle Research International Inc. (Tri) now gives OEMs an easy and affordable way to custom-build an in-house PLC for their own line of equipment products. The SmartTILE is a PLC CPU-board that plugs onto a carrier I/O board which an OEM is free to design, the combination being a fully integrated PLC optimized for the OEM’s current and future needs. Compared to building a controller from scratch each time, this significantly shortens the OEM’s time-to-market for all new product line releases, saves money from reduced assembly manhours and unnecessary components that come with off-the-shelf PLCs, and allows for a consistent standard for post-sales equipment troubleshooting, maintenance and related-training. Programmed in the powerful iTRILOGI Ladder+BASIC language, the SmartTILE includes an integrated Ethernet port and carries all the capabilities of Tri’s established F-series PLCs, including floating-point math.
Sensirion’s Liquid Flow Sensor Flies into Space

On May 6, a rocket will lift off bound for the International Space Station (ISS). On board will be a liquid flow sensor from Sensirion AG, Stäfa, Switzerland. The sensor is part of a research project by Minnehaha Academy in Minnesota that is investigating the impact of microgravity on the effectiveness of liquid flow. Among other potential findings, the project aims to shed light on the effects of weightlessness on the circulatory system. The LS16 liquid flow sensor from Sensirion on board the ISS will measure the flow of demineralized water generated by a piezoelectric pump in zero gravity, and compare the results with those of a control experiment on earth. Numerous applications in fluid dynamics, physics, biology and hemodynamics (the forces involved in the circulation of blood) will benefit from these findings. The research team is extremely satisfied with the sensor. “Our experience with the Sensirion LS16 sensor has been outstanding – the LS16 has been accurate, precise and durable,” says Max Thompson, Student Project Manager at Minnehaha Academy. But Sensirion did not just supply the sensor for the experiment, its developers also helped out in an advisory function. For instance, the LS16 liquid flow sensor had to be modified for the launch: to ensure that the sensor could withstand the enormous forces during lift-off, Sensirion’s development team replaced the capillary glass tubes with robust capillary steel tubes. “Our project would not have been possible without the generosity, time and amazing commitment from Sensirion to make the flow sensor flight-ready,” says Max Thompson. Sensirion stands for innovation. Existing and new products from Sensirion continue to set standards and show that in the measurement of temperature, humidity, gas and liquid flows using sensors, Sensirion’s expertise and innovative drive have made it number one worldwide. The research team at Minnehaha Academy was delighted to have Sensirion’s support: “Sensirion has been a champion of companies to have as a partner to make this student-designed project a reality.”
Exar: Wide Frequency Universal Clocks with Ultra-Low Jitter
Exar Corporation (NYSE: EXAR), a leading supplier of high performance analog mixed-signal components and video and data management solutions, today announced an addition to its range of telecommunications timing products with a new family of Universal Clocks. The XR811xx series offers a wide range of output frequencies from 10 MHz to 1.5 GHz, with ultra-low phase noise jitter of less than 200 fs. Designed for communications, audio/video and industrial applications, the QFN-10 and TSSOP-8 packaged devices are footprint compatible with industry standard synthesizers, providing a superior performance second source option. The design of Exar’s XR811xx synthesizers utilizes a highly flexible delta-sigma modulator and a very wide-ranging VCO in a PLL block that has been optimized to be extremely power efficient. With a core current consumption of just 20 mA, these parts dissipate 60% less power than equivalent competitive devices, providing a very compelling power efficiency benefit to system designers. The PLL can operate from either an input system clock or a crystal and incorporates both an integer divider and a high-resolution (<1 Hz) fractional divider for increased flexibility to generate any clock frequency. Additionally up to four different frequency multiplier settings can be stored allowing for different application configurations providing BOM savings compared to multiple synthesizers. The XR811xx family also offers a choice of LVCMOS, LVDS or LVPECL output drivers.
Exar’s Universal Clock devices deliver extremely low, sub-200 fs, output phase noise jitter as integrated over a PLL loop bandwidth of 1.875 MHz to 20 MHz. This spans the requirements of most WAN and LAN systems and supports communications standards including: 10 GHz Ethernet, 2.5 GHz and 10 GHz SONET/SDH/OTN, xDSL and PCIe, as well as many other synchronized clock system applications.

