USB Hub with Legacy Serial Ports

2x: RS-232
2x: RS-422/RS-485
3x: USB
in one box

USB 1 ↩
USB 2 ↩
USB 3 ↩
USB ↩
RS422 2 ↩
RS422 1 ↩
RS232 2 ↩
RS232 1 ↩

! Zero microcontrollers
Zero system software

Make it to:
Make it at:
Made in:
Munich
electronica 2014
inside tomorrow

USB Hub feat. RS-232/-422/-485 │ **Varilab 402** │ **Microcontroller BootCamp (7, final)**

DesignSpark Tips & Tricks │ **Precise Nixie Clock** │ C Modules │ **Weird Components: Neon Bulbs**

Minuscule Prototyping & Stonehenging │ Post on Elektor.Labs & get a job

Econais WiSmart IoT Development Kit │ **Make It in Munich** │ SMD Feedthrough Capacitors

Add 3D Sensing to your Micro or PC │ Heathkit AA-100 Tube Amplifier (1960)

Hexadoku │ **Gerard's Columns**

ARM® Software Development Tools

The ARM software development environments provide you with all the necessary tools for every stage of your development workflow for ARM-based systems, from microcontrollers to application processors.

Keil® MDK-ARM™ Version 5

Keil MDK is the comprehensive software development environment for ARM Cortex-M processor based microcontrollers. Keil MDK contains the well-known µVision IDE and Debugger, ARM Compilation Tools and the new Pack Installer that installs and updates Software Packs for device support, board support, CMSIS and middleware.

Keil ULINK™ Debug Adapters

The Keil ULINK USB debug adapters provide a quick and painless connection to your target device. The ULINKpro top-end model includes unique streaming trace technology that allows comprehensive performance analysis and complete code coverage.

ARM DS-5 Development Studio

DS-5 is an end-to-end tools solution for embedded software development, especially crafted by ARM to enable you to develop robust, highly optimized products based on ARM processors. Whether you are a system-on-chip (SoC) designer, a real-time firmware engineer or an Android app developer, DS-5 helps you to take full advantage of the leading edge technologies in the ARM architecture.

ARM®KEIL®
Microcontroller Tools

MDK-ARM® Version 5
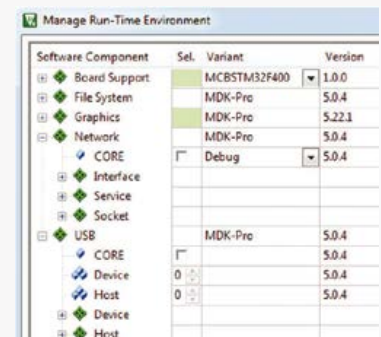
MDK supports the complete ARM® Cortex®-M series and introduces component-based software development.

MDK Core
µVision IDE with Editor
ARM C/C++ Compiler
µVision Debugger with Trace

MDK Professional Middleware

TCP/IP Networking	File System
USB Host Stack	Graphical User Interface
USB Device Stack	RTX RTOS
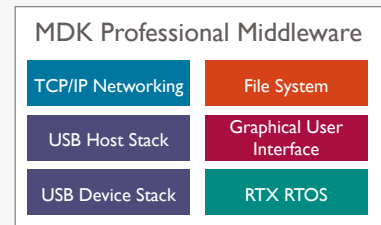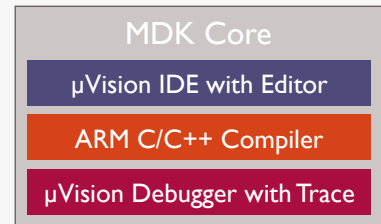
MDK Professional includes middleware for communication and standardized drivers for peripherals. These software components are managed in the Run-Time Environment.

Software Component		Sel.	Variant		Version
⊞ ◆ Board Support			MCBSTM32F400	▼	1.0.0
⊞ ◆ File System			MDK-Pro		5.0.4
⊞ ◆ Graphics			MDK-Pro		5.22.1
⊟ ◆ Network			MDK-Pro		5.0.4
◆ CORE		☐	Debug	▼	5.0.4
⊞ ◆ Interface					
⊞ ◆ Service					
⊞ ◆ Socket					
⊟ ◆ USB			MDK-Pro		5.0.4
◆ CORE		☐			5.0.4
⚙ Device		0			5.0.4
⚙ Host		0			5.0.4
⊞ ◆ Device					
⊞ ◆ Host					

keil.com/mdk5
+49 89 456040-20

reichelt electronics

Your competent online partner for

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

red pitaya

Convenient ordering!

Simply scan QR code and all items you need to start using Red Pitaya are in the shopping basket!

http://rch.lt/RPC

Micro SDHC Card, 8 GB
for the Red Pitaya SAMSUNG
- Read: 24 MB/s
- Specification: Class 6

SAMS MB-MS08DA (~ £ 7.25) **9,10**

Modular Probe:
for the Red Pitaya TESTEC
- Pitch: x 1/ x 10
- Bandwidth: 15/150 MHz
- Rise time: 24/2.3 ns

TESTEC LF 312 (~ £ 13.51) **16,95**

Micro USB power adapter
for the Red Pitaya
- Input voltage: 100 – 240 V, 50 – 60 Hz
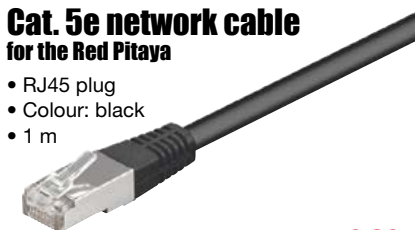- Output: 5V DC, 1.2 A
- 1,40 m

goobay

NT MICRO USB 1,2 (~ £ 3.15) **3,95**

!Possibly you'll need this adapter to use this power supply in UK.
TRA EUR GB (~ £ 2.19) **2,75**

Cat. 5e network cable
for the Red Pitaya
- RJ45 plug
- Colour: black
- 1 m

PATCHKABEL 1 SW (~ £ 0.64) **0,80**

HF adapter
for the Red Pitaya
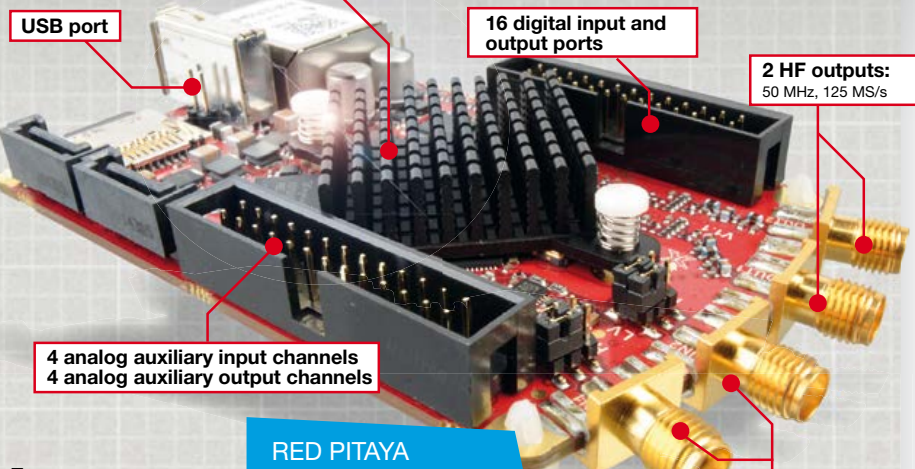- SMA plug to BNC socket

K 512 (~ £ 0.92) **1,15**

RED Pitaya
Open source measurement lab, 50 MHz

CPU/FPGA: Xilinx Zynq 7010 SoC, Dual ARM® Cortex™-A9

USB port

16 digital input and output ports

2 HF outputs: 50 MHz, 125 MS/s

4 analog auxiliary input channels
4 analog auxiliary output channels

2 HF inputs: 50 MHz, 125 MS/s

RED PITAYA
446,25 (~ £ 355.84)

Features:
- Channels: 2
- Bandwidth: 50 MHz
- Sampling rate: 125 MS/s
- Horizontal time base: 4 ns
- Vertical resolution: 14 bit
- DDR3 RAM 512 MB
- Power supply: micro-USB
- Dimensions: 107 x 21 x 60 mm

The Red Pitaya turns your smartphone, tablet or PC into a versatile measuring instrument!

The Red Pitaya:

The Red Pitaya is an open source measuring device in credit cardformat that you can configure based on your individual requirements. It allows you to replace numerous high-priced measuring devices such as oscilloscopes, function generators, spectrum analysers, etc. with just one device.

The device is integrated in a network based on a GNU/Linux operating system and can be individually programmed with numerous software interfaces. The user interface for all measurement processes is implemented through a web service by means of HTML pages. This allows control of the measurement platform from smartphones, tablet PCs or computers.

Order now! www.reichelt.com

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

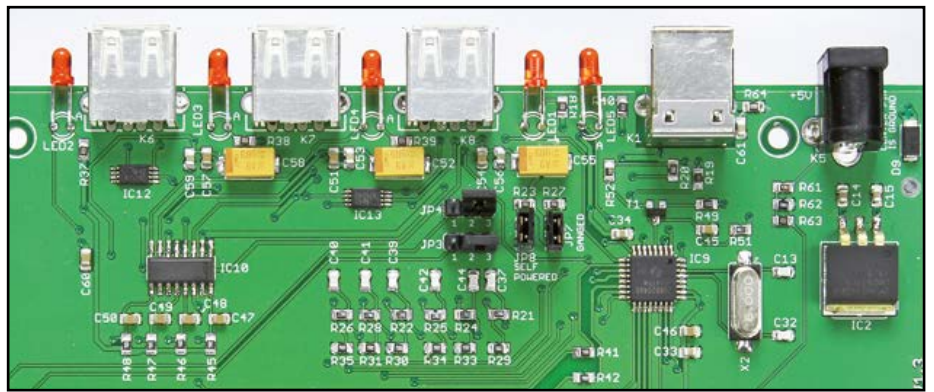Shop languages: Daily prices! Prices as of: 05/09/2014 International payment method: PayPal

Contents

● EIM Promotion

● Projects

● Review

● Labs

● DesignSpark

● Industry

● Regulars

•Community

Make It to Munich—Make It in Munich?

At the time of writing the 2014 *electronica* show in Munich is about five weeks up on my stack. Here at Elektor many different departments are contributing to Elektor Labs' presence at the show, called *Maker Space*. Editorial, Labs, Sales, Online, Logistics all put in their best effort. In the buildup to the show we secured backup from various companies and institutions also, each with their own preferred method of helping us out: hard sponsoring (solder irons, tools, parts, oscilloscopes!), soft sponsoring (demos and workshops), and in some cases plain advertising (see pages 44-49). All contributions have been marked with the MADE IN MUNICH stamp you'll find at several places in this issue.

If you can make it to Munich on November 11-14, you should be able to make it to Elektor's *Maker Space* in Hall A6, booth 380. Dropping by and participating in some of the activities at *Maker Space* hopefully gives a feeling of 'having made it' in Munich when you return home. Be warned though, Elektor Labs staffers are real engineers like you, they do not wear suits and take "make it" literally—to them it means soldering, measuring, debugging and problem solving. Their chats are tech ultimately, not sales pitch.

The current edition showcases at least two examples of projects classified in highly popular fields of electronics. First, there's Connecting-The-Old-With-The-New in the shape of our USB Hub featuring RS-232/-422/-485 which provides a single box on your desk to connect both USB devices and legacy serial equipment like that old EPROM programmer or A3 plotter you are so fond of. Second, in the Power Supplies division, we have a meticulously designed specimen called VariLab 402. Considering that both projects have made it all the way to Elektor magazine, for sure they will make it to Munich too.

Happy Reading,

come talk to me @ *electronica* 2014

Jan Buiting

Editor-in-Chief – Elektor International Media

The Team

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

Our Network

elektor

VOICE COIL

circuit cellar

hifi.nl
MUZIEK - FILM - LIFESTYLE

audioxpress

techthe future

Connects You To

282735 members in 83 countries

Supporting Companies

Not a supporting company yet?

Contact Johan Dijk (johan.dijk@eimworld.com, Phone +31 615 894 245,
to reserve your own space in Elektor Magazine, Elektor•POST or Elektor.com

WIZnet Connect the Magic 2014 Design Challenge:
Winners Announcement

WIZnet's Connect the Magic 2014 Design Challenge provided electronics enthusiasts with the opportunity to use WIZnet's WIZ550io Ethernet module in a project for a chance to win a share of $15,000 in prizes. The submission deadline was August 3, 2014, and soon thereafter the judges began scoring the entries. Eligible entries were judged on their technical merit, originality, usefulness, cost-effectiveness, and design optimization. We're excited to announce that the results are now in.

You can study the complete projects (documentation, schematics, photos, code, and more) at http://circuitcellar.com/wiznet2014/. Thanks everyone for participating and congratulations to the following winners!

First Prize
Chimaera: The Poly-Magneto-Phonic Theremin
Hans Peter Portner (Switzerland)

The Chimaera is a touch-less, expressive, network-ready, polyphonic music controller released as open source hardware. It is a mixed analog/digital offspring of the Theremin. An array of analog, linear Hall effect sensors make up a continuous two-dimensional interaction space. The sensors are excited with Neodymium magnets worn on fingers. The device continuously tracks and interpolates position and vicinity of multiple present magnets along the sensor array to

produce corresponding low-latency event signals. Those are encoded as Open Sound Control bundles and transmitted via UDP/TCP to a software synthesizer. The DSP unit is a mixed-signal board and handles sensor readout, event detection and host communication. It is based on an ARM Cortex M4 microcontroller in combination with WIZnet W5500 chip, which takes care of all low-level networking protocols via UDP/TCP.

Second Prize
LCDTV Server:
Streaming Media Using Ethernet/USB Adapter
Lindsay Meek
(Australia)

The WIZnet WIZ550io-based LCDTV Server project enables an LCD TV equipped with a USB port to stream media across a LAN. The small adaptor converts the Mass Storage Device requests coming from the USB into LAN Media requests using a virtual file system. When combined with a Power-Line to Ethernet Bridge, the user can watch digital video on an older TV from anywhere in the neighborhood.

Third Prize
WIZ Security Network
Claudiu Chiculita (Romania)

The project is a security system composed of multiple nodes. The nodes can collect and process information independently, and can generate alarms and communicate with the others in order to examine the threats from multiple angles. Each node has a WIZnet W5500 network chip, passive Power over Ethernet, PIR, servo motor, storage, video camera, and image-processing capabilities. A PC application can provide monitoring and configuration functions.

Honorable Mention
Sentry
David Penrose (USA)

The Sentry project uses an array of passive IR sensors placed in various rooms of a senior citizen residence to track motion about the residence. The system also uses motion sensors attached to chairs/beds to help decide if the resident is occupying the chair/bed. The systems are connected over an RF link to a processor with the attached WIZ550io to unobtrusively monitor the residents' activity and decide if normal activity is present. Should the processing decide that the movement patterns are outside of normal behavior an alert is sent to a remote family member or caregiver. The device allows a senior citizen to have privacy in their residence while still providing a degree of comfort that someone is available if they should need help.

Honorable Mention
Automatic Animal Feeder
Dean Boman (USA)

The Automatic Animal Feeder system is designed to automatically feed hay to small farm animals such as goats and sheep. The control system's network interface connection is provided by a WIZnet WIZ550io network module. The design extends the Internet of Things to the barnyard, which allows the operation of the feeder to be controlled and monitored remotely via the Internet.

Honorable Mention
WIZpix Connected Pixel controller
Robert Gasiorowski (USA)

The WIZpix pixel controller uses a WIZnet W5500 to connect to the Internet and an MCU to interface with W5500 and drive intelligent pixels. Main reason for this project is to create smart Christmas lights. However, CPC can be used anywhere animated lights are required (parties, displays, shows, home decor, etc.). It eliminates the need for expensive and complex DMX system. Thanks to built-in PoE, only one cable is required.

Honorable Mention
The Instrument of Things
Radko Bankras (The Netherlands)

The Instrument of Things (IoT) shows how to extend your custom electrical instruments with industry-standard capabilities for remote control via a TCP/IP interface. The WIZnet WIZ550io module is used to enable a basic web server, a portmap service, and a server for the remote control of the instrument using the VXI-11 communications protocol. The ultimate goal of The Instrument of Things is to easily add the VXI-11 communications protocol and LAN eXtensions for Instruments (LXI) technology to any electrical instrument project.

Honorable Mention
Radio Telescope Controller
Clayton Gumbrell (Australia)

This controller is for a radio telescope that's designed to observe the universe at the Hydrogen Line emission—a frequency of 1,420 MHz (21 cm wavelength). The radio telescope consists of a 1.7-m dish antenna mounted on a motorized azimuth/elevation mount, steerable in any direction above the horizon. A WIZnet WIZ550io Ethernet module provides the connectivity to the antenna controller, allowing the telescope to be located with a clear view of the sky and the operator and controlling computers to be located elsewhere, interconnected by the Internet.

(140309)

USB Hub *feat.*
RS-232/RS-422/RS-485

The e-engineer's delight: 3 x USB, 4 x legacy serial port, 0 x microcontroller, in one box!

By
**Sébastien
Guerreiro de Brito**
(France)

Since the DB25 / DB9 serial port has disappeared on modern PCs, it is getting complicated to connect up legacy equipment with RS-232 connectivity. Sure, there are worthwhile workarounds, but these tend to be limited by annoying draw-backs—in particular where multiple devices need to be connected. This was the driving force to design and present this universal converter with its two RS-232 ports and two RS-422 ports.

Yes, USB ⇆ RS-232 converters are very handy—but if you have to use several devices fitted with RS-232 connectors, you need as many convert-ers. And if in addition the interface is a true dif-ferential RS-422/RS-485 link, it's no longer as simple as all that to find a suitable converter. So I had the idea of developing a converter that can handle several RS-232 and RS-422/RS-485 links simultaneously. As so often, finding a case to house the project being designed is an important step, especially when there are lots of connec-tors to fit on. It's even the choice of case that to a large extent determines the dimensions of the PCB. After finding one that seemed just right and would allow me to comfortably accommo-date the various connectors, I realized that my original circuit would only take up 50 % of the space available. What could I do with the vacant space? Why, top off the project with a USB hub, of course!

Two stages
As I designed the circuit in two stages—first the converter, then the addition of the hub part—it's only logical to find it in the form of two dis-

tinct blocks on the block diagram (**Figure 1**). The detailed circuit appears in **Figure 2**; you'll probably find it helpful to refer back and forth between these two.

Let's start with the **circuit power supply**. This comes from two distinct sources—either from the host PC via the USB interface (K1), or from an external line PSU (K5) (5 V). Depending on the source selected, one jumper may need to be fitted. This is JP8—we'll come back to that later. When using an external PSU, the circuit is pro-tected against polarity reversal by Schottky diode D9. The LM2937ES-3.3 regulator (IC2) takes care of converting the 5 V to 3.3 V to bring the supply voltage down to the working voltage for the ICs. The hub part (**USB HUB** in Figure 1) of my inter-face is principally based around IC9, a TUSB-2046BVF from Texas Instruments. Along with the power switch IC10, it handles the power distri-bution automatically. From a single input port (K1), it can handle up to four USB output ports. Based on a state machine, this device has no on-chip software and doesn't need any config-uring to take care of controlling the supplies and
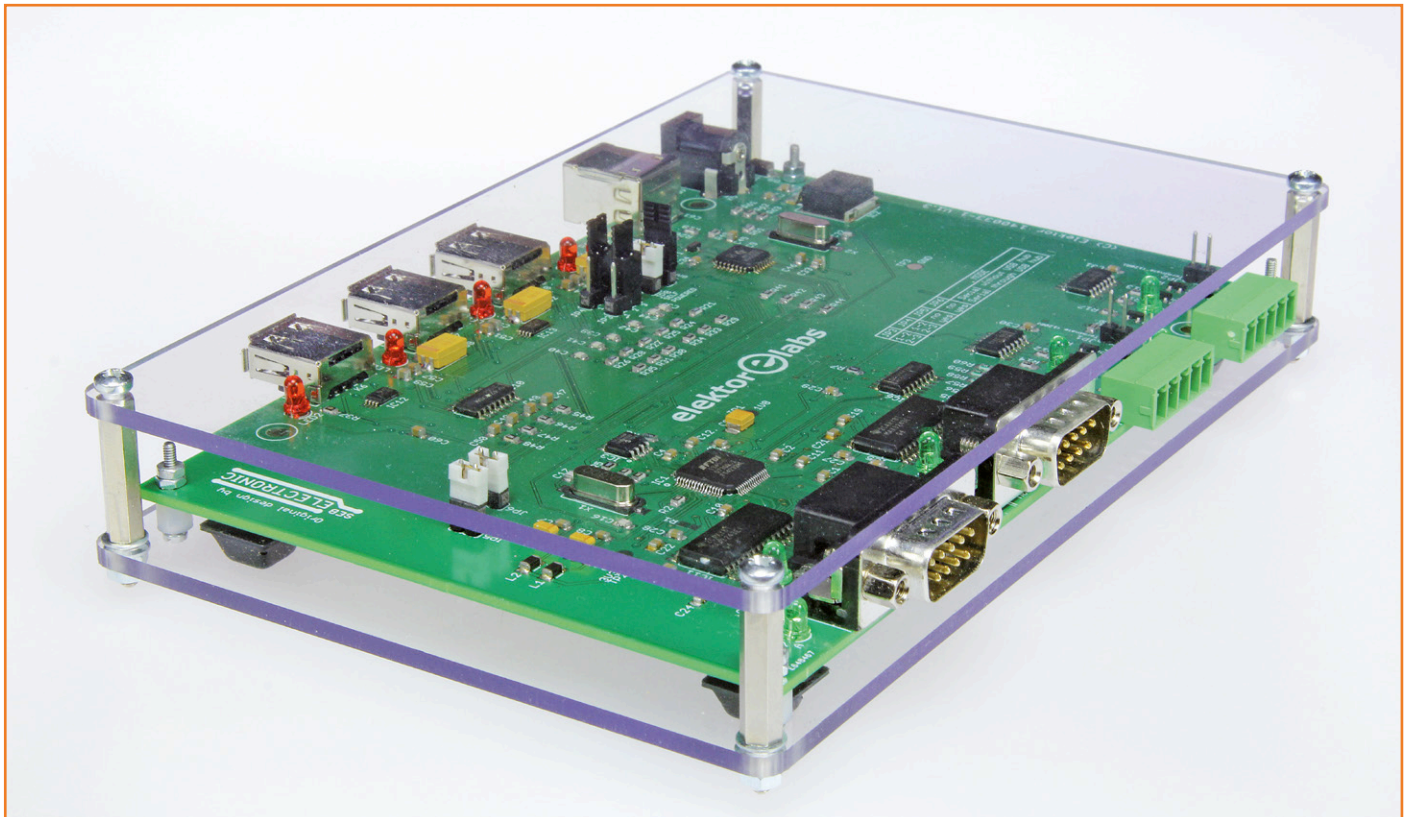
managing the USB ports. No need for a micro-controller here!

IC10 controls the **distribution of the supply current** to the USB devices (and to the UART part of this circuit). The TPS2044 can supply up to 500 mA per channel. Not only does it limit the current so it never exceeds this limit, but it also checks its own temperature via the internal transistor. Its default outputs ($\overline{OC}n$) are open-drain, which is why the pull-up resistors R45–48 are there to set the voltage. The RC networks on channels $\overline{OC}n$ make it possible to filter out overcurrent detection errors associated with the high current surges likely to occur during channel supply switching.

LEDs 1–5 indicate the activation of the USB peripherals: LED1 for the USB hub and LEDs 2–4 for the four elements on the hub.

All the inputs and outputs to this circuit may be subjected to transient interference. If these are of sufficient amplitude and duration, they may damage the USB hub or the USB-UART converter.

Figure 1.
Block diagram of converter/hub.
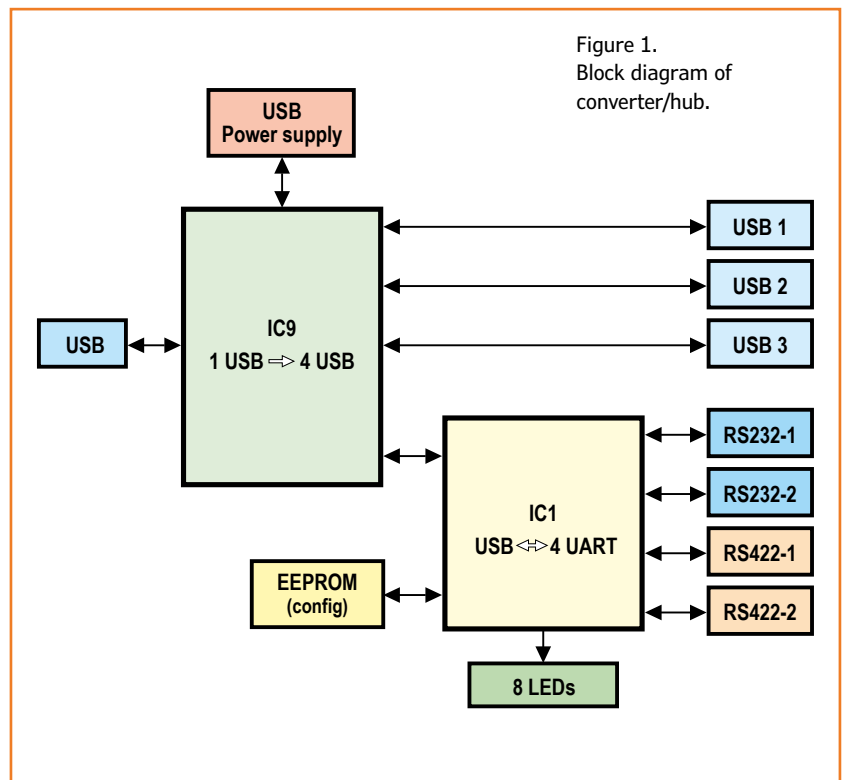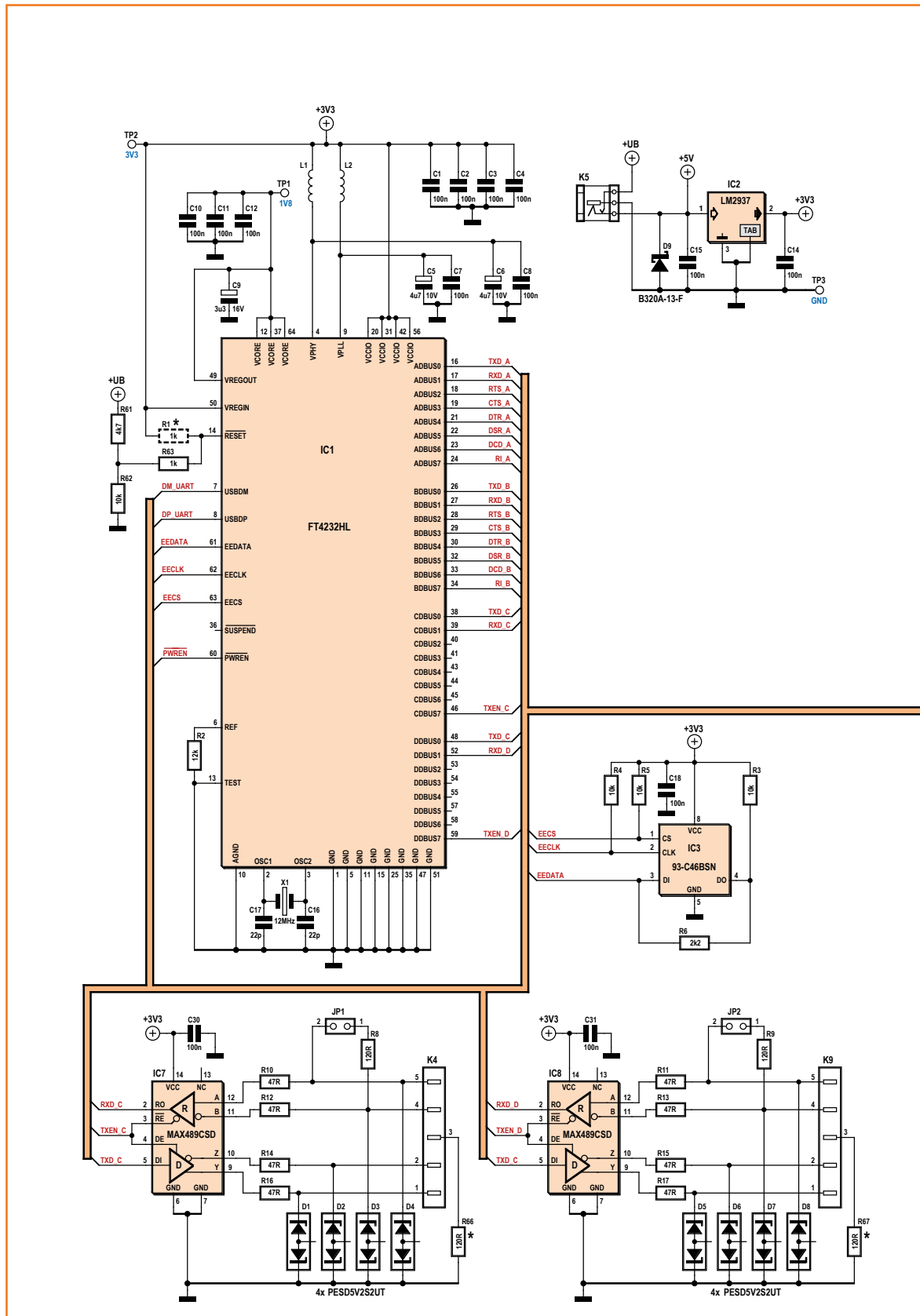
Figure 2.
Detailed circuit of the
converter/hub.
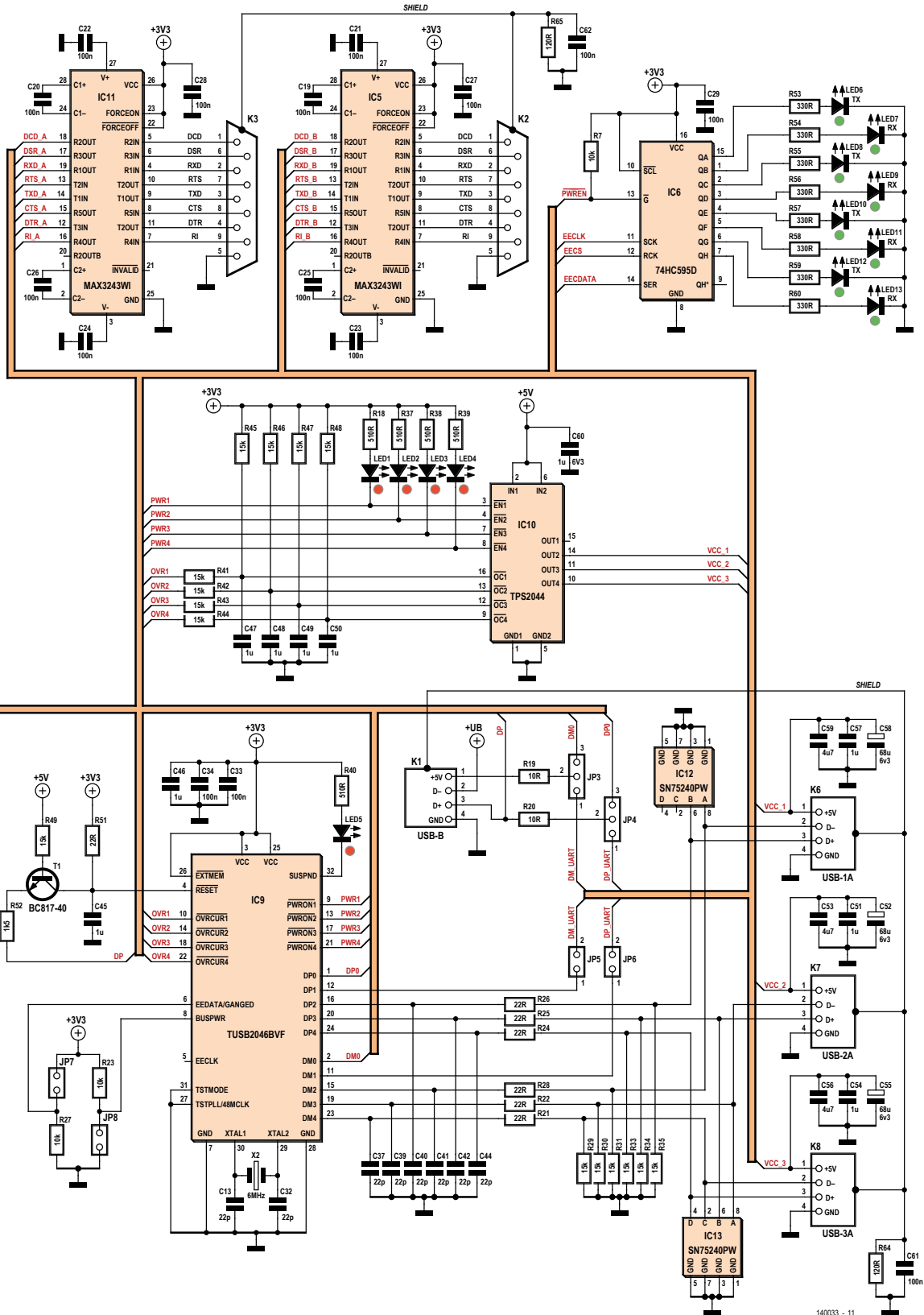
140033 - 11

To combat this interference, the circuit uses specialized SN75240PW protective devices from TI (IC12, IC13).

Attention: these interference suppressors introduce a capacitance that renders this converter unsuitable for USB 2.0 applications at high speed.

UART
The UART block in Figure 1 is IC1 in **Figure 2**, an FT4232HL from FTDI, which takes care of converting USB 2.0 (480 Mb/s) to four UARTs (Universal Asynchronous Receiver Transmitter). In our case, two UARTs are going to be used for the RS-232 links and the other two configured as RS-422/485 links (we'll come back to this when we talk about configuration).
The configuration selected is stored in an EEPROM (IC3).

In order to observe the input and output signals on the serial links, LEDs 6–13 are connected to IC6, an 8-bit serial in/parallel out shift register, interfaced directly with the converter IC1. To avoid the LEDs lighting spuriously, the $\overline{PWREN}$ signal issued by IC1 is used to disable the parallel outputs from IC6 while the USB converter was not recognized and configured by the PC driver.

To convert the TTL signals from IC1 into RS-232 signals, I use two MAX3243EIDW converters (IC11 and IC5). These handle the conversion of all the useful signals on the RS-232 link (full-duplex) (K2 and K3 at top right of Figure 2). This device is designed to also provide protection against air electrostatic discharges of ± 15 kV and ± 8 kV for contact electrostatic discharges.

The network formed by R65 and C62 connects the shield of the connector to the ground of our circuit, whilst providing EMC filtering (this function is also used for protection on the USB connector shield).

For the RS-422/RS-485 link (K4 and K9, bottom left) I'm using MAX489CSD+ converters (IC7 and IC8). Their outputs are protected against ESD discharges by D1–D8. It only remains to set the load resistance; jumpers JP1 and JP2 are used to load the receiver with the characteristic resistance of the cable: 120 Ω (see the box below about the terminating resistance).
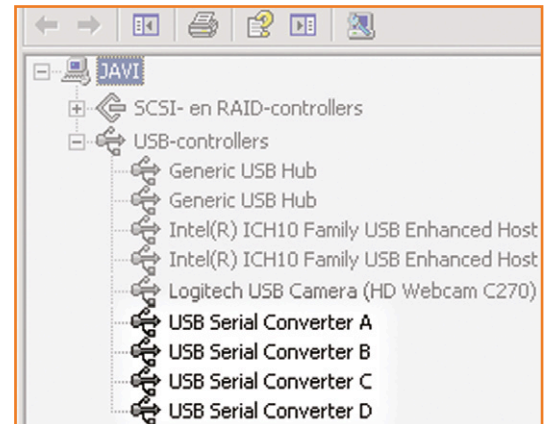
When the circuit is connected, the host system operating system will ask you to install the drivers [1]. Once these drivers have been installed, you will have four COM ports (virtual COM ports) on your machine. The two first are the RS-232 links and the other two the RS-422/RS-485 links.

The circuit has been tested under Windows and Linux. And it was immediately adopted by our lab guys, where everyone's very pleased to have it in service, as there's always a need for a legacy RS-232 or RS-485 interface somewhere or other.

Note: For the PC, a USB Serial Converter is not a serial port. To put the port into serial, the simplest thing is to open its Properties in Device Manager and, under the advanced settings, check the VCP or Virtual Com Port box. Repeat this operation for each USB Serial Converter. Then you'll need to unplug and reconnect the board to make Windows list the serial ports. This may take 5 minutes or more… afterwards, you'll find yourself with four serial ports (see **screenshot)**.
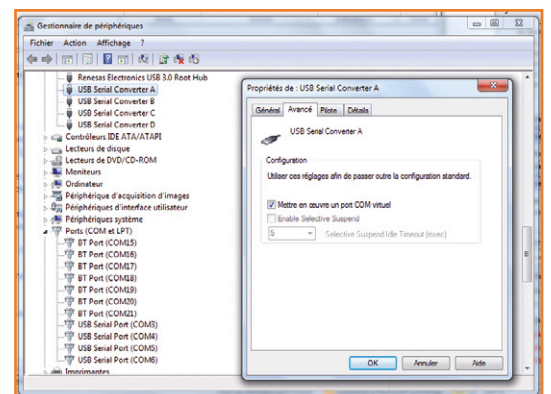
The board is available preassembled and tested from our e-store www.elektor.com/USB-serial-hub

Construction and mechanical assembly
With the PCB layout (**Figure 3**), the photo of the prototype, and the Component List, you shouldn't encounter any special difficulty building this device, as long as you have the appropriate tools for fitting surface-mount devices (SMDs) and good experience at it, especially for IC1, as well as IC9, IC12, and IC13, whose pins are very closely spaced. If you do set about it, don't overlook the small details like the polarity of the tantalum capacitors (C5, C6, C9, C52, C55 & C58). Take the trouble to check the data sheet!