www.exar.com (140116-IV)

Semi-Custom Li-Ion or Li-Polymer Soft Pack Battery Solutions
Announced recently by VARTA Microbattery is the CellPac BLOX system. The CellPac BLOX system is a design program implemented to offer OEMs a semi-custom power solution when off-the-shelf lithium ion (li-ion) or lithium polymer (LiPO) soft pack battery assemblies do not meet the design criteria. A dedicated CellPac BLOX engineering staff is specifically trained to help accelerate time-to-market and minimize non-recurring engineering (NRE) by consulting with customers throughout every phase of the design cycle. Battery packs provided include those optimized to satisfy the most basic as well as the most complex requirements, in addition to those with high or low capacity specifications.
Comprised of standard cells and PCM (Protection Circuit Module) components, VARTA Microbattery’s semi-custom soft pack battery assemblies are ideal for a broad range of portable industrial, medical and data communications instruments utilizing embedded rechargeable batteries. They are frequently employed in assorted medical equipment, ultrasound, health monitoring and diagnostic devices. The batteries are equally suitable in industrial applications such as barcode scanners, measurement, control products, telecom and data communication devices. The multi-cell pack assemblies are available with li-ion cylindrical, prismatic and pouch cells, and packaged in a soft pack design or shrink sleeve. The nominal capacity of the battery packs range from 150 mAh to 9000 mAh, with voltages ranging from 3.7 V to 11.1 V. Customers may specify wire length, connector type and labeling requirements. Devices are available in prototype to production quantities.

www.varta-microbattery.com (140116-VI)
Want to talk to us directly?
Share your interests and opinions!
Check out our New Social Media Outlets for direct engagement!
Back in 1986 at the crowded Elektor Electronics laboratory, somebody approved the idea proposed by a young Dutch BSc. graduate of Electronics to make an oscilloscope add-on for personal computers. I had been appointed as an assistant editor just months before, and being able to wield an electric typewriter and an English Dictionary I was asked to write a deft letter to ITT Semiconductors of Freiburg, Germany, asking for a few samples of their latest UVC3101 device.

I was told it was a "high-speed" ADC and DAC in one chip and “difficult to get”. I got a favorable reply by telephone from Reinhard W. Preuss, the most officious and gentlemanly PR and Press Officer I have come across in the electronics scene these past 25 years. A few days later the ICs arrived, complete with engineering notes. The lot allowed Rene van Linden to start designing his ComputerScope project and hopefully complete his BSc. thesis.

**Time perspective**

In those days it was hot to make computers do things other than the usual spreadsheeting, PacMan’ing, word processing and ASCII art doodling. It was the exciting era of the BBC Micro, the Acorn Electron, Philips MSX and the first IBM PC clones starting to appear in households and in electronics engineers’ workshops. It was a time also when computers were totally accessible in terms of hardware and software. As long as a computer had a port of some sort using an easy to obtain connector, engineers, British school children and Elektor trainees got into developing peripherals the computer manufacturers could not have foreseen. There was a lot of POKE(16xxx) going on at that time.

In the case of the Elektor ComputerScope, the idea was to make a personal computer handle the visualization of measured data on its monitor with composite video input. Paradoxically, that monitor had a CRT just like the traditional oscilloscope, only larger and more colorful, so the crux or cost advantage of the computerized ‘scope over, say, a refurbished Tek I find hard to grasp.

In good Elektor tradition for large projects from the lab, the magazine article consisted of two instalments. The first in September 1986 gave the circuit diagram, the basic method of operation and some screen dumps showing not too complex waveforms. In the October 1986 edition the construction, adjustments and software aspects were discussed, and the project was carried on the front cover of the magazine.
(Figure 1). Note the visually implied hierarchy of two home computers pictured: the BBC Micro majestically in front and a Commodore C64, well, also visible.