Don't fit R1—it's an orphan from an earlier version of the circuit.

In order to align the green LEDs fitted in three stacked pairs, you'll need to take great care bending their leads so they are as accurately as possible the same.

Given the usefulness of this project to any electronics technician who still has excellent devices in perfect working order with the sole "defect" of only communicating via RS-232 or RS-422/RS-485, Elektor believes it is worth offering a ready-to-use assembled version (140033-91). So why don't you too take advantage of the convenience offered by the ElektorPCBservice [2]? The board you've either built or bought ready-to-use is fitted into a case, e.g. a model from Vero, for which cut-out drawings for the front and rear panels are available [2] in DXF or FPD formats (the latter is a native format from Schaeffer [3], of whom I am one satisfied customer).
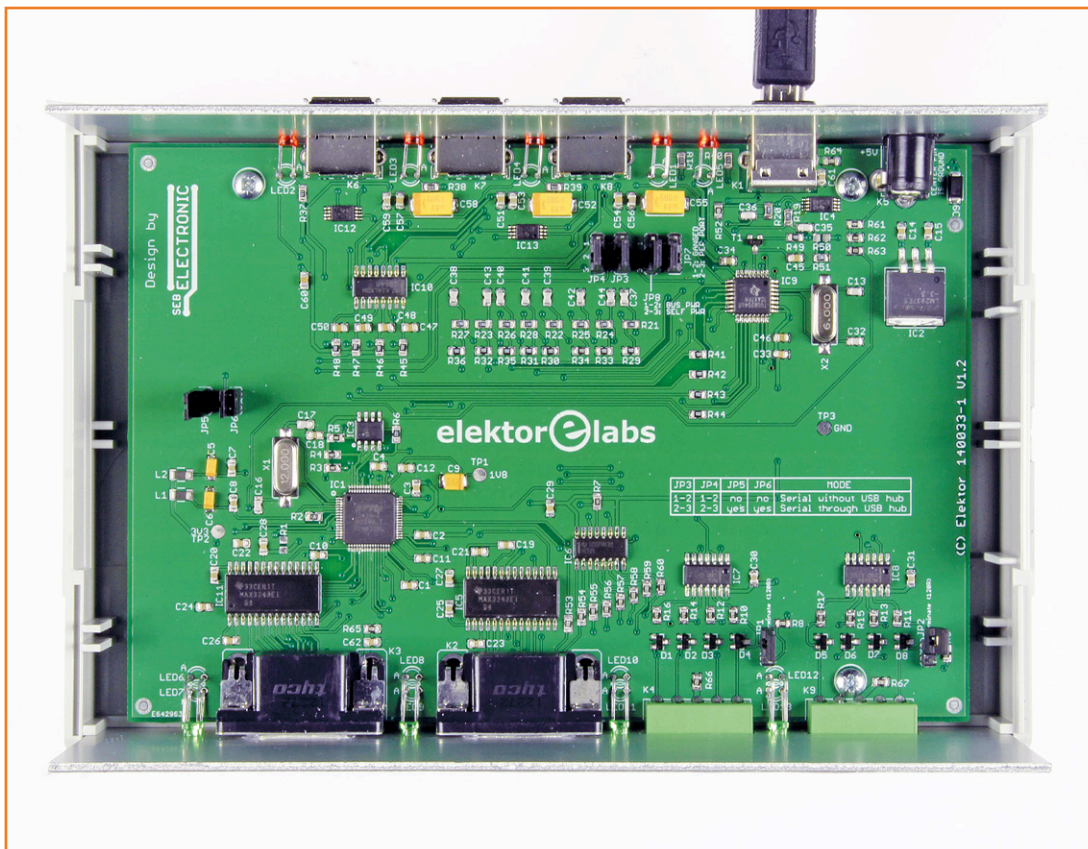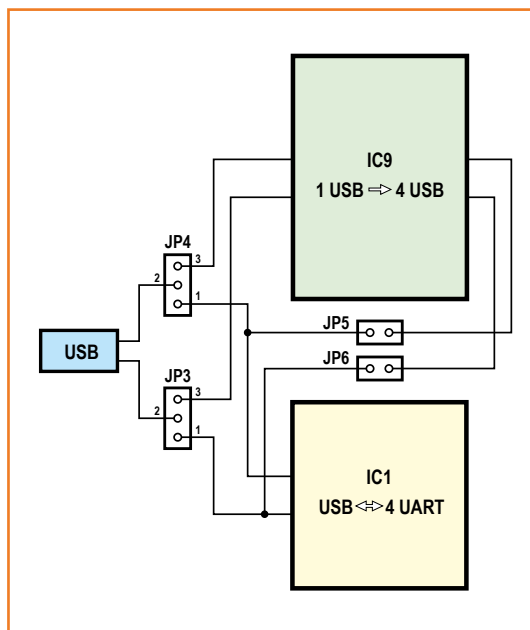
Figure 4.
Configuration of jumpers
JP3, 4, 5, and 6.

The power connector K5 (note in passing that it's a type with built-in switch) wasn't there in the original version of my cut-out drawings, I added it at the last minute for this published version, which has only been verified by simulation: the hole corresponds perfectly, but I'm not making any guarantees.

If you fit the PCB into the VERO case mentioned in the component list, you'll see there's no fixing hole adjacent to K3. To get a solid fixing, use the spacers on the DB9 connectors to fix them to the front panel.

Attention, the central pin of the power connector is the ground!

Configuration, settings, and drivers
For setting up the UART section (the USB converter has to be configured to operate with an

Component List

Resistors
(SMD 0805)
R19,R20 = 10Ω
R21,R22,R24,R25,R26,R28,R51 = 22Ω
R10,R11,R12,R13,R14,R15, R16,R17 = 47Ω
R8,R9,R64,R65,R66,R67 = 120Ω
R53,R54,R55,R56,R57,R58,R59,R60 = 330Ω
R18,R37,R38,R39,R40 = 510Ω
R23,R27 = 10kΩ
R1*,R63 = 1kΩ
R52 = 1.5kΩ
R6 = 2.2kΩ
R61 = 4.7kΩ
R3,R4,R5,R7,R62 = 10kΩ
R2 = 12kΩ 1%
R29,R30,R31,R33,R34,R35,R41,R42,R43,R44,R45,R46,
  R47,R48,R49 = 15kΩ
* do not fit

Capacitors
(Default: SMD 0805)
C13,C16,C17,C32,C37,C39,C40,C41,C42,C44 = 22pF
C1–C4,C7,C8,C10,C11,C12,C14,C15,C18–
  C31,C33,C34,C61,C62 = 100nF
C45–C51,C54,C57,C60 = 1µF
C9 = 3.3µF 16V tantalum (E)
C53,C56,C59 = 4.7µF
C5,C6 = 4.7 µF 10V, tantalum (A)
C52,C55,C58 = 68µF 6.3V, tantalum (D)

Inductors
L1,L2 = ferrite, 70Ω @ 100MHz, SMD 1206

Semiconductors
D1–D8 = PESD5V2S2UT
D9 = B320A-13-F
IC1 = FT4232HL

IC2 = LM2937ES-3.3
IC3 = 93LC46B-I/SN
IC12,IC13 = SN75240PWR
IC5,IC11 = MAX3243EIDW
IC6 = 74HC595D
IC7,IC8 = MAX489CSD
IC9 = TUSB2046BVF
IC10 = TPS2044BD
LED6,LED7,LED8,LED9,LED10,LED11,LED12,LED13 =
  LED, green, 3mm
LED1,LED2,LED3,LED4,LED5 = LED, red, 3mm
T1 = BC817-40

Miscellaneous
JP1,JP2,JP5,JP6,JP7,JP8 = 2-pin pinheader, 0.1" pitch
For JP1 and JP2: 120Ω termination resistor for RS-485
  (optional, see text box)
JP3,JP4 = 3-pin pinheader, 0.1" pitch
Jumpers, 0.1" pitch, for JP1-8
K1 = USB-B receptacle, right angled, PCB mount
K2, K3 = 9-pin sub-D plug
K4, K9 = 5-pin pinheader, PCB mount, 0.15" (3.81mm)
  pitch
K5 = barrel jack supply socket, 12V/3A with switch,
  0.075" (1.95mm) pin
K6, K7, K8 = USB-A receptacle, right angled, PCB
  mount
X1 = 12MHz quartz crystal, SMD
X2 = 6MHz quartz crystal, SMD
Case, ABS, 65x180x120 mm e.g. VERO 75-265742
PCB # 140033-1-v1.2
Ready assembled unit: # 140033-91 [2]

Note: the gaps in the components list (e.g. IC4, R50,
  etc.) are not oversights—simply components that
  have become redundant, but didn't justify renumber-
  ing everything.

Figure 3.
PCB for the converter/
hub. The red LEDs ought
to have been bent over like
the green LEDs so they can
be seen properly once the
circuit is fitted into a case.

RS-485 link, we'll come back to this later) **without** the USB hub section, there's a group of jumpers (**Figure 4**) to allow you to isolate the two blocks. To isolate the UART section from the USB hub section, all you have to do is:

- fit JP4 between 1 and 2;
- fit JP3 between 1 and 2;
- remove JP5 and JP6.

In normal operation, the jumpers must be positioned as follows:

-
- JP4 between 2 and 3;
- JP3 between 2 and 3;
- JP5 fitted;
- JP6 fitted.

Two other jumpers are used to configure the behavior of IC9.

JP	Function			
	Jumpers			
1	R_t 120 Ω*			
2	R_t 120 Ω*			
3	1-2 :	*serial without hub*	2-3 :	*serial through hub*
4	1-2 :		2-3 :	
5	0 :		1 :	
6	0 :		1 :	
7	1 : *all outputs switched*		0 : *single output switched*	
8	1 : *external power*		0 : *bus powered*	
	1 = present	0 = absent	* (option, see text)			

- JP7 determines the way the device reacts to information about overload on one of the ports: if the GANGED input (pin 6) is at 3.3 V (i.e. the jumper is fitted), **all the outputs** are shut down in the event that an
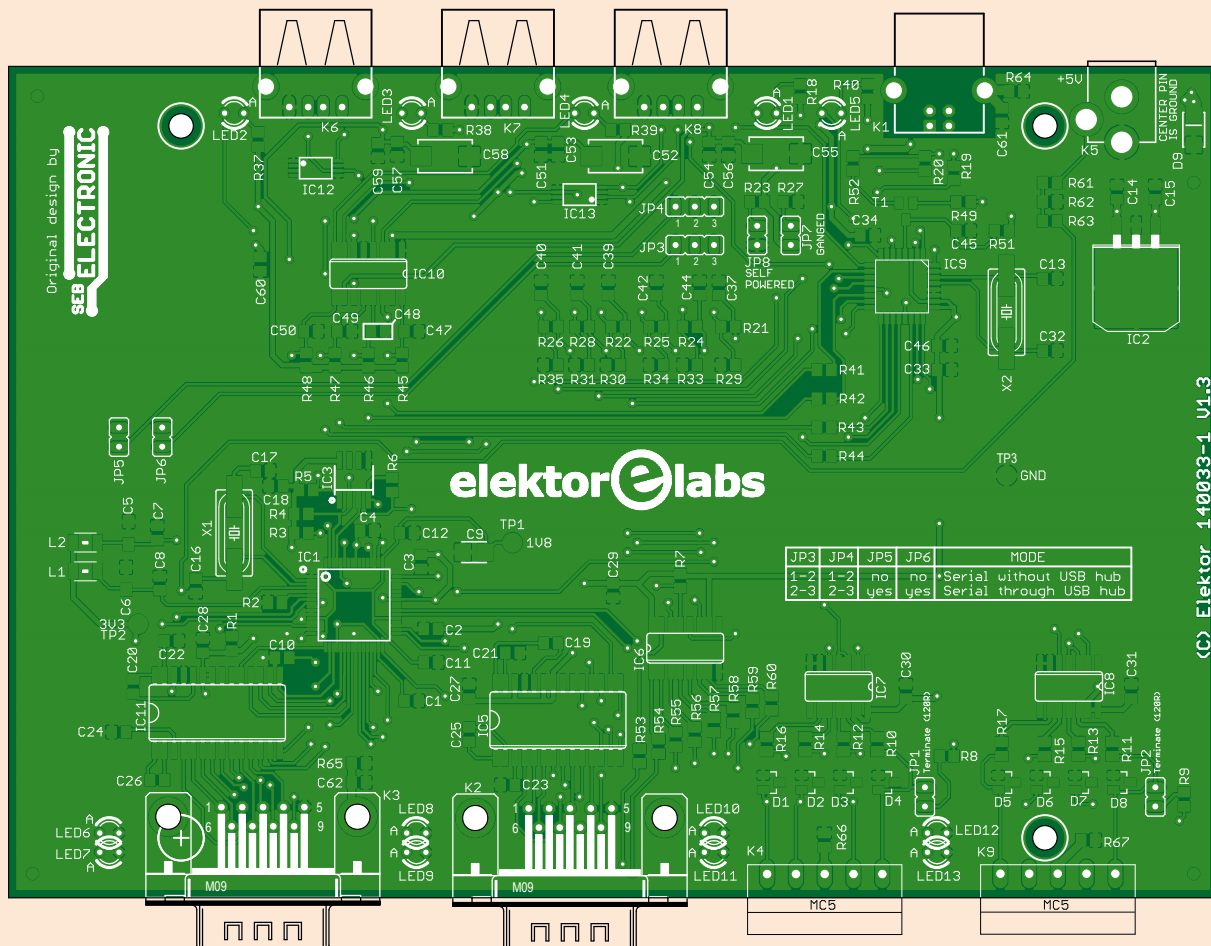
Figure 5.
The converter in its case with the front panel designed by the author.

overload is detected on one of the channels. If the GANGED input is at 0 (i.e. the jumper is not fitted), only the overloaded output will be shut down.

● JP8 tells the device what power source is being used. If JP8 is omitted, BUSPWR is at 3.3 V, **and** the power will come from the USB bus. If jumper JP8 is fitted, the power comes from an external supply.

IMPORTANT: Jumper JP8 must be fitted **the first time** the board is connected to a computer (with
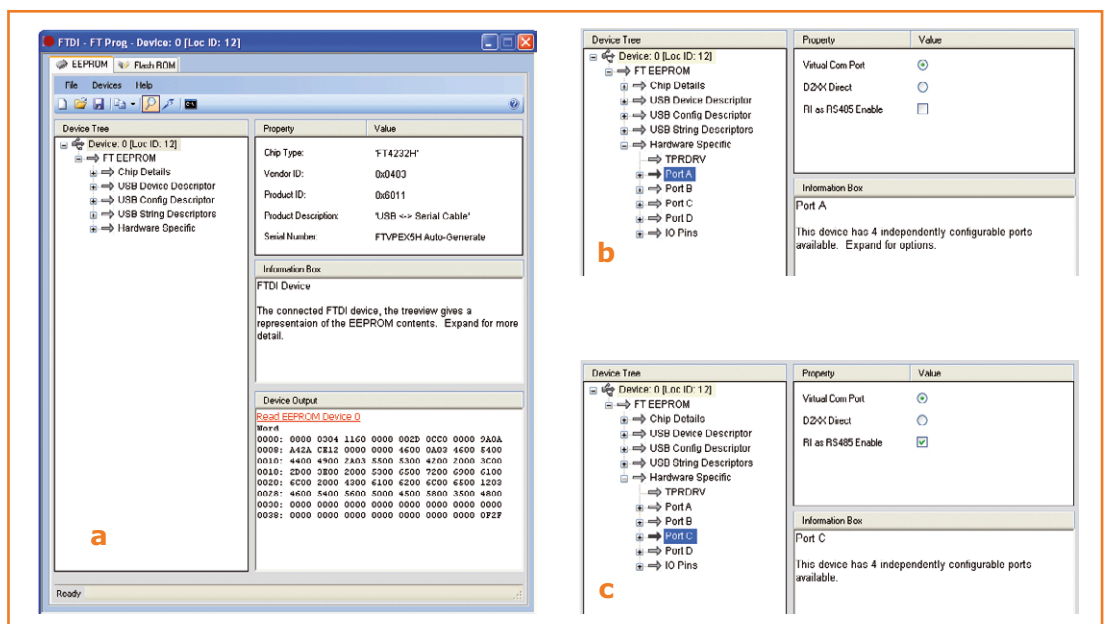
Figure 6.
Three screenshots for setting up and configuring using the FTPROG software from FTDI.

RS-422/RS-485 differential links, recapping

The RS-422 bus is defined in the EIA-422-B-1994 standard as a simplex bus, i.e. there is only a single transmitter at any one moment, which can drive up to 10 UL (Unit Loads). The signals are transmitted differentially in order to obtain highly robust data transmission over long distances, even in a noisy environment.

Baudrate vs. range
The transmitter is capable of sending data over a distance of approx. 1200 m (4000 feet), but the baudrate is reduced as the range increases. The standard specifies that the data rate may reach 10 Mbps, but only over short distances (**Figure 7**).

Terminating resistor
Because of the difference in impedance between the cable and the receiver, a phenomenon of reflection of the transmitted wave can occur on the line when transmitting over long distances or at high data rates. To overcome this phenomenon, a terminating resistor (on JP1 and JP2) is needed to match the load to the characteristic resistance of the cable (typically 120 Ω).
There are other termination modes, but as they are less common, I won't go into them here.

RS-485 link
The RS-485 bus (**Figure 8**) is based on the RS-422 bus, with certain additional functions like the number of ULs increased to a maximum of 32. The RS-485 bus is able to support multi-point configuration. This configuration makes it possible to have one transmitter communicating with several receivers.

This is the base configuration of this bus, using semi-duplex communication (two lines for the data). In semi-duplex, the receivers must be able to recognize that the sequence currently being transmitted is intended for them. This implies the use of an addressing system to allow the receiver to be identified. Commands can also be sent in 'broadcast' by sending a sequence to a virtual receiver with a reserved address (e.g. 255). In this way, all the receivers will react to the sequence received.
Otherwise, a command can be sent to a specific receiver quoting its address in the sequence, but there is one major drawback: receivers cannot take over the line to send information unless they have specifically received the order to do so (master/slave notion). Established practice dictates that in this type of communication, the master transmits and the slaves (receivers) listen. When the master has requested specific information from a slave, it and it alone responds. If the master has sent information in broadcast (to all the receivers), the receivers must not respond.
Contrary to popular belief, the RS-485 bus needs not two but three wires for communication: Data+, Data−, and Ground!

Unit Load
The standard lays down that for a given transmitter, a maximum of 32 transmitter/receivers can be connected. This is the notion of UL (Unit Load), connected to the receiver input resistance (defined by the standard as 12 kΩ). As it is now possible to find converters that only present ⅛ UL, it is thus (theoretically) possible to connect 8 × 32, i.e. 256 receivers on the same bus.
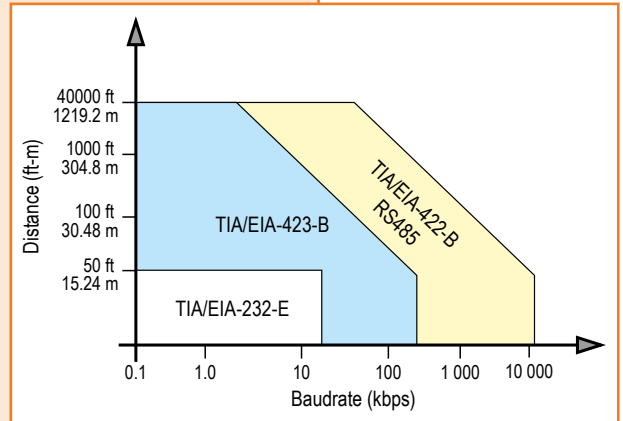
Figure 7.
Relationship between range and data rate for RS-422/RS-485 differential links.

Figure 8.
Typical RS-485 topology.
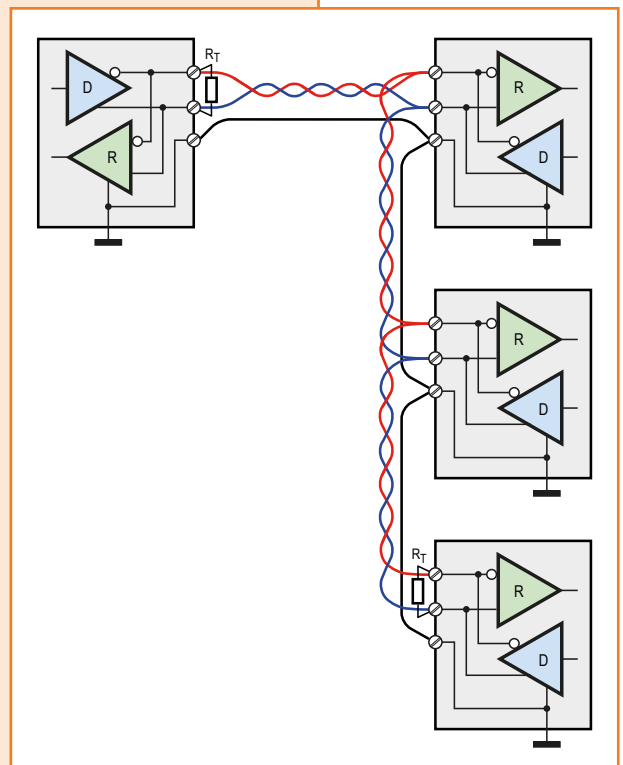
A passion

It was 25 years ago during the summer vacation! It was raining. To while away the time, I asked my mother to buy me a magazine that I didn't know— the Elektor Project Generator edition… Revelation! Ever since then, electronics has been part of my life. My working life as manager of EmkaElectronique technical consultants, as well as my private life, where I spend practically all my free time on it. Never a day goes by (and sometimes several nights) without my thinking about, designing, or building a board. I am only able to live out this passion to the full thanks to the good offices of my wife and daughters.

Thanks too to the enthusiastic Pascal for his sensible advice.
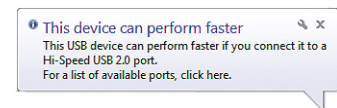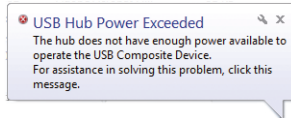
or without an external supply). This procedure avoids an overload warning from the computer's USB hub, which then prevents installation of the board's drivers (**screenshots**). Once these drivers have been installed, JP8 can be removed (or left in place, if you are using an external supply).

To configure the circuit with its two differential serial links, we need to use the FTPROG software available from the FTDI website [1]; all the USB drivers for the operating system are also available there. Once the driver(s) has/have been installed, first run the FTPROG software, then click on the magnifying-glass so that the converter is detected (**Figure 6a**).

The converter is then correctly detected by the software. Then click on "Hardware Specific". (**Figure 6b**)

For the **PORT A** and **PORT B** ports, nothing has to be touched. For the **PORT C** and **PORT D** ports, you'll need to validate the TX enable pin: still using the FTPROG tool, check the **RI as RS-485 Enable** box. (**Figure 6c**)

Lastly, it only remains to save the configuration into the EEPROM by clicking on the 'lightning' icon.

140033

FTDI & Android

Just before putting this article to bed, we carried out some trials with tablets. The ones where the USB interface operates in 'host' mode do recognize the USB hub, but we didn't get much further than managing to connect up a single USB stick. It would actually be useful to be able to communicate with the RS-232/RS-485 ports, but to do this the FT4232 would need to talk to Android, as we can see it doing here [6]; this link is useful for the documentation to which it links. Maybe this will be the subject of a future article, with your participation?

Web Links

[1] FTDI website for drivers and the FTPROG software: www.ftdichip.com

[2] www.elektor.com/USB-serial-hub, www.elektor-magazine.com/140033

[3] Panel drilling: www.schaeffer-ag.de

[4] AN-1031 "TIA/EIA-422-B Overview": www.ti.com/lit/an/snla044b/snla044b.pdf

[5] RS-422 and RS-485 Applications e-Book: www.bb-elec.com/Learning-Center/All-White-Papers/Serial/RS-422-and-RS-485-Applications-eBook.aspx

[6] FTDI and Android: www.youtube.com/watch?v=QSR7IAAWL1c

Econais 'WiSmart' IoT Development Kit

By **Jan Buiting,**
Editor-in-Chief

1

Many products launched during the present storm of innovation called IoT are black boxes unsuitable for R & D and product development from the ground up. If you want to make a difference, yet rely on trusted, free, software tools to get down to the chip level of things, a development kit can do a great job in more than one way.

Greece and Silicon Valley based company Econais are promoting their EC19D 'WiSmart' IC in a way that should appeal to advanced engineers as well as enthusiasts wishing to develop nifty products for the IoT market—products that are (1) smart, (2) ecofriendly and (3) Wi-Fi related. Econais' latest development kit type **EC19D01DK** is a promising approach to developing your very own IoT devices for small production runs as well as volume production. In doing so you will learn an awful lot.

Standard tools do a great job
Econais probably differs from much of the competition by not insisting on a .5 GB proprietary tools suite to be downloaded. Barring a little Python program called *SimpleCom.py* their strategy is for the entire process of developing, debugging and implementing a Wi-Fi'd IoT device to be based on names and products makers like you want to hear because they are familiar and/or free: Linux, Windows, SPI, UART, JTAG, AT command set, FTDI USB-TTL, SD card, Open Hardware, Gerber, Python.

After a simple login at the Econais website [1], you can grab a vast amount of documentation available on the EC19D chip and the associated EC19D01DK dev kit (launched late August 2014). Elektor was one of the first recipients of the new kit. The -DK supersedes an earlier version and at virtually the same price now includes a debugger board.

From TTL-TxD/RxD to Custom MAC
In the box are the expected cables, a multi-standard power supply, the EC19D00SD Wi-Fi board (**1**), the EC19D01EX expansion board (**2**), and the EC19D01DBG JTAG/Debugger board (**3**). Also

the obligatory get-you-going cards with information so terse as to appear 'challenging' initially. Although the docs are a bit rough round the edges the kit is well thought out in terms of education. The first hurdle is high though—getting the -EX board to talk to you, as I experienced. This can be done in two ways, depending on what you want to hear from the EC19D chip (through the -EX board): 'machine' talk or 'human' talk. The first is pure commands and numbers, the second, not unlike AT commands. Both methods employ an FTDI USB-TTL virtual COM port (RxD-TXD-CTS-RTS) and a terminal program like RealTerm (Windows) or Cuteterm (Linux) for the old skool approach, or the Python way with 'SimpleCom'. SimpleCom requires Python to be installed, which is good considering the RPi's potential for IoT. Once the comms are running your ARM, RPi, AVR or Arduino can take over the SPI/UART port on the -EX board and you're well on the way to WiFi'd IoT. All traffic on the port is human 'readable', i.e. consisting of commands, datawords and parameters.

In a really sophisticated setup, the -SD, -EX and -DBG board are interconnected as on the photograph and linked to the PC with a USB cable. This setup is currently only supported in Linux. The -DBG board doubles as an advanced flash programmer for the EC19D chip. Being a relatively new product it is in need of technical description and application examples.
If you are ambitious about IoT programming this kit is for you. Among the exciting things awaiting you are *OTA* (over the air) upgrading, Smart Metering, channel data capturing and logging with *Wireshark*, and waking any number of hidden (!), dormant (!) Wi-Fi devices with the *ProbMe* app using the Wi-Fi Direct mode on your Android smartphone.

Conclusion

With your effort put in, the EC19D01DK enables the complete IoT application development cycle to be realized from various angles: programming and soldering, Linux and Windows, USB and microcontroller I/O. Everything in the kit has a feeling of pioneering and exploring about it, which is underscored by a ton of Linux programming examples.

The people at Econais have worked with Elektor to make their latest product available at a

2
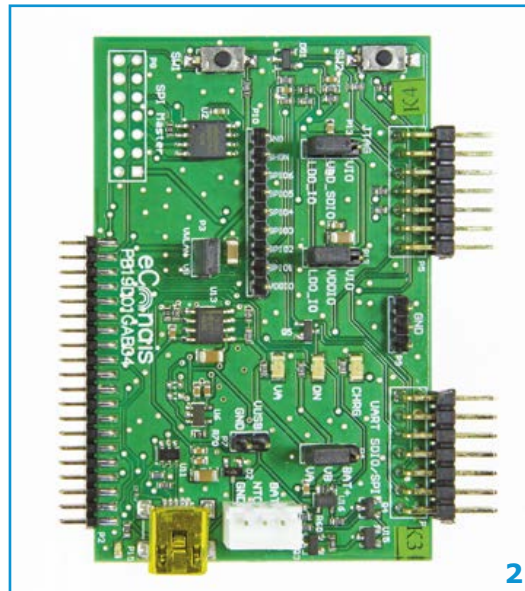
special reduced price. The EC19D01DK is now available exclusively from the Elektor Store [2]. I'd expect an Elektor reader to be the first to submit and publish a smashing IoT application with the EC19D.

(140338)

Web Link

[1] www.econais.com
(follow: Support -> Dev Kit Resources)

[2] www.elektor.com

3
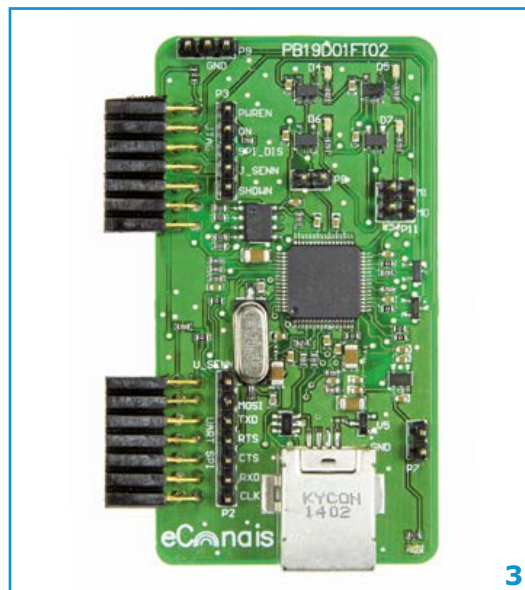
VariLab 402 (1)
A 40-V 2-A lab power supply with two-stage regulation
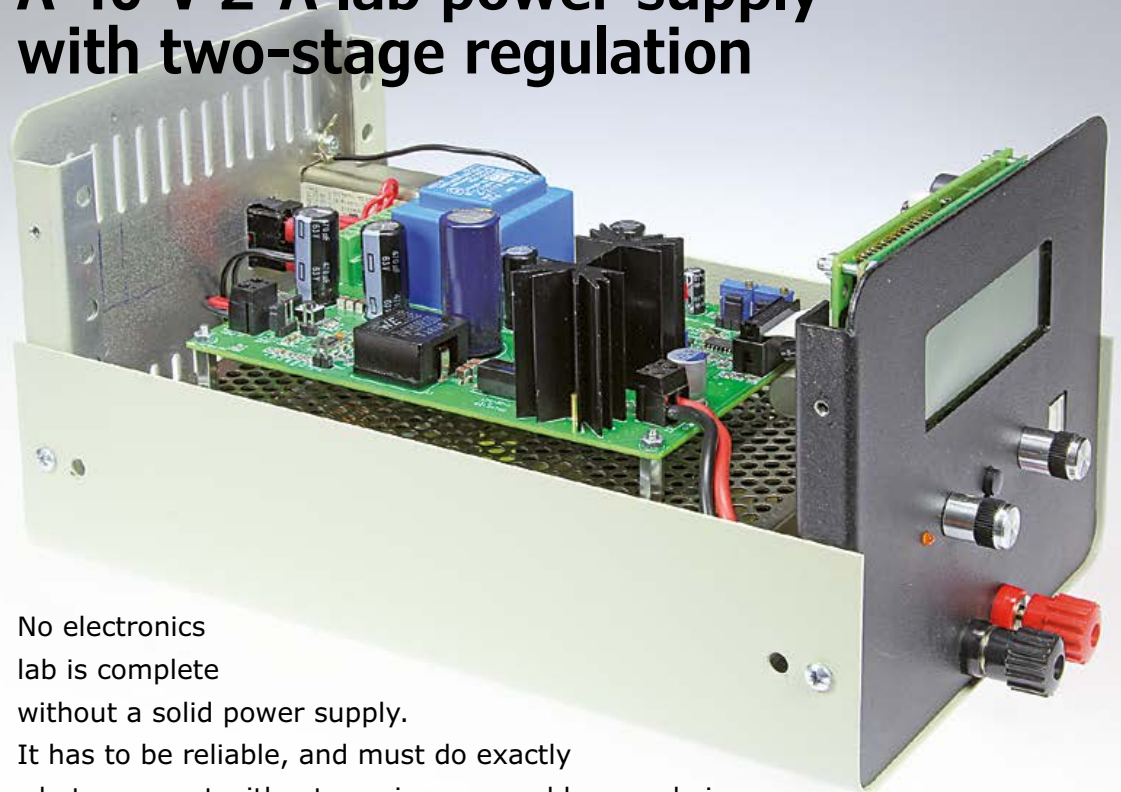
By **Ton Giesberts**
(Elektor Labs)

No electronics
lab is complete
without a solid power supply.
It has to be reliable, and must do exactly
what you want without causing any problems or being
upset by sudden load changes. The lab power supply described here features
two-stage regulation with switching and linear stages, along with a wide output
voltage range from 0 to 40 V.

Every electronics enthusiast or professional who occasionally designs something needs to have at least a few instruments, a soldering iron and a lab power supply on their bench. The power supply must be trustworthy and stable under all conditions, and it must be able to supply the most commonly used voltages and currents. A lot of low-cost lab power supplies are available nowadays, but you have to ask yourself what sort of quality and durability you get with these products. The price of a high-quality lab power supply can easily run into the three-figure or four-figure range.

Elektor has been around for more than 50 years, and during that time we have developed and published a large number of power supply designs. They are especially popular among the DIY com-munity. In the last few months we have already devoted attention to several lab power supply models, but the one described here has been developed completely in the Elektor Labs and offers a bit more than most other lab power supplies.

All (affordable) standard lab power supplies have a maximum output voltage of 30 V, which is not quite enough for some applications. That's why one of the requirements for this Elektor power supply project was a maximum output voltage of 40 V with output current up to 2 A. We also thought it would be nice to incorporate a microcontroller to manage the power supply and drive the display, and to enable communication with a PC.
With these requirements in mind, we started looking for a suitable design concept and suit-

able components for our envisaged power supply. Due to the large voltage range of 0 to 40 V, a standard linear regulator is not a good idea. It would have to dissipate about 80 W or more at the maximum output current. That requires a large heat sink or forced-air cooling to handle the maximum dissipation over an extended period. Although there are various tricks that could be considered to mitigate this situation, such as working with several fixed output voltages, there's no getting around this considerable power dissipation. That's why we chose a switched mode power supply (SMPS) architecture for our design. One of the key advantages of an SMPS is high efficiency. Unfortunately, there are also downsides to an SMPS, such as ripple on the output voltage and switching noise. To remedy this, we decided to connect a linear stage after the SMPS in order to reduce these imperfections to an acceptable level.

Features

- Output voltage range 0–40 V; output current range 0–2 A
- Stabilized output voltage with two-stage regulation
- Input voltage from standard 48 V power module
- Voltage and current can be adjusted as desired using on-board potentiometers or the microcontroller
- Microcontroller control with ATxmega128A4U-AU
- Two rotary encoders and pushbuttons for user operation
- Four-line display shows U_{set}, I_{set}, U_{out}, I_{out}, output power, crest factor and other parameters
- USB port for connection to a PC
- Pushbutton for output enable/disable
- Output ripple <15 mV$_{pp}$ at 40 V, 2 A
- Output regulation 220 mV$_{pp}$ with pulsed load (100 mA, 1 A load, 50 Hz rate, 50% duty cycle)
- No-load power consumption of power supply board <4 W
- High efficiency: 92% at 40 V, 2 A; 61% at 5 V, 1.75 A (measured between input and output of power supply board)

Block diagram

Figure 1 shows the block diagram of the VariLab 402. If this looks familiar, it's probably because of certain similarities to the professional lab power supply project published in the previous issue of Elektor. That's not surprising when you consider that Elektor Labs were involved in the development of that power supply as well.

A power module at the input supplies a fixed voltage of 48 V at a current of 3 A. This voltage is then reduced by a switching regulator to a voltage in the range of 3 to 43 V, depending on the output voltage set by the user. The output of the switching regulator section is fed to a linear regulator section, which reduces any ripple and noise present on the output voltage. The voltage drop over the linear section is set to a fixed value of approximately 3 V (see the description of the schematic diagram), which is high enough for good filtering and low enough to keep the dissipation of this section low (6 W at 2 A). The filtered voltage passes through a current sensing section to the output.

The operating functions and settings for the various sections are handled by a microcontroller and associated circuitry. The microcontroller receives information about the output voltage and current, and it adjusts the operating points of the switching and linear regulators based on the user settings. All settings and measured values are shown on a four-line display. There are also two rotary encoders and a pushbutton in the user interface. The microcontroller additionally has a USB port for connection to a PC, so that data can be exchanged between the two systems.

It doesn't take a lot of time or effort to draw a block diagram, but after that you have to work out all the practical details. We can mention some of the important choices here; the rest are covered in the description of the final version of the schematic diagram. Incidentally, in this article we only describe the power supply portion; the microcontroller and display portion will come next month.

Figure 1.
Block diagram of the VariLab 402. The overall regulator circuit consists of a switching regulator and a linear regular connected in series.