The design
As a peripheral the ComputerScope requires a fast and wide highway into the computer’s memory and display systems. However, these are digital, while the scope probe will pick up analog signals mostly, so ADC-ing is called for. The port traffic is not one-way either, as the scope input settings including the AC-DC selector and attenuator/amplifier need to be controlled from the computer. A couple of Clare DIP relays were used for the purpose. There are no controls or knobs on the Elektor ComputerScope to fiddle with, hence the instrument looks insipid in its grey case except perhaps for the stylized S.

Compatible on paper with Acorn Electron, BBC Micro, C64, MSX, IBM PC?

The crux of the circuit is found in the “fast” ADC and DAC inside the UVC3101, and a small memory section based on Inmos IMS1420 RAMs. These parts were rare and expensive at the time—fortunately some advertisers in the magazine started to supply kits of the project.

The ADC and RAM section of the schematic is reproduced in Figure 2. The DAC (not visible here) was used to add an offset to the input signal if required. The datawidth from the ADC was 7 bits; bit 8 was the trigger indicator. The memory area actually used for data buffering was 512 bytes. The RAM section had its own address counter built from TTL ICs from the 74HC series (no microcontroller in sight). The board was packed with 74HC ICs for that matter, and a few straggling 74LS and CD4000 ones, since a mass of control and timing functions had to be catered for—like the complete timebase. I counted 24 logic ICs (Figure 3), which would be labeled a component grave these days.

The lot was powered by a separate supply board type EPS 9968-5.
A small modification involving a slide switch on the back panel (Figure 4) allowed the instrument to be used with IBM (clone) PCs also. Those of you yearning for the complete article from 1986 are advised that the Elektor 1980’s DVD is about to be published; watch www.elektor.com or your weekly E-Post Newsletter.

**Those Listings**

If you ordered the ComputerScope board # 86083 through the Double Dutch named Elektuur Print Service (EPS) it came with program listings on paper for the Acorn Electron, the BBC Micro, the Commodore C64, and MSX machines. I remember stacks of this paperware on the shelves in our stores, alongside the boards, and supplications on my desk from readers all over the globe to send them the listings on disk, or without the board! To help owners of “other makes of computer” to compile their own program for the ComputerScope, a pitiful flow diagram was printed at the end of the October 1986 instalment.

**Love story**

Now compare the names of our two editorial secretaries duly printed in the October 1986 edition with those in November 1986—look on page 13. One surname changed. Besides passing his BSc. Electronic Engineering exam with good notes for the ComputerScope project, lab trainee R (for Rene) v(an) Linden, a silent, almost invisible worker, in the course of 1985 also managed to more than befriend “our” cherished secretary W (for Wilma) Wijnen.

**Dumpster story**

Elektor moved offices in 2006 and in an orchestrated attempt to “shake off the dark past” the technical staff and editors got strong advice to clear their stables and make a fresh start at Elektor Castle. One Friday afternoon, CEO attending, the old lab space in Beek (Holland) was cleared out rigorously by quasi volunteers, with a ton of stuff disappearing in an x cubic feet dumpster carefully parked on the driveway right under the lab windows. There was beer and pizza to celebrate the almost vacant building, and reward the workers.

Of the twenty or so lab projects and prototypes I was able to safeguard during Operation Augias, most—I admit—were of the ‘finished’ type, meaning there was a case around them. I managed to keep these out of harm’s way and was happy to see that a few colleagues did the same for sentimental reasons. Today the small booty of that boisterous Friday afternoon is incorporated in the Retronics Collection, which is an “asset to the company” under the current management, and virtualized at www.elektor-labs.com/attic. Look at the item on the shelf straight above the Mugen Hybrid Amp—that’s the final resting place of the 1986 Elektor ComputerScope.