120437 - 12
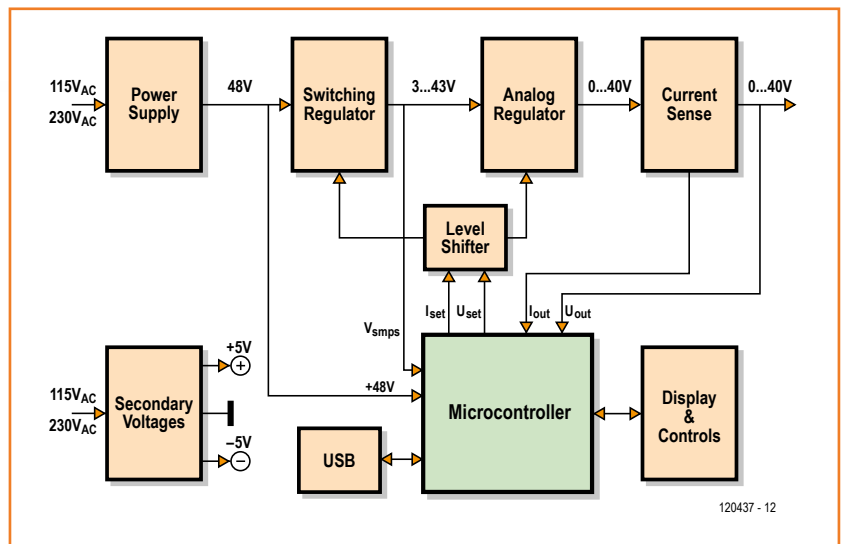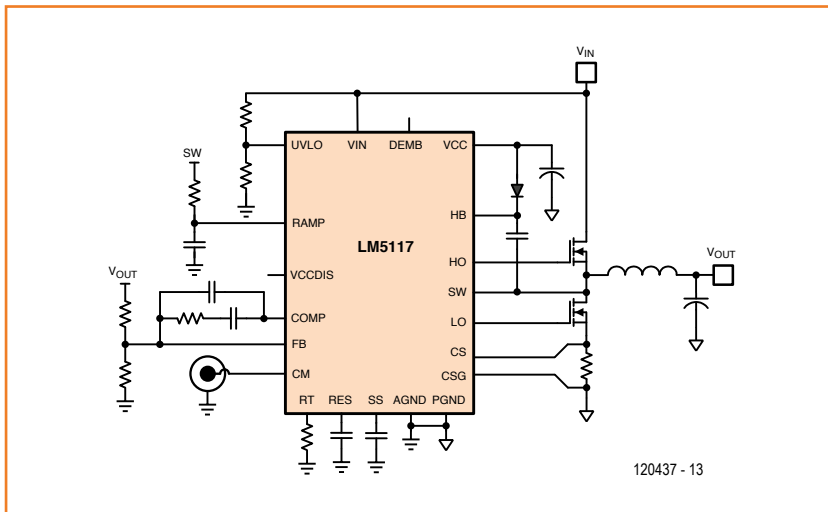
Figure 2.
Basic diagram of the
circuitry around the LM5117
buck converter controller.

To avoid using a large (and expensive) power transformer, we chose a standard commercial 150-W power module from Mean Well as the DC voltage source. It can supply slightly more than 3 A at a voltage of 48 V.

In our search for a suitable synchronous buck controller for the subsequent switching regulator stage, we discovered that there are not many ICs available that can handle 48 V. After several selection rounds, we settled on the LM5117 (for more information, see the datasheet [1] and application note AN-2103 [2]). The basic design of a buck converter built around this IC is shown in **Figure 2**. The design used in the actual circuit is fairly close to this, as can be seen later on. The IC controls two external power MOSFETs, which in turn drive an LC circuit at a relatively high switching frequency (approximately 100 kHz). The current through the MOSFETs is measured using a sense resistor in the source line of one of the MOSFETs. The other components are used to set the output voltages and various timing functions.

The actual circuit
Figure 3 shows the actual schematic diagram of the power supply portion. The microcontroller and display portion is not shown here—we're saving it for next month.

Switching section
The 48 V DC voltage from the power module enters at the left side of the schematic. It is chopped up by the buck converter built around IC1, and the pieces are sent to the LC network (L1 and C13–C16) by the power MOSFETS (T1

and T2). The pulse width depends on the feedback signal present on pin 8 of the IC.
The circuitry around IC1 largely corresponds to the standard application circuit from TI [2]. However, we opted for a lower switching frequency here (100 kHz). This results in less high-frequency noise on a double-sided PCB, as well as lower switching losses. Another consideration, perhaps even more important, is that the maximum duty cycle at 100 kHz is approximately 96%. The switching frequency is determined by the value of R4.
The soft-start time of the IC after power-up is set by C4. We chose a relatively small value here (470 nF), which yields a time of approximately 38 ms.
Voltage divider R2/R1 determines the input voltage level where the IC starts working. It is dimensioned for a value of approximately 44 V. The combination of R7, R8 and C8 filters the measurement signal from the voltage over R9 due to the current through the MOSFETs. According to the data sheet this is not essential, but we decided to include it anyhow.
C7 and R6 provide the sawtooth signal necessary for pulse width control. We chose a value of 1 nF for this capacitor, which is half of the maximum specified value. According to the formula in the data sheet, R6 should then have a value of 820 kΩ, but in practice we found that regulation at different output voltages was more stable with a value of 1.2 MΩ.
We stayed with the datasheet values for decoupling capacitors C9 (V_{CC}) and C10 (HB), since there was no reason to change them. The diode for the bootstrap function (D1) was chosen on account of its very low voltage drop (0.57 V at 1 A).
The NXP MOSFETS recommended in the data sheet for the switching transistors (T1 and T2) are truly ideal for this application; there's hardly anything better available. Gate resistors R11 and R10 prevent parasitic oscillations, but they also reduce the dead time. However, this is not a problem because the HO and LO outputs source more current to the MOSFETs than they sink from them, so T1 and T2 are switched off faster than they are switched on.
The snubber network R13/C12 suppresses voltage spikes that can occur at the MOSFET outputs due the inductance of L1.
The value of output inductor L1 was calculated using the formula in the data sheet, based on a fixed voltage of 48 V and a permissible ripple
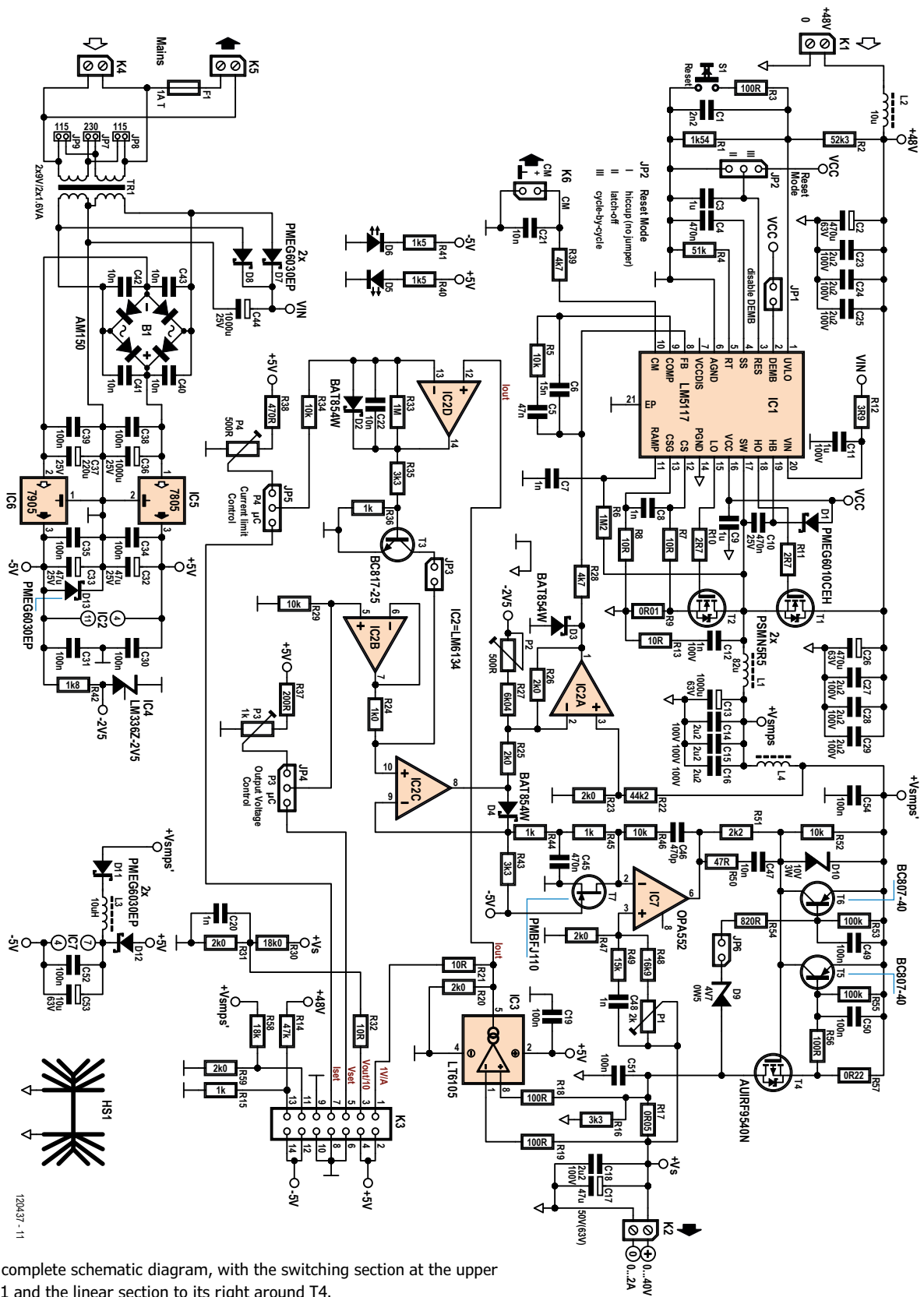
Figure 3. The complete schematic diagram, with the switching section at the upper left around IC1 and the linear section to its right around T4.

current of 40% of the maximum output current, which translates into 0.8 A. The calculated value was 83.33 mH, which is very close to a standard E12 value. The selected inductor from Würth Elektronik is significantly overdimensioned for what we need, but that has the advantage that the power dissipation is very low. The inductor can handle more than twice the actual output current without going into saturation. The SMPS section of this power supply can actually deliver a lot more current than the linear output section will ever need. We did that on purpose because it reduces the impact of the SMPS on the overall power supply regulation characteristics, particularly when it is operating in current regulation mode. We chose a value of 12 A as the actual limit. This may appear much too high, but we found that performance

was less stable in various tests with lower limits.

For buffer capacitor C13 we chose a type with low ESR, a high ripple current rating (2.77 A at 100 kHz) and long life (3,000 hours at 105 °C). The high ripple current rating is also necessary for handling large pulse loads. Capacitors C14, C15, C16 are connected in parallel to keep the ESR as low as possible, but even with this arrangement the ripple on the output is about 40 mV at 20 V. However, that is eliminated by the following linear section.

The internal voltage regulator of the LM5117 (7.6 V typical) is connected to a separate supply voltage provided by transformer TR1, which also provides the ±5 V supply voltages for the opamps. The internal power dissipation of IC1 is relatively low because this voltage is much lower than the 48 V supply voltage. Filter network R12/C11 suppresses high-frequency switching noise. The CM output of the LM5117 (pin 10) provides a signal proportional to the average output current, but it is only valid during continuous current flow. This signal is available for test purposes on connector K6.

The LM5117 has three different reset modes: hiccup, latch-off and cycle-by-cycle. Hiccup mode is usually the best option; see the data sheet for more information. The mode can be set by a jumper on JP2. After the jumper setting is changed, the regulator starts up in the new mode after the reset button (S1) is pressed or after the supply voltage is switched off and then on again. In hiccup mode capacitor C3 determines the soft-start time after current limiting has occurred.

Linear section

The linear regulator stage consists of a power MOSFET (T4) driven by an opamp (IC7). The advantage of using a MOSFET is that current can also flow backward through the MOSFET without any detrimental effects as long as the current is relatively low. We opted for a source follower configuration with a p-channel MOSFET to minimize the effect of current variation on the voltage drop over the MOSFET. IC7 drives the MOSFET via voltage divider R52/R51. This network is included to ensure that the opamp can completely cut off the MOSFET, since the minimum voltage between the output and the supply rail of IC7 (approximately 2.7 V) is not sufficient to allow the MOSFET to be cut off under worst-case conditions (U_{GS} −2 V min.).

The network R50/C47 is connected in parallel with R51 to make the control loop a bit faster. The value of C47 was chosen to be several times greater than the input capacitance of the MOSFET (1.3 nF).

For IC7 we chose the OPA552 opamp due to its special characteristics. It is suitable for supply voltages over the range of ±4 V to ±30 V and

has a maximum output current rating of 200 mA. Since output voltage adjustment down to 0 V is required and the voltage on the gate of the MOSFET must be negative under this condition, the opamp is provided with a negative supply voltage of −5 V.

The network R45/R46/C46 keeps regulation stable, especially in current regulation mode where the control loop is more complex. Components R49 and C48 in the feedback path from the power supply output to the non-inverting input of IC7 also stabilize the output characteristics, especially with transient loads. Note that the non-inverting input of IC7 is actually the inverting input of the linear output stage (IC7 and T4) because T4 inverts the signal. Capacitor C45 filters the opamp input and is also part of the control loop in current regulation mode. C51 suppresses high-frequency noise.

IC7 has a separate supply circuit consisting of D11, L3 and D12. This prevents the supply voltage for IC7 from dropping too low for properly driving T4 when the output voltage is set to a very low level (around 0 V). IC7 normally draws its supply voltage from the SMPS output voltage V_{smps}, ahead of T4, but diode D12 allows the 5 V supply rail to take over when necessary. FET T7 is placed at the inverting input of IC7 to avoid problems with IC2c control of the output voltage of IC7 at low power supply output voltage levels (under 3 V) in the absence of the ±5 V supply voltages. The FET conducts when the −5 V supply voltage is absent, which causes T4 to be cut off so the output voltage drops.

The output current is measured using a sense resistor (R17) in series with the output. Compared to the usual arrangement with the sense resistor in the return line, this has the advantage that all ground points of the circuit are connected directly together. IC3 measures the voltage over R17, which is directly proportional to the current through R17. The output voltage is buffered and filtered by C17 and C18.

To compensate for the higher output impedance resulting from the addition of R17, the feedback network to the opamp IC7 (R47, R48 and P1) is connected directly to output connector K2. This way all resistances between the MOSFET and K2, including all solder joints, are included in the control loop. The output voltage can be adjusted by P1, or optionally by the microcontroller.

We chose a type LT6105 IC for the current-sense

opamp IC3. This IC has a large input voltage range extending up to 44 V, while requiring only 5 V as its supply voltage. See the data sheet [3] for a more detailed description. The gain is determined by the ratio between R20 and R18 or R19. Here the gain is set to 20, which results in an output signal of 1 V/A on R20. From measurements we discovered that the accuracy of the sense amplifier is somewhat dependent on the output voltage, but as long as you limit yourself to a three-digit readout you don't need to worry about this. For a more detailed explanation, see the description on the Elektor Labs website [4]. The power MOSFET has its own protection circuit to safeguard it against short-circuit conditions. For this purpose, a 0.22 Ω current sense resistor (R57) is placed ahead of the MOSFET. When the current through R57 is approximately 2.7 A, the voltage over the resistor will be large enough to drive transistor T5 into conduction, which reduces the gate-source voltage. This is a very fast form of current limiting. After that the microcontroller has to take over and ensure that the voltage over the MOSFET, and thus the power dissipated to the heat sink, remains within limits. There is a similar arrangement for the voltage over T4. If it ever becomes too high (the normal voltage drop is only 2.6 V), transistor T6 is driven into conduction by D9 and R54 and the MOSFET is cut off. The microcontroller normally keeps an eye on this, but if you use the power supply portion on its own (without the µC/display board) you can enable this protection by fitting a jumper on JP6.

Regulation circuitry

The output voltage and maximum output current can be set using multiturn potentiometers on the PCB (P3 and P4) or by using the microcontroller (µC/display board). Jumpers JP4 and JP5 select the desired option. The control signal on JP4 for the voltage setting is buffered by IC2b. Resistor R24 in series with the output limits the output current when transistor T3 conducts. This transistor pulls the signal to ground when the set current level is exceeded. If jumper JP3 is removed, current limiting is handled solely by the microcontroller (no hardware limiting).

Opamp IC2d compares the control voltage for current limiting on JP5 to the output current measured by IC3. To ensure stable regulation, the gain and bandwidth of the opamp are limited by R33, R34 and C22. Schottky diode D2 limits the negative output excursion of IC2d. If the output

current exceeds the set value, IC2d drives T3 into conduction, which pulls the non-inverting input of IC2c low.