(140034)

---

**Retronics 2004**

Retronics is a monthly section covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com
ORDERING INFORMATION
To order, contact customer service for your region:

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

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

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

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

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

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

TERMS OF BUSINESS
Shipping Note:
All orders will be shipped from Europe. Please allow 2–4 weeks for delivery.

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

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

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

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

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

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

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

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

Go to www.elektor.com/member.
Hexadoku The Original Elektorized Sudoku

Nothing to solder, nothing to program? Why—for this month’s hexadoku all you need is a pencil, some logic reasoning and you are active in electronics. Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of five Elektor book vouchers.

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

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

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

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

Prize winners
The solution of the April 2014 Hexadoku is: A0263.
The €50 / £40 / $70 book vouchers have been awarded to: Arno Habermann (Netherlands), Eric Chamouard (France), Pascual Alagon Luna (Spain), Tim Royall (United Kingdom), and Panagiotis S Krokidis (Greece).
Congratulations everyone!

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

I immediately got some criticism from someone in management about my approach. “Why not just make one large program?” “Because this way is easier.” I replied—several times to the same question. (I knew saying “I’m lazy.” wasn’t the best response.) And since I didn’t want to re-write things because someone objected, I coded the output program first. I printed a page of simulated output to pass around and get comments. The immediate response was, “Can you simulate a whole program?” “Sure!” I said. I took a real program that had a lot of repetitive code, manually created an input file and printed it out. As I recall, it was about fifty pages long with every line being a test step (or about 2500 steps). (Typical manual test procedures are about 10 to 25 steps.)

They took the output, bound it with a boiler-plate preface and presented it to the project director as a sample document. After examining it he realized that a manual test procedure based on the software was not all that useful and dropped the requirement. In just a few days, my lazy approach had eliminated the need for dozens of silly and useless documents. And, of course, I didn’t have to finish the reverse-interpreter either.

Repetitive Steps

Nowadays, I do a lot of product development jobs that usually include an embedded computer. My clients don’t really care what microprocessor (µP) I use as long as it meets their requirements. So, being lazy, I use the same µP for nearly all my projects.

Of course, I re-use subroutines. But instead of placing them in an “include” file, I put them right into the µP as part of the program. This makes every program look and feel the same. They have the same subroutine names, same variable names and a similar layout. They’re easy to program, debug, document, compile, test and follow. Modern µPs have loads of memory that’s just wasted. So why not use it? I haven’t run out of memory space yet. And if that time comes, all I have to do is “comment out” unused routines to free up space. It’s a bit more complicated for hardware. It’s harder to create re-useable hardware modules that can be plugged into a Printed Circuit Board (PCB). However, since I use the same µP for most of my designs, I standardize the pin usage. And since most designs use common interface components (LCD display, LEDs, switches), the layout of these parts is very similar. I keep a basic PCB layout that I start with. It’s easy to take out those things I don’t need and then add in new parts. So each new PCB is really a modification of the same basic PCB.

I also keep a collection of useful circuits on hand. It saves me time and effort in the design stage. It’s much easier to duplicate or modify a known working circuit than it is to design, debug and test a new one. Remember that your time is valuable and that it is a limited resource. Being lazy is just a way to optimize your work.
The LTspice IV Simulator

LTspice IV is one of the most extensive SPICE simulators currently available, with over 1.5 million users worldwide. It is completely free, allowing you to work with an unlimited number of nodes, components and simulations. This book is the ultimate guide to LTspice IV. In a sturdy, hard-cover format, The LTspice IV Simulator describes the operation of the program, all available commands, the various editors, dealing with SPICE models, the use of non-linear components and more. The LTspice IV Simulator is more than just a manual. It also offers a variety of tips, methods and examples, all carefully illustrated using almost 500 drawings, diagrams and screenshots on high-quality paper. The book is designed so that it is suitable for both beginner and veteran SPICE users.