As previously mentioned in the description of the block diagram, we opted for a fixed voltage drop over the linear regulator section around T4. Here the level shifter shown in the block diagram consists of IC2c and D4. The control voltage for the linear section (at the junction of R43 and R44) is equal to the voltage at JP4. The control voltage for the switching section (on R25) is somewhat higher due to the voltage drop over D4. The

value of R43 determines the current through D4 and therefore the voltage over D4. At an output voltage of 0 V (5 V over R43), we measured a voltage of 0.252 V over D4 in our prototype, and with an output voltage of 40 V (9 V over R43) we measured 0.268 V. That is steady enough for our purpose. Since the control around IC1 and IC2a, and the control around T4 and IC7 both have a gain of 10, the voltage difference between the two sections is 2.52 to 2.68 V.
Opamp IC2a supplies the control signal for the feedback input of IC1. Components R27 and P2 are included to enable compensation for the

Component List

Resistors
Default: SMD 0805
R1 = 1.54kΩ 1%, 125mW
R2 = 52.3kΩ 1%, 125mW
R3,R18,R19,R56 = 100Ω 1%, 125mW
R4 = 51kΩ 1%, 125mW
R5,R29,R34,R46,R52 = 10kΩ 1%, 125mW
R6 = 1.2MΩ 1%, 125mW
R7,R8,R21,R32 = 10Ω 1%, 125mW
R9 = 0.01Ω 5%, 1 W (TE Connectivity/CGS CGSSL1R01J)
R10,R11 = 2.7Ω 1%, 125mW
R12 = 3.9Ω 5%, 125mW
R13 = 10Ω 1%, 0.75W (Vishay Draloric CRCW201010R0FKEF, 2010)
R14 = 47kΩ 1%, 125mW
R15,R24,R36,R44,R45 = 1kΩ 1%, 125mW
R16 = 3.3kΩ 1%, 1W (Multicomp, RCL1218 3K3 1% 100 PPM/K E3, 1218)
R17 = 0.05Ω 1%, 1W (Bourns, CRM2010-FZ-R050ELF, 2010)
R20,R23,R25,R26,R31,R47,R59 = 2.00kΩ 1%, 125mW
R22 = 44.2kΩ 1%, 125mW
R27 = 6.04kΩ 1%, 125mW
R28,R39 = 4.7kΩ 1%, 125mW
R30,R58 = 18.0kΩ 1%, 125mW
R33 = 1MΩ 1%, 125mW
R35,R43 = 3.3kΩ 5%, 125mW
R37 = 200Ω 1%, 125mW
R38 = 470Ω 1%, 125mW
R40,R41 = 1.5kΩ 5%, 125mW
R42 = 1.8kΩ 5%, 125mW
R48 = 16.9kΩ 1%, 125mW
R49 = 15kΩ 5%, 125mW
R50 = 47Ω 5%, 125mW
R51 = 2.2kΩ 5%, 125mW
R53,R55 = 100kΩ 5%, 125mW
R54 = 820Ω 5%, 125mW
R57 = 0.22Ω 5%, 5 W (TE Connectivity/CGS, SMW5R22JT, SMW5R22JT)
P1 = 2kΩ 10%, 0.5 W, 25-turn trimpot, top adjustment (Bourns 3296Y-1-202LF)
P2,P4 = 500Ω 10%, 0,5 W, 25-turn trimpot, top adjustment (Bourns, 3296Y-1-501LF)
P3 = 1kΩ 10%, 0.5 W, 25-turn trimpot, top adjustment (Bourns, 3296Y-1-102LF)

Capacitors
Default: SMD 0805
C1 = 2.2nF 50V, 5%, C0G/NP0

C2,C26 = 470µF 63V, 20%, ESR 0.027 Ω, I_{AC} 2A (Panasonic EEUFR1J471)
C3,C9 = 1µF 16V, 10%, X7R
C4,C10,C45 = 470nF 25V, 10%, X7R
C5 = 47nF 50V, 10%, X7R
C6 = 15nF 50V, 10%, C0G/NP0
C7,C8,C12,C20,C48 = 1nF 100V, 5%, C0G/NP0
C11 = 1µF 100V, 10%, X7R (Murata GRM32CR72A105KA35L, 1210)
C13 = 1000µF 63V, 20%, D 16mm, 7.5mm pitch, I_{AC} 2.77 A (Panasonic EEUFC1J102)
C14,C15,C16,C18,C23,C24,C25,C27,C28,C29 = 2.2µF 100V, 10%, X7R (TDK C3225X7R2A225K, 1210)
C17 = 47µF 50V, 20%, D 10 mm, 5mm pitch, ESR 29mΩ (Nichicon PLX1H470MDL1TD)
C19,C30,C31,C49,C50 = 100nF 25V, 10%, X7R
C21,C22,C47 = 10nF 50V, 10%, X7R
C32,C33 = 47µF 35V, 20%, 2.5mm pitch
C34,C35,C38,C39 = 100nF 63V, 5%, 5/7.5mm pitch
C36,C44 = 1000µF 25V, 20%, D 10mm, 5mm pitch, ESR 0.02Ω (Panasonic EEUFR1E102)
C37 = 220µF 25V, 20%, D 8mm, 3.5mm pitch, I_{AC} 950 mA (Panasonic ECA1CM221)
C40–C43 = 10nF 100V, 10%, 5mm pitch
C46 = 470pF 100V, 10%, X7R
C51,C52,C54 = 100nF 100V, 10%, X7R
C53 = 10µF 63V, 20%, D 6.3mm, 2.5mm pitch (Rubycon 100YXF10MEFC6.3X11)

Inductors
L1 = 82µH, 7A, SMD, high-current, R_{DC} 30.4mΩ (Würth Elektronik 74435588200)
L2 = 10µH, 7.2A, SMD, high-current, R_{DC} 16.3mΩ (Würth Elektronik 7443251000)
L3 = 10µH, 950 mA, radial, 2mm pitch, R_{DC} 0.14Ω (Murata Power Solutions 11R103C)
L4 = 70Ω @ 100MHz, 3.5 A, 0.022Ω, 0603 (Murata BLM18KG700TN1D)

Semiconductors
D1 = PMEG6010CEH (SOD-123F)
D2,D3,D4 = BAT85W (SOT-123)
D5 = LED, red, 3mm, leaded
D6 = LED, green, 3mm, leaded
D7,D8,D11,D12,D13 = PMEG6030EP (SOD-128)
D9 = 4.7V 500mW zener diode (SOD-123F)
D10 = 10V 3W zener diode (SMB)
B1 = 50V 1.5A bridge rectifier (Multicomp AM150)
T1,T2 = PSMN5R5-60YS (LFPAK/SOT669)

internal reference voltage of IC1 (0.8 V). They are connected to a stable −2.5 V reference voltage provided by IC4. In theory this arrangement can be used to reduce the output voltage of the switching section all the way to 0 V, but that is not necessary in this design with a liner regulator output stage. If desired (or if necessary), you can use P2 to shift the output voltage of the SMPS slightly. Schottky diode D3 at the output of IC2a prevents the output of this opamp from going negative, since the LM5117 is not designed to handle that. For more information about the dimensioning of this section, see [4].

Microcontroller connection

All relevant signals on the power supply board, as well as signals for the onboard supply voltages, are available on connector K3. It is connected to the µC/display board, which will be described in next month's issue. The signals for the measured output current (1 V/A) and the measured output voltage (V_{out}/10 via voltage divider R30/R31) are on pins 1 and 3, and the microcontroller can provide the control voltages for the voltage and current settings on pins 5 and 7. The signal for the voltage at the output of the switching section (V_{smps}/10 via R58/R59) is available on

T3 = BC817-25 (SOT-23)
T4 = AUIRF9540N (TO-220)
T5,T6 = BC807-40 (SOT-23)
T7 = PMBFJ110 (SOT-23)
IC1 = LM5117 (MXA20A)
IC2 = LM6134 (SO-14)
IC3 = LT6105 (MSOP-8)
IC4 = LM336Z-2.5 (TO-92)
IC5 = 7805 (TO-220)
IC6 = 7905 (TO-220)
IC7 = OPA552UA (SO-8)

Miscellaneous
K1,K2 = 2-way PCB screw terminal block, 5mm pitch
K3 = 14-pin boxheader, 0.1"pitch

K4,K5 = 2-way PCB screw terminal block, 7.5mm pitch
K6,JP1,JP3,JP6 = 2-pin pinheader, 0.1"pitch
JP2,JP4,JP5 = 3-pin pinheader, 0.1"pitch
JP1–JP6 = jumper 0.1"pitch
S1 = pushbutton 6x6mm SPST-NO (TE Connectivity/Alcoswitch FSM4JH)
F1 = 1AT (slow) fuse with PCB mount fuse holder and cap
TR1 = power transformer, PCB mount, prim. 2x115V, sec. 2x9V, 3.2VA (Block AVB3.2/2/9)
HS1 = heatsink, PCB mount, e.g. Fischer SK 129 38,1 STS, 6.5 K/W
Power Supply Module 48V 3A, e.g. Mean Well S-150-48
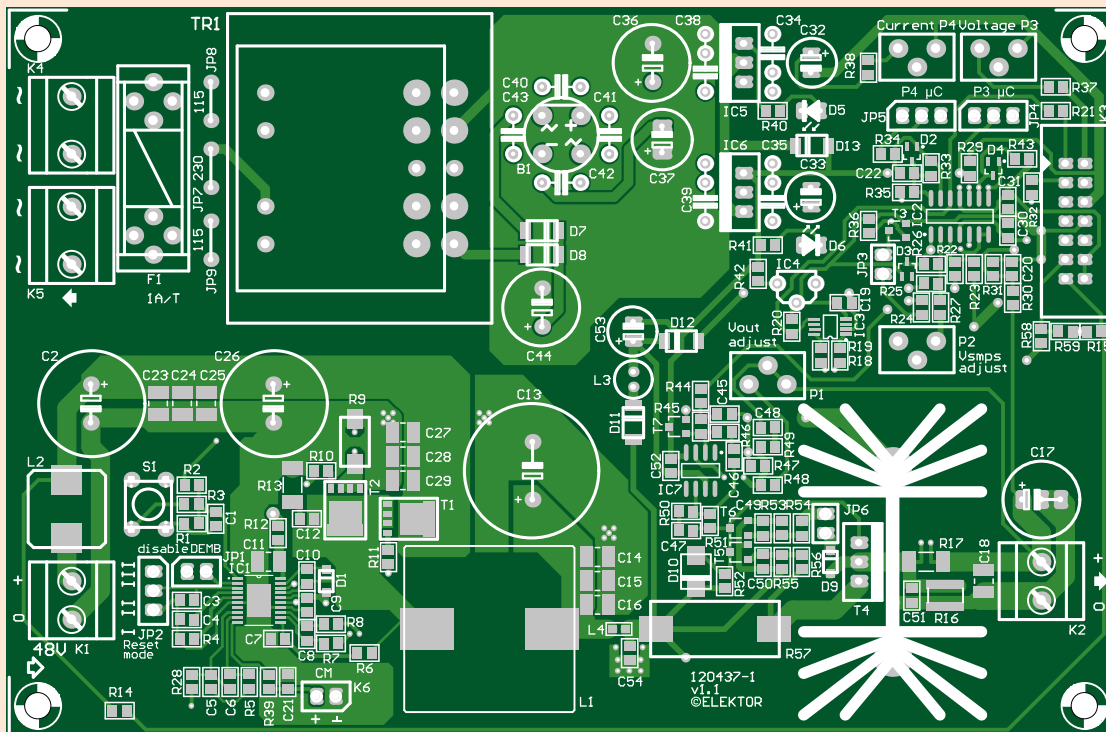PCB # 120437-1

Figure 4.
The circuit board for the lab power supply has a mix of SMDs and leaded components—not exactly the ideal soldering exercise for beginners.

pin 11, and the signal for the 48 V input voltage (+48V/48 via R14/R15) is available on pin 13.

Power source

As mentioned already at the start of this article, the 48 V supply voltage is provided by a ready-made power module from Mean Well (*sic*) with an output current rating of 3 A. Of course, you could also use a different type or brand with the same output specifications.

The supply voltages for the opamps are provided by a small analog power supply section to avoid the effects of switching noise. It consists primarily of the PCB transformer TR1 (which can be configured for 115 or 230 V AC line voltage with wire links JP7, JP8, JP9), bridge rectifier B1 and a pair of voltage regulators (IC5 and IC6) with their peripheral components. As previously mentioned, the transformer also provides the supply voltage for IC1 via diodes D7 and D8. Schottky diode D13 is placed at the output of IC6 to protect the negative regulator against positive supply voltages on its output (e.g. during shutdown). This diode prevents the voltage from rising above 0.2 V. The ±5 V supply voltages are also available on connector K3 for the µC/display board. LEDs D5 and D6 indicate the presence of the two supply voltages. The AC line of the power module for the 48 V supply voltage can be connected via K5. Fuse F1 is a supplementary safety measure, since you never know what might happen with an external power module. Suitable decoupling measures have been taken at the input for the 48 V supply voltage and the input of the switching MOSFETs. They start with choke L2 (10 mH, R_{DC} 16.3 mΩ), which is followed by C2, C23, C24 and C25 in parallel. Capacitors C26–C29 provide decoupling for the MOSFETs. C2 and C26 have an ESR of just 27 mΩ and can handle nearly 2 A at 100 kHz. The other capacitors are SMDs, which are very effective at relatively high frequencies

Construction

Now that we've described just about every component on the schematic, it's time to look at how to build the project. **Figure 4** shows the layout of the double-sided PCB designed for this power supply. There's not a lot that needs to be said about construction, although you do need to have some experience with soldering SMDs—otherwise you are bound to get into trouble. Be sure to use exactly the components specified

in the components list. In particular, some of the electrolytic capacitors absolutely cannot be replaced by types with lower specs. MOSFET T4 is mounted on a small heat sink. Insulating hardware must be used for mounting the MOSFET because the heatsink is connected to ground by its mounting pins.

Fit one or two wire links to configure the AC line voltage setting—for 230 Vac fit a wire link JP7; for 115 V fit wire links JP8 and JP9.

At this point you are just about ready to connect a power module with an output voltage of 48 V and see whether the circuit works. For this purpose you must fit jumpers on JP3, JP4 (P3 connection), JP5 (P4 connection) and JP6. First set P1 and P2 to the midrange position, and then you can switch on AC power. Connect a moderate load to K2 along with a voltmeter and an ammeter and see whether you can adjust the output voltage with P3. It should also be possible to adjust P4 to limit the current, causing the

output voltage to drop.

If everything works properly so far, you are ready for the second part of the project, where we discuss the µC/display board and the associated software and describe how to case up the complete power supply.

(120437-I)

Web Links

[1] www.ti.com/lit/ds/symlink/lm5117.pdf

[2] www.ti.com/lit/ug/snva466b/snva466b.pdf

[3] http://cds.linear.com/docs/en/datasheet/6105fa.pdf

[4] www.elektor-labs.com/120437

Microcontroller BootCamp (7)
The I²C bus

By **Burkhard Kainka**
(Germany)

If you are running short of I/O lines on an Arduino Uno board, a remedy is available. The I²C bus needs only two port pins and can address up to 127 external ICs. There are countless devices available with I²C interfaces, including simple port expanders, EEPROMs and a wide variety of sensors. In the final instalment of our Bascom course series, we show you how the I²C bus works. As usual, the main focus is on interesting demo applications.

The Inter-IC bus or I²C bus (barring code, also sloppily written as I2C) is a two-wire data bus consisting of a data line and a clock line, originally developed by Philips (not: Phillips) for consumer electronics applications. Most television sets or video recorders have a central processor that controls a large number of modules. With a data bus consisting of just two lines, connecting all these modules to the processor is easy. The processor is the bus master, as with the SPI bus, and the peripheral devices are the slaves. A particular feature of the I²C bus is that every slave device has a 7-bit address. That allows a large number of ICs to be connected to the bus without interfering with each other. Along with RAMs, EEPROMs, port expanders, real-time clocks, A/D and D/A converters, there are a large number of special-purpose I²C devices such as display drivers, PLL ICs and many others. An excellent reference book on I²C is available from Elektor, see **Further Reading** [a].

Data transfer and addressing
The I²C bus consists of a serial data line (SDA) and a clock line (SCL). One bit per clock pulse (as in a shift register) is transferred on the data line. Usually the address bits are sent first, followed by the data bits. Each line has a pull-up resistor, and each line can be pulled low by any device on the bus. **Figure 1** shows the basic bus architecture. The master generates the clock signal. The data can come from the master or from the slave.

The I²C bus can work with 5 V microcontrollers and ICs as well as 3.3 V devices. You can even connect both types to the same bus. The two pull-up resistors, which typically have a value of 2.2 kΩ, hold the bus lines at 3.3 V or 5 V (logic High level) when the lines are not pulled low by any of the pull-down transistors in the devices connected to the bus. Any 5 V devices on the bus also see 3.3 V as a High level because it is significantly higher than 2.5 V (half of the supply voltage), and of course 0 V is logic Low in any system. This means that you can easily connect the Arduino Uno board to a 3.3 V slave device. That's handy because many recent ICs are only designed to operate at 3.3 V.

The ATmega328 on the Uno board has an integrated hardware I²C interface connected to pins PC4 and PC5. However, Bascom also has special commands that can be used to implement a software I²C interface using any desired port pins, and of course you can also write your own functions to set the lines High or Low using individual code lines. Here we only use the Bascom software I²C interface, but with the same port pins as used by the hardware I²C interface integrated in the microcontroller.

These port pins are also used on the Elektor Extension shield [1] for the Arduino Uno. There they are routed to the EEC/Gnublin connector K2, which can be used to connect Gnublin modules with an I²C interface over a flat cable, such as a module with eight relays [2]. The bus lines also have 330 Ω series resistors, which can be omit-

ted if desired. However, they provide protection against false signals resulting from reflections on long bus lines, and they can help avoid problems that may occur on buses with devices operating at different supply voltages. For example, the input currents of 3.3 V peripheral devices could exceed the maximum rated value if one of the bus lines is accidentally set to a 5 V high level for a prolonged period. The series resistors limit the input current to a safe level.

The I²C bus protocol defines several specific states that allow every device on the bus to detect the start and end of a transfer:

- Quiescent state: SDA and SCL are high and therefore inactive. The Bascom instruction **I2cinit** puts both lines in the quiescent state but without internal pull-up resistors, since they are located externally.
- Start condition SDA is pulled low by the master while SCL remains high (Bascom instruction: **I2cstart**).
- Stop condition: SDA changes from low to high while SCL remains high (Bascom instruction: **I2cstop**).
- Data transfer: The current sender places eight data bits on the data line SDA, which are shifted out by clock pulses on the clock line SCL generated by the master. The transfer starts with the most significant bit (Bascom instruction: **I2cwbyte Data**).
- Acknowledge (Ack): The currently addressed receiver acknowledges reception of a byte by holding the SDA line low until the master has generated a new clock pulse on the SCL line. This acknowledgement also means that another byte is expected to be received. If the device wishes to end the transfer, it must indicate this by omitting the acknowledgement (Nack).
  The transfer is terminated by the stop condition (Bascom instruction: **i2crbyte Data, Ack** or **i2crbyte Data, Nack**).

Addresses are transferred and acknowledged in the same way as data. In the simplest case of a data transfer from the master to a slave, such as an output port, the following procedure is used. The master generates the start condition and then transfers the address of the port IC in bits 1 to 7 and the desired direction of the data transfer in bit 0 – in this case, 0 for writing to the device. The address is acknowledged

by the addressed slave. Then the master sends the data byte, which is also acknowledged. The connection can be ended now by generating the stop condition, or another byte can be sent to the same slave.

I²C bus address with data direction							
A6	A5	A4	A3	A2	A1	A0	R/W

If the master want to read data from a slave, the address must be sent with the data direction bit set to 1. The master then generates eight clock pulses and receives eight data bits. If reception is confirmed by an acknowledgement on the ninth clock pulse, it can receive another data byte. At the end the master terminate the transfer with a stop condition when no acknowledgement is received.

Every I²C device has a fixed address. Part of the address is specific to the device type, and the rest can be configured by the user with the address lines A0, A1, etc. fed out from the device. These address lines are tied high or low in the circuit to set the address bits. If the device has three address lines fed out, such as the PCF8574 port expander, up to eight different addresses can be set. This means that up to eight devices of the same type can be connected to one bus. This port expander provides eight digital outputs, and the signal levels on the outputs are determined

Figure 1.
I²C bus connections between master and slave devices.