ISBN 978-3-89929-258-9
£42.95 • € 49.00 • US $67.00

The RPi in Control Applications

Raspberry Pi Hardware Projects

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

£34.95 • € 39.95 • US $54.00

Lots of power with low distortion

Q-Watt Audio Power Amplifier

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

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

Android Breakout Board

The FTDI FT311D is a flexible bridge that can interface your circuit to an Android smartphone or tablet. This Elektor Android Breakout Board offers options for seven digital outputs, four PWM outputs, asynchronous serial and I2C and SPI interfaces. The board is compatible with Android 3.1 (Honeycomb) or higher (Android Open Accessory Mode should be supported).

Ready-built module
Art.# 130516-91
£26.95 • € 29.95 • US $41.00

A Small Basic approach

PC Programming

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

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

All articles in Elektor Volume 2013

**DVD Elektor 2013**

This DVD-ROM contains all editorial articles published in Volume 2013 of the English, American, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy of PCB layouts at printer resolution, adapt PCB layouts using your favorite graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.

£23.95 • € 27.50 • US $38.00

**The First Elektor Chip**

### E-Lock

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

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

An essential source of reference material

**Process Measurements with C# Applications**

Measurement is vital to the successful control of any process. This book introduces PC based measurement systems and software tools for those needing to understand the underlying principles or apply such techniques. Throughout the book, the C# programming language is used to give the reader immediate practical desktop involvement. C-Sharp has a wide support base and is a popular choice for engineering solutions. The basics of measurement and data capture systems are presented, followed by examples of software post-processing. Application examples are provided from a range of process industries, with reference to remote monitoring, distributed systems and current industrial practices.

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

**Circuits & Projects Guide**

### Arduino

The Arduino user is supported by an array of software libraries. In many cases, detailed descriptions are missing, and poorly described projects tend to confuse rather than elucidate. This book represents a different approach. All projects are presented in a systematical manner, guiding into various theme areas. In the coverage of must-know theory great attention is given to practical directions users can absorb, including essential programming techniques like A/D conversion, timers and interrupts—all contained in the hands-on projects. In this way readers of the book create running lights, a wakeup light, fully functional voltmeters, precision digital thermometers, clocks of many varieties, reaction speed meters, or mouse controlled robotic arms. While actively working on these projects the reader gets to truly comprehend and master the basics of the underlying controller technology.

£34.95 • € 39.95 • US $54.00
Drag 'n Drop

**ADAU1701 Universal Audio DSP Board**

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

**Semi-kit:**
- all TH components, PCB with DSP preassembled

Art.# 130232-71
£63.95 • € 72.95 • US $99.00

**110 Elektor Editions, Over 2500 Articles**

**DVD Elektor 2000 through 2009**

This DVD-ROM contains all circuits and projects published in Elektor magazine’s year volumes 2000 through 2009. The 2500+ articles are ordered chronologically by release date (month/year), and arranged in alphabetical order. A global index allows you to search specific content across the whole DVD. Every article is printable using a simple print function. This DVD is packed with ideas, circuits and projects that are ideal for any electronics enthusiast, student or professional, regardless of whether they are at home or elsewhere.

£77.95 • € 89.00 • US $121.00

**Explore the RPi in 45 Electronics Projects**

**Raspberry Pi**

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

£34.95 • € 39.95 • US $56.40

**Helped By Arduino**

**Mastering Microcontrollers**

The aim of this book is not only to let you enter the World of Arduino, but also to help you emerge victorious and continue your microcontroller programming learning experience by yourself. In this book theory is put into practice on an Arduino board using the Arduino programming environment. Having completed this fun and playful course, you will be able to program any microcontroller, tackling and mastering I/O, memory, interrupts, communication (serial, I2C, SPI, 1-wire, SMBus), A/D converter, and more. This book will be your first book about microcontrollers with a happy ending!