140293 - 11
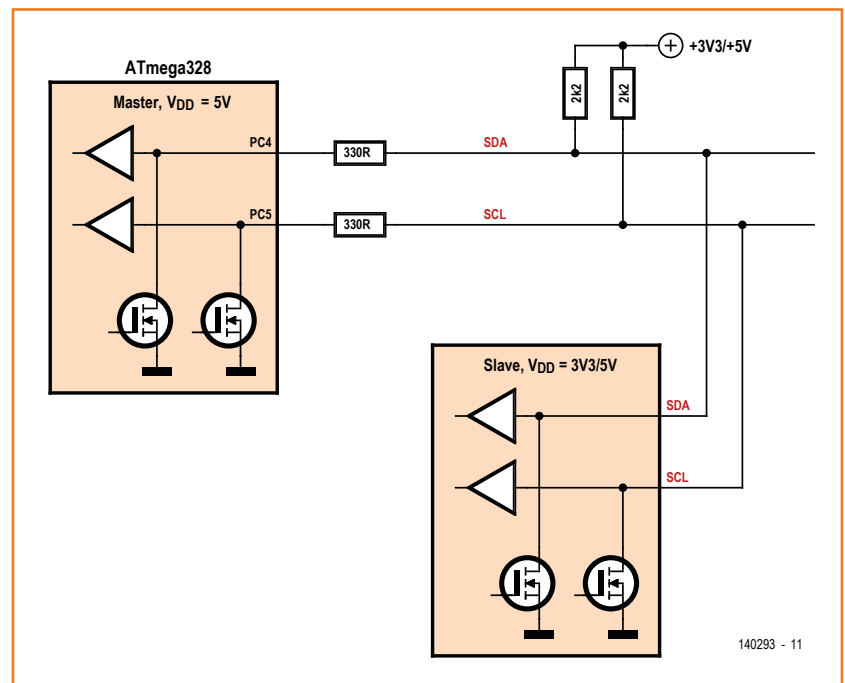
Listing 1.
Polling for active I²C addresses.

```
'-----------------------------------
'UNO_I2C1.BAS Test for valid Adresses
'-----------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim Addr As Byte
...
Config Scl = Portc.5
Config Sda = Portc.4
I2cinit

For Addr = 2 To 254 Step 2
  I2cstart
  I2cwbyte Addr
  If Err = 0 Then
    Print Addr
    Locate 2 , 1
    Lcd Addr
    Waitms 1000
  End If
  I2cstop
Next Addr
End
```

Many ICs allows the user to set the last three address bits. The PCF8583 real-time clock IC has the same base address as a RAM because it also contains a RAM. If you want to use a RAM and a real-time clock on the same bus, you have to give them different addresses.

Incidentally, there are two different notations for the address, which sometimes cause confusion and laborious troubleshooting. Some data sheets only give the 7-bit address without the read/write bit. In that notation the base address of the PCF8574 would be 20_{hex} (decimal 32) By contrast, in Bascom this IC has a write address of 64 (40_{hex} = **&H40**) and a read address of 65 (**&H41**).

To learn the addresses of the devices connected to the bus, you can use the small Bascom program shown in **Listing 1** (downloadable from [3]). It polls every possible bus address to see whether a device responds. After a device address is output, the Bascom system variable ERR (which does not have to be declared with **Dim** because Bascom has already done this for you) is set to 1 if no acknowledgement is received or to 0 if an Ack signal is received. The latter case means that the address is valid. All even addresses from 2 to 254 are tested, since the odd addresses are the corresponding read addresses of the same devices. For the circuit shown in **Figure 2**, the program reports the addresses 64, 144, 160 and 162 just as expected.

There's another special feature of the I²C bus protocol: every device on the bus can halt the master for a while if it needs a bit more time. To do so it pulls the clock line Low, which forces the master to wait until the line is released again. Bascom follows this convention faithfully. How-

by the eight bits in the data byte sent to the IC. The base address of the port expander is 40_{hex} (decimal 64). The base addresses of some typical first-generation Philips I²C devices are:

- PCF8591 A/D converter: 90_{hex} (decimal 144)
- RAMs and EEPROMs: $A0_{hex}$ (decimal 160)
- PCF8583 real-time clock: $A0_{hex}$ (decimal 160)
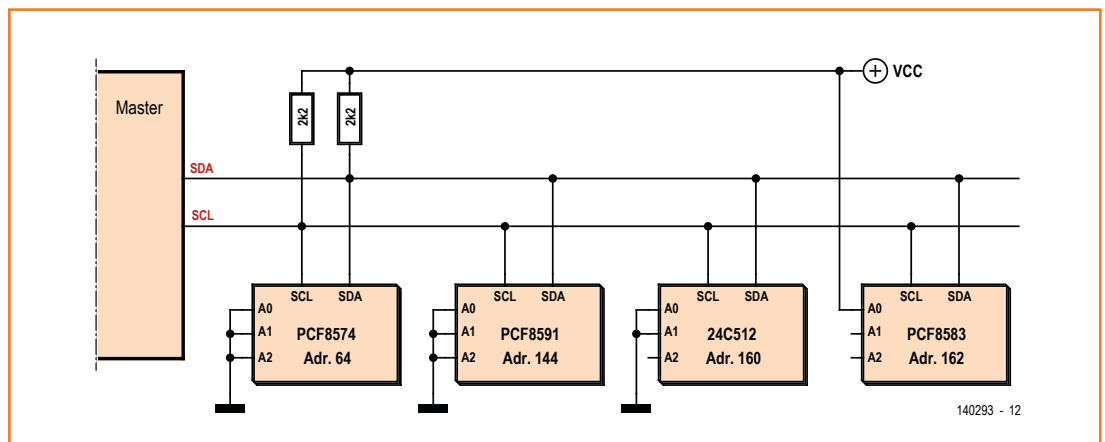
Figure 2.
A master and four slaves.

ever, this means that a program that uses the I²C bus will hang if no pull-up resistors are connected. From other projects you may be used to the idea that you can sometimes test software without connecting the associated hardware. As a result, you might find yourself staring at an oscilloscope while checking out your software to see whether there are any signals at all on the SCL and SDA lines, while the cable to the Gnublin board is not yet connected. What you have overlooked is that the I²C bus pull-up resistors are on the Gnublin board. Since no pull-ups are present on the host board, everything remains in suspended animation and there are no signals on the I²C bus.

The PCF8574 port expander

The PCF8574 is a port expander IC with eight bidirectional port pins. It does not have a data direction control function. Instead, the port pins have internal pull-up resistors that cause a high level to be present in the quiescent state. This means that after booting you will initially read the port status 255 (binary 11111111). You will only see low values (logic 0) if the port pins are actively pulled to ground by an external device. It is possible to use some of the port pins as outputs and the others as inputs. This requires first configuring all of the input pins in the high state, the same as the output pins.

Figure 3 shows the bus connections and a potential application of the port expander as a digital tester. The port expander can be powered from 3.3 V or 5 V according to the operating voltage of the item under test. Any desired digital circuit with four inputs can be driven using the output port pins P0 to P3. For example, you could apply an incrementing digital value to these four pins to obtain all possible combinations of signal levels on the circuit inputs. Cables and connectors can also be tested the same way. Every open circuit and short circuit can be detected.

Listing 2 shows an example of how to use a mixed set of inputs and outputs. Here pins P0 to P3 output a continuously incrementing digital value. Pins P4 to P7 are used as inputs and must therefore be set high in the write operation (**Or &B11110000**) The IC is addressed twice: first in the write direction (address 64) and then in the read direction (address 65). Incrementing the value on the four outputs is the simplest possible type of test stimulus. Depending on the item under test, a completely different test sequence
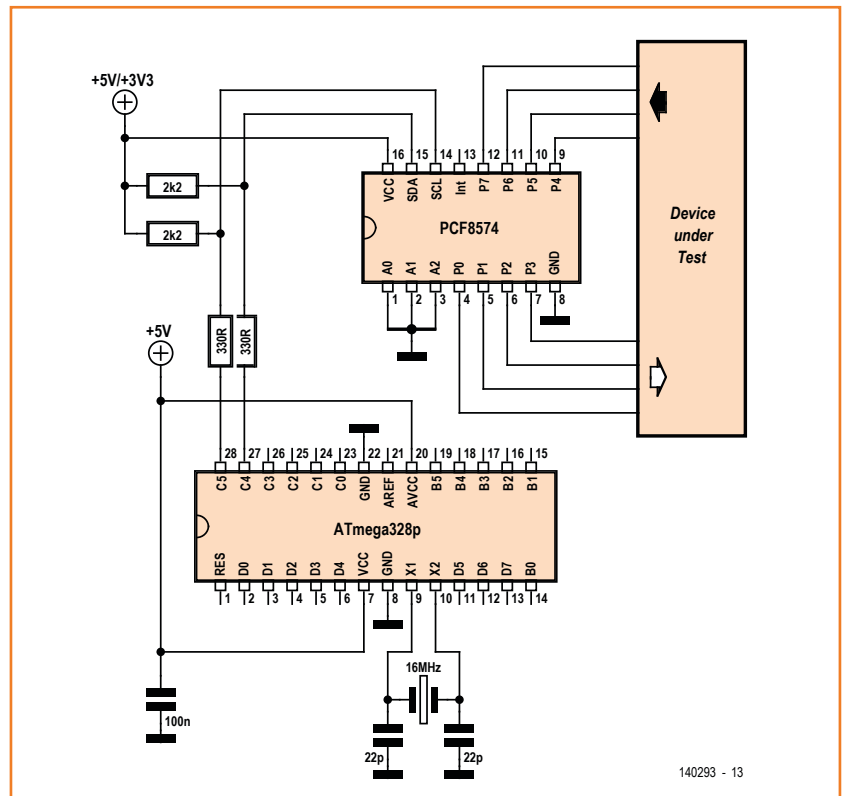
may be necessary, which means you will need different test software.

PCA9555 16-bit I/O port

Sometimes eight more lines are not enough. The Gnublin port expander module [2] provides 16 I/O lines using an NXP PCA9555 IC. The board can be plugged directly into the EEC connector on the Elektor Extension shield. There is also a Gnublin relay board that uses the same IC. NXP is the successor to Philips, and the PCA9555 is the rightful successor to the PCF8574. That's why the two devices have the same bus address: 64 (**&H40**). This sort of address recycling makes sense because the address space is limited. In any case, why would you need a PCF8574 when you have a PCA9555? Along with twice as many I/O pins, the new IC has additional functions such as data direction control and inversion of the input data.

Figure 4 shows how the IC can be connected to the Elektor Extension shield using the EEC connector. The I²C bus lines and supply voltage lines are connected using a flat cable. The two pull-up resistors on the Gnublin board are connected to the bus lines by a pair of jumpers. These jumpers must be fitted if only one board is connected.

Figure 3.
Using the PCF8574 port expander.

Listing 2. Accessing the PCF8574 ports.

```
'----------------------------------
'UNO_I2C2.BAS input/output PCF8574
'----------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim N As Byte
Dim D As Byte
...
Do
  For N = 0 To 15
    I2cstart
    I2cwbyte 64          'write
    D = N Or &B11110000
    I2cwbyte D
    I2cstop
    Print N;
                              Print " ";
                              Locate 2 , 1
                              Lcd N
                              Lcd " "

                              I2cstart
                              I2cwbyte 65            'read
                              I2crbyte D , Nack
                              I2cstop
                              Shift D , Right , 4
                              Print D
                              Locate 2 , 5
                              Lcd D
                              Lcd " "
                              Waitms 100
                          Next N
                      Loop
```

If you use several boards, perhaps connected using the Gnublin distributor board [2], make sure that the pull-up resistors are only enabled on one board. It is possible to use several port expander boards because each board has jumpers for configuring address lines A0 to A2. In the default state, all three lines are tied to ground.

That yields a bus address of 64. A total of eight boards can be connected, with addresses from 64 to 72. With 16 I/O pins per board, that gives you a grand total of 128 I/O lines.

Listing 3 shows an application for the PCA9555, which can also be used for testing other circuits. As in the previous example, the port pins are

Listing 3. Using the PCA9555 port expander.

```
'----------------------------------
'UNO_LCD1.BAS input/output PCA9555
'----------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim N As Byte
Dim D As Byte
...
I2cstart
I2cwbyte 64      'Gnublin port expander
I2cwbyte 6
I2cwbyte 0       'Port 0 output
I2cwbyte 255    'Port 1 input
I2cstop

Do
  For N = 0 To 255
    I2cstart
    I2cwbyte 64      'write
    I2cwbyte 2
                              I2cwbyte N
                              I2cstop
                              Print N;
                              Print " ";
                              Locate 2 , 1
                              Lcd N
                              Lcd " "

                              I2cstart
                              I2cwbyte 64        'write
                              I2cwbyte 1
                              I2cstart
                              I2cwbyte 65        'read
                              I2crbyte D , Nack
                              I2cstop
                              Print D
                              Locate 2 , 5
                              Lcd D
                              Lcd " "
                              Waitms 100
                          Next N
                      Loop
```

divided up with half of them used for outputs and the other half used for inputs. Since things are a bit more complicated here, a command byte has to be sent after each address [4]. The command selects a register address in the IC. After that you can send or receive one or two bytes of data. For example, if you want to configure port 0 (with eight pins) and port 1 (with the other eight pins), you first send the command '6' after the address and then write two data bytes, which are placed in registers 6 and 7. Zero bits in these bytes mean that you want to configure the corresponding pins as outputs. One bits stand for inputs. In our example, all pins of port 0 are configured as outputs and all pins of port 1 are configured as inputs. Incidentally, pins configured as inputs have integrated high-impedance pull-up resistors (approximately 100 kΩ), so open inputs are read as logic 1. You can use the following command byte values:

0   Input Port 0
1   Input Port 1
2   Output Port 0
3   Output Port 1
4   Polarity Inversion Port 0
5   Polarity Inversion Port 1
6   Configuration Port 0
7   Configuration Port 1

The commands 2 (Output Port 0) and 1 (Input Port 1) are used iteratively in the data loop of the sample program. The IC must be addressed anew
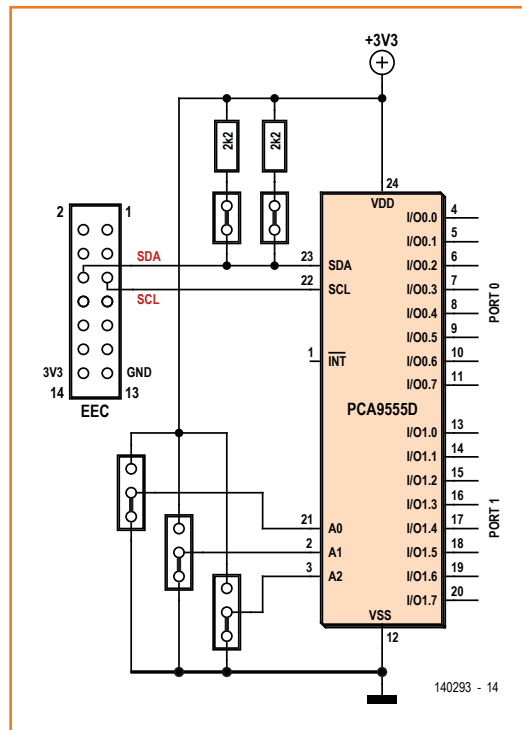
Figure 4.
The PCA9555 on the Gnublin board.

for each command. In order to read a port, the IC must also be addressed again with the read bit set (address 65). If you examine the code closely, you may wonder why there is no **I2cstop** instruction. That's because the code implements a 'repeated start' without a previous stop condition, since the two accesses always belong together: writing the command to select the register to be read and reading the register contents.

Other interesting I²C components

Anyone who reads Elektor regularly is always running into interesting ICs with an I²C interface. They often form the inspiration for new projects. Some particularly significant types are:

- I²C EEPROMs up to 64 KB (for example, the 24C512). These can be used to build data loggers and lots of other things.
- The CY27EE16 is a crystal clock generator that can be programmed over the I²C bus. It is used in the Elektor Software Defined Radio project. Software control with a microcontroller opens the door to new possibilities.
- The SI4735 is a complete AM/FM receiver and has already been used in several Elektor projects along with Bascom. Any desired frequency can be set using just a few I²C commands.
- High-resolution A/D and D/A converters often have I²C ports. One example is the ADS1115 16-bit A/D converter recently described in Elektor.

If you want to delve deeper into this subject, you can even build your own I²C bus IC. A Bascom library for programming I²C slave devices is available for this purpose. That is a bit more difficult than programming a bus master because the slave device must be able to handle the bus speed set by the master. The *Mastering the I²C Bus* book has all the details.

Analog I/O with the PCF8591

The PCF8591 contains an 8-bit A/D converter with four inputs along with an 8-bit D/A converter in the same package. You probably won't need the A/D converter by itself because the Arduino Uno already has enough analog inputs and higher resolution for A/D conversion. However, a real A/D converter can come in handy. Unlike a PWM output, it delivers a true DC voltage. You can use this to build a simple diode tester that measures the forward voltage at a defined current level (see **Figure 5**). The circuit operates at 5 V so that it can also be used to measure the relatively high forward voltage of LEDs.

The IC has a control register that must be written right after the bus address is sent. A control byte value of 64 enables the D/A converter and selects input channel 0. The following byte is put into the D/A register and results in an output voltage in the range of 0 to 5 V, corresponding to a data range of 0 to 255. With this 8-bit resolution, the output voltage increment is approximately 20 mV. The demo program in **Listing 4** generates a rising voltage ramp and measures the voltage across the 1 kΩ sense resistor at the same time. The PCF8591 has to be addressed again in the read direction (address 145) to read the measured voltage. The read byte value rep-

Figure 5.
A diode tester using the PCF8591.

Listing 4. A diode tester using the PCF8591.

```
'-----------------------------------
'UNO_I2C4.BAS AD/DA PCF8591
'-----------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim N As Byte
Dim D As Byte
Dim U As Word
...
N = 0
Do
  I2cstart
  I2cwbyte 144            'write
  I2cwbyte 64            'DA enable
  I2cwbyte N
  Print N;
  Print " ";
  Locate 2 , 1
  Lcd N
  Lcd "    "
    I2cstart
    I2cwbyte 145            'read
    I2crbyte D , Nack      'Ain0
    I2cstop
    Print D
    Locate 2 , 5
    Lcd D
    Lcd "  "
    Waitms 100
    N = N + 1
    If D >= 50 Then Exit Do
  Loop
  Cls
  Locate 1 , 1
  Lcd " 1 mA"
  U = N - D
  U = U * 20
  Locate 2 , 2
  Lcd U
  Lcd " mV"
  End
```

resents the voltage on input Ai0. A reading of 50 indicates a voltage of 1 V, which corresponds to a diode current of 1 mA. At this point the ramp is stopped. Now the forward voltage of the diode can be calculated from the difference between the output voltage and the input voltage. It is displayed in millivolts. With a sample blue-green LED, the following results were displayed at the end of the measurement cycle:

1 mA
2920 mV

Future prospects

This is the last installment of our Microcontroller BootCamp series, but there will be other articles on Bascom applications for the Arduino Uno and the Elektor Extension shield from time to time. We hope we have aroused your enthusiasm for developing your own programs. If so, we encourage you to share the results of your efforts with other members of the Elektor community at www.

elektor-labs.com and http://forum.elektor.com (Microcontrollers section). Relatively small projects or applications still in the development stage are especially welcome.
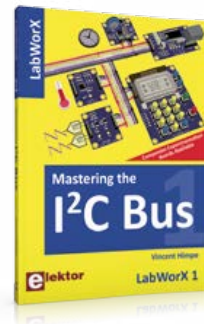
(140293-I)

Web Links

[1] www.elektor-magazine.com/140009

[2] www.elektor.com/tools/gnublin

[3] www.elektor-magazine.com/140293

[4] www.nxp.com/documents/data_sheet/
PCA9555.pdf

Further Reading

[a] *Mastering the I²C Bus,* Vincent Himpe, Elektor International Media Publishers, ISBN 978-0-905705-98-9.
Also available as an E-book.

DesignSpark Tips & Tricks
Day #15: Renaming Components

By **Neil Gruending**
(Canada)

Today we'll look at how to rename components in DesignSpark PCB.

Ever wonder how some boards have all of their reference designators in sequence so that they're easy to find? Well today we will learn how to renumber PCB components and then update the schematic with the changes using a process called backwards annotation.