348 pages • ISBN 978-1-907920-23-3
£34.95 • € 39.95 • US $54.00

**MIFARE and Contactless Cards in Application**

**RFID**

MIFARE is the most widely used RFID technology, and this book provides a practical and comprehensive introduction to it. Among other things, the initial chapters cover physical fundamentals, relevant standards, RFID antenna design, security considerations and cryptography. The
complete design of a reader’s hardware and software is described in detail. The reader’s firmware and the associated PC software support programming using any .NET language. The specially developed PC program, “Smart Card Magic.NET”, is a simple development environment that supports sending commands to a card at the click of a mouse, as well as the ability to create C# scripts. Alternatively, one may follow all of the examples using Visual Studio 2010 Express Edition. Finally, the major smart card reader API standards are introduced. The focus is on programming contactless smartcards using standard PC/SC readers using C/C++, Java and C#.

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

Ideal reading for students and engineers Practical Digital Signal Processing using Microcontrollers
This book on Digital Signal Processing (DSP) reflects the growing importance of discrete time signals and their use in everyday microcontroller based systems. The author presents the basic theory of DSP with minimum mathematical treatment and teaches the reader how to design and implement DSP algorithms using popular PIC microcontrollers. The author’s approach is practical and the book is backed with many worked examples and tested and working microcontroller programs. The book should be ideal reading for students at all levels and for the practicing engineers who may want to design and develop intelligent DSP based systems. Undergraduate students should find the theory and the practical projects invaluable during their final year projects. Similarly, postgraduate students should be able to develop advanced DSP based projects with the aid of the book.

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

Time & Date Floating in the Air
UltiProp Clock
Electronics is never so fine as when it skillfully combines magic with physics, mechanics with software, imagination with thoroughness and precision, and a taste for beauty with good workmanship. This timepiece was designed to display the time and date in an original way—but we admit we did also design it to draw cries of amazement from the visitors who find it in our laboratory. We’ll bet many of you will want to do the same in your own homes!

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

Ideal reading for students and engineers Practical Digital Signal Processing using Microcontrollers

Further Information and Ordering: www.elektor.com/store or contact customer service for your region
Elektor Project Generator Edition 2014 — The annual extra-thick edition with a bunch of circuits

Next month we publish the famed double issue for the months of July and August. Coveted, highly valued and collected by readers, Elektor’s Summer PGE is dreaded by the competition who never managed to publish anything remotely similar. The edition contains a mix of large and small projects. Editors and designers are now burning their midnight oil to fill the 2014 Summer PGE with detailed high quality texts, illustrations, schematics, boards, programs and original electronic applications. Don’t miss it, this Jumbo edition of Elektor packed with a ton of Elektoized electronic engineering.

Some large projects in PGE 2014:

Switch-mode Lab Power Supply
An ingenious circuit incorporating a custom-designed flat transformer with PCB track windings. The PSU has two LED displays for the current and voltage readout. The output is interruptible using a relay, and capable of sourcing up to 30 V, 1 A.

Ultrasonic Distance Meter
A handy little meter built around the ATtiny4313 and using two standard ultrasonic sensors. A 2-line display indicates the measured distance in mm, cm, inches, feet or yards selected by the user.

IO-Warrior Extension Board
This multifunctional board offers vast number of inputs and outputs make it highly suited to a variety of measurement and control applications via the PC. The communication with the PC is effectively via an IO Warrior module, for which a demo program is available written in Visual Basic Express.

See what’s brewing @ Elektor Labs 24/7

Check out www.elektor-labs.com and join, share, participate!
7 functions of PicoScope:

8 CHANNEL PC OSCILLOSCOPE

For just £1395
• High resolution
• USB powered
• Deep memory

Includes automatic measurements, spectrum analyzer, SDK, advanced triggers,
color persistence, serial decoding (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), masks, math channels,
all as standard, with free updates

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

www.picotech.com/PS243
PROTEUS DESIGN SUITE VERSION 8

Featuring a brand new application framework, common parts database, live netlist and 3D visualisation, a built in debugging environment and a WYSIWYG Bill of Materials module, Proteus 8 is our most integrated and easy to use design system ever. Other features include:

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

Version 8.1 has now been released with a host of additional exciting new features.
For more information visit.
www.labcenter.com