Renaming Components

DesignSpark's component renaming tool works by dividing each side of the board into strips and then changing/renaming the reference designators within them as necessary. You configure the tool by using the physical directions Left to Right, Top to Bottom, Right to Left and Bottom to Top. Normally you would renumber components Left to Right and Top to Bottom as shown in the left side of **Figure 1** but you can use any combination you want. For example, the right side of the figure shows what combining Right to Left and Bottom to Top would look like.

In DesignSpark you specify the strip direction first and then then the direction to use within the strip. For our Left to Right and Top to Bottom
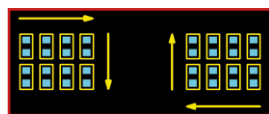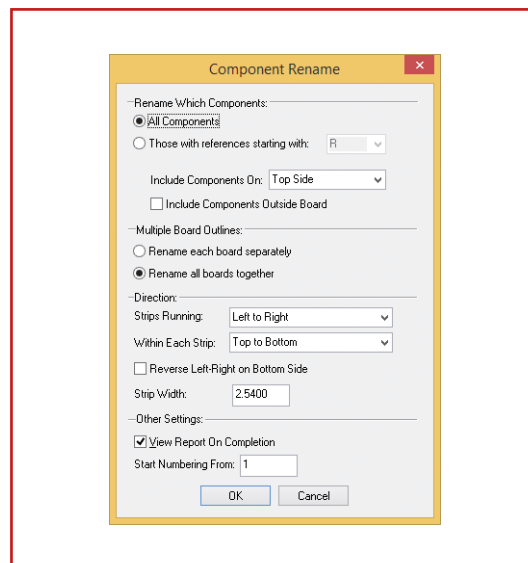
example, the strip direction would be set to Left to Right and the direction within the strip would be Top to Bottom. The only stipulation imposed by DesignSpark is that you can only combine a horizontal direction and a vertical direction which means that you can't use Top to Bottom and Bottom to Top for example.

The rename tool scans each strip in the desired direction looking for components. The first component will be renamed to the next available reference designator starting at the number one. If multiple components are aligned with the search direction then the tool will rename them using the within strip direction. In our example, the tool would search each strip from left to right and then top to bottom for vertically aligned components like in the left side of Figure 1.

Now let's take a look at the Component Rename window shown in **Figure 2** by going into the Tools menu and clicking Auto Rename Components. The Rename Which Components section lets you choose which components you want to rename. Normally you would choose All Components but you can also choose to rename components based on the reference designator and the side of the board that they're on. The Multiple Board Outlines section is useful if you have multiple boards in one design file so leave it at its default setting to rename the boards together.

We've already talked about how to set the renaming direction in the Direction section, but the strip width is a new parameter which tells the renaming tool how wide it should make each strip when dividing up the board. I also recommend enabling the Reverse Left-Right option to tell DesignSpark to reverse the left and right directions on the bottom side of the board. This is super handy because it's like flipping the board before doing the rename operation which is almost always what you want. And finally we have the Other Settings section where you can specify the number to start renumbering from.

Figure 3 shows an example where I took the board on the left and used the renaming tool to create the board on the right. I set the direction

Figure 1.
Renaming direction.

Figure 2. Component rename window.

to Left to Right and Top to Bottom. I deliberately set the strip width to be pretty narrow (5 mm) to show how a board with multiple strips would be renumbered. It's also worth noting that the transistor reference designator didn't change and that's because DesignSpark keeps a separate counter for each reference designator type (R, C, Q, etc).

Updating the Schematic

Now that we've renamed the PCB the way we want it's time to update the schematic with the new reference designators. The process is called backwards annotation and fortunately DesignSpark has a tool for that as well which you access in the Tools → Backwards Annotation menu. It will import all of the reference designator changes from the PCB file into the schematic so that everything will match up. It will not transfer over any other design changes though like net connectivity or component property changes. The annotation tool always assumes that there are one or more schematics associated with the PCB file. If you aren't using a project file then the tool assumes that you are using a single schematic file with the same name as the PCB. But if you do have a project set up then you can have multiple schematic files with different names. Unfortunately you can't change this behavior or manually select the schematic files to update.

Figure 4 shows what the backwards annotation windows looks like. Pressing the OK button will perform the schematic updates but you can see what would change first by clicking on View Renames. This can be useful if you verify the changes before the schematic is updated. The Delete Renames button is a little different because it won't update the schematic but will mark the updates being applied in the PCB file. You could use this if you've updated the schematic by hand for example. But you need to be careful using it because once the renames have been cleared you cannot reapply them later. The final option is the View Report on Completion check box which will generate a text file for you detailing all of the changes made. I recommend that you save all of your schematic files before clicking OK because once you accept the changes they can't be undone using the undo command. The left side of **Figure 5** shows our example before it was renamed and the right side is after. As you can see, the annotation tool updated all of
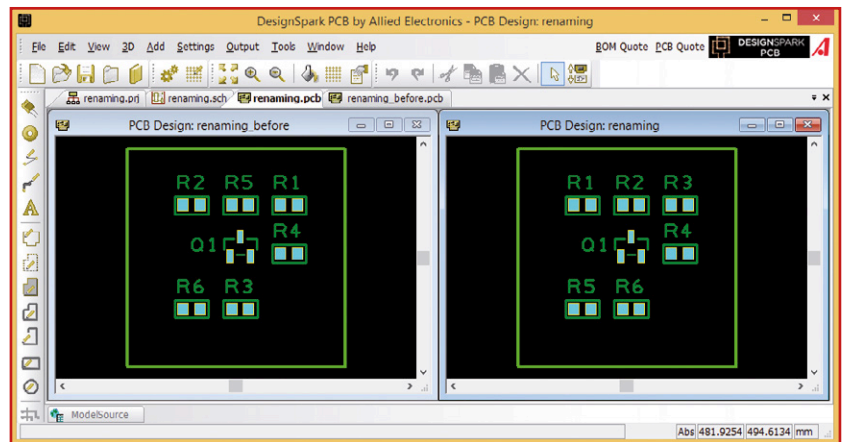
Figure 3.
Renamed components, before and after.
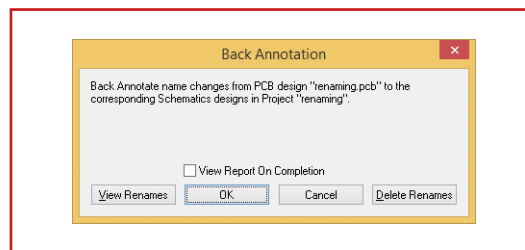
Figure 4.
Backwards annotation tool.

the reference designators with the PCB changes without making another change to the schematic.

Conclusion

Today we looked at the component renaming tool in DesignSpark. It's the only way to automatically rename components and it's a great way to make it easy to find those components on your next board. Next time we'll continue our focus on components and how they are used in schematics and PCB designs.

(140292)

Figure 5.
Renamed schematic example.

ELEKTOR-MAKER-SPACE

WHO:
YOU AND
ELEKTOR INTERNATIONAL MEDIA

WHAT:
ELEKTOR LAB's MAKER SPACE

WHERE:
ELECTRONICA, MUNICH,
HALL A6 / BOOTH 380

WHEN:
NOVEMBER 11–14, 2014

Make It in Munich

meet the Elektor Makers
at the 2014 *electronica* show

By **Wisse Hettinga** (Director, Elektor Labs)

What's your expectation when attending a massive show like *electronica* in Munich: immaculately trimmed booths and pavilions, sharp dressed people with attractively arranged products behind glass, all in wide, exquisitely cleaned halls with waiters in the restaurants? Don't get me wrong, I'm not against dressing for the occasion—I do it every day and I can recommend it to everyone, but now and then you need to dress down a little—relax, go easy, hang out, chill, do some real work and get your hands dirty. That kind of summarizes what Elektor Labs will roll out at the show: real work in our own little corner of the show, a *space*, not a booth.

We call it the **Elektor Labs Maker Space** and we are dressing down on purpose. We did our best to make it an easy and relaxing experience. First and most important are the big tables. That's where everything starts, from a simple "hi, have a seat" to "fancy a coffee?" to "here's power to charge your phone" right up to "feel free to use the equipment and do some soldering or measuring". The big tables are also the center of mini workshops you can attend; they're hands-on meaning you can learn new technology right there and then. Free. The Elektor Labs team is there and you can chat and rag chew your favorite project or get their tech advice.

So, what more to explore at the Elektor Labs space?

On our tables are all sorts of equipment to work with, perfect your soldering skills with Conrad's soldering irons, catch that elusive signal on the Hameg oscilloscope, or get clever with the National Instruments Virtual Bench measurement equipment There will be T-Boards all over the place and lots of other embedded boards to play with. We will bring some Elektor Preferred Parts (ELPP) boxes for you to take parts out and do some real-life breadboarding.

Every day we will have a series of mini workshops. Some we do ourselves and they should be fun simply because we are not slick presenters. Others are done by companies we like (and *vice versa*) or simply around a subject we think is interesting.

We are joined by Editor Jan Buiting normally seen carrying Retronics stuff around and an expert in connecting old and new technologies. Jan will bring vintage equipment to the space and do a few short talks on it. There's also Luc Lemmens demoing Elektor's new T-Boards.

Wednesday afternoon @16:00 CET we will have a Labs Q&A Webinar live from the Munich space brought to you by presenters Jan & Jaime. Join them at the space or online, learn from the questions asked by others—or better, participate!

For more than 2,000 visitors there will be Goody Bags stuffed with magazines and other material—leave your name, email address or business card and receive a bag.

So, if you're tired of the walking & talking, your tie needs straightening, your batteries are flat, you need free Wi-Fi or just a straightforward 'Clooney coffee'… if you crave to get hands-on with electronics again, or simply want the Elektor Goody Bag... on behalf of the Elektor Team; **WELCOME!**

(140327)

Precise Nixie Clock
With a GPS receiver for the exact time

By **Willem Tak**
(Netherlands)

The warm glow of the numbers in Nixie tubes always gives a special touch to the equipment in which they are used. A special feature of the Nixie clock described here is that it combines legacy technology with new technology. The tubes show the time in an elegant manner, while the advanced GPS module with integrated antenna ensures that the displayed time is always correct.

Many people find Nixie clocks fascinating. The warm glow of the tubes and their unique design are very attractive. In this project we present a four-digit Nixie clock (with hours and minutes) that receives time data from a GPS module. That's a change from the usual radio time signal receiver (WWV, MSF, DCF77), and certainly just as accurate. This means that you never have to set or adjust the clock, and it can be built into a compact enclosure with only the tubes protruding.

For the GPS receiver, the author used a module made by GlobalSat (type EM-406 or EM-411). For the sake of better availability, Elektor Labs chose the Maestro A2035H module instead, which is also used on the FPGA extension board. Although the GlobalSat modules can also be used without any firmware changes, they require some manual modifications to the PCB. A nice feature of both types is that they provide satisfactory reception indoors under virtually all conditions.

The hardware
The key components of this circuit are of course the Nixie tubes: V1 to V4 on the schematic diagram in **Figure 1**. They have already been used

in various Elektor projects (including [1] and [2]), so there is no need to discuss them in more detail here. Suffice it to say that they are cold-cathode tubes with a separate anode in the tube for each digit. If a sufficiently high voltage is applied to one of the anodes (in this case 180 V), the selected digit is surrounded

by a neon glow. Here the required high voltage is generated by a small circuit built around a type MC34063 step-up converter (IC7). Using a fast diode (D1) and an inductor (L1), the input voltage is converted into a high voltage by the flyback effect and used to charge a high-voltage capacitor (C10). After assembling the board,

Figure 1.
Schematic of the Nixie clock with four-digit display and integrated GPS receiver.
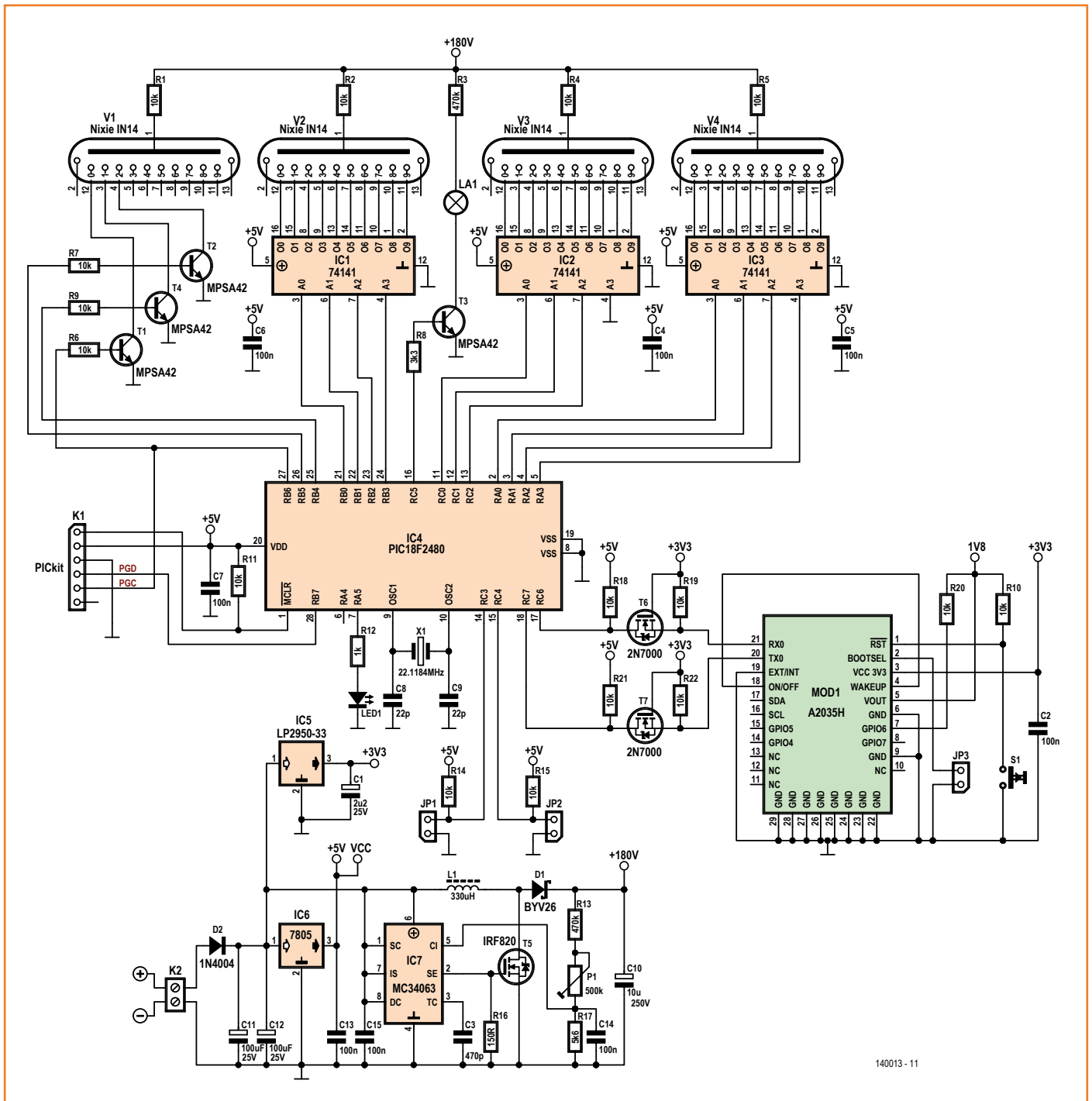
140013 - 11

check carefully to ensure that the feedback path to pin 5 of the IC is intact. If it is missing or open-circuited, the supply voltage can rise to a very high level, possibly resulting in an exploded capacitor. The 180 V supply voltage is fed to the anodes of the Nixie tubes via resistors R1, R2, R4 and R5. There are many different types of Nixie tube available, so you need to find out the rated current for the tubes you intend to use before you build the circuit. That usually results in a resistor value from 10 to 15 kΩ; in the circuit described here we chose 10 kΩ. In the Elektor prototype we used type IN-16 Nixie tubes. These Russian-made tubes are affordable and fairly easy to obtain.

As usual, for the Nixie driver ICs we chose the type 74141 ICs specifically designed for this task, but they are very hard to find. A possible alternative is the Russian K155ID1. A separate IC is not used to drive the Nixie tube for the tens of hours position (V1, the leftmost tube as seen from the front), but instead three transistors for the 0, 1 and 2 digits in the tube. A neon lamp (also driven by a transistor) is located between the hours and minutes sections. When the clock is operating properly, this lamp blinks once per second or is continuously lit (configurable with JP2).

A PIC18F2480 microcontroller (IC4) is used to read the data from the GPS module and generate the control signals for the Nixie drivers and the transistors. It runs at a clock frequency of 22.1184 MHz to allow the frequency necessary for the 4,800 baud output of the GPS module (MOD1) to be obtained by dividing down. The microcontroller also drives an (optional) diagnostic LED (LED1) that blinks briefly after a reset and subsequently blinks each time a correct GPS string is received. If this LED is continuously on, there is a problem with the circuit. The microcontroller can be programmed in-circuit using a PICkit 2 programmer connected to K1.

There are two jumpers for the microcontroller. JP1 determines whether or not a leading zero is displayed (JP1 open = no leading zero; JP1 closed = leading zero), while JP2 selects between a blinking and non-blinking hours/minutes separator (JP2 open = not blinking; JP1 closed = blinking). The GPS module (A2035H) contains all the circuitry necessary to receive GPS signals, along with an antenna. It's just about self-contained. Components S1 and JP3 on the schematic may be omitted from the board; they are only necessary if the firmware of the A2035 has to be upgraded. Since the GPS module works at 3.3 V and the microcontroller works at 5 V, level converters consisting of FETs T6 and T7 along with resistor pairs R18/R19 and R21/R22 are included in the data lines between these two components. A pair of voltage regulator ICs supply power to the "low-voltage" circuitry: a 7805 (IC6) supplies 5 V for the microcontroller and the driver ICs, while an LP2950-33 supplies 3.3 V for the GPS module. The entire circuit can be powered from an AC adapter with an output voltage of 9 to 15 V DC.

The software

The source code and hex code of the software for the clock can be downloaded free of charge from the Elektor website, and a preprogrammed microcontroller (140013-41) is available from the Elektor Shop [3].

The microcontroller software is designed to be able to control clocks with seconds display (six digits) as well. Although the present design has only four digits, the entire software is described here. GPS data acquisition is implemented in the usual way. The GPS module supplies the GPRMC GPS string. The initialization routine configures the module to send only this string at one-second intervals. All other GPS strings are disabled.

The microcontroller checks for the availability of new asynchronous data from the GPS module in a routine that polls the RS232 buffer to see whether a character is present. The internal

watchdog monitors the RS232 data stream, and if it is not established within approximately 1 second after startup the microcontroller is reset. The diagnostic LED is lit constantly during this period. After a valid string has been received, the RS232 line is still monitored but in a different manner. Here again a continuously lit diagnostic LED indicates a problem.

When data is received, it is first checked to verify that it is a GPRMC string (which should always be the case), and if it is then the routine waits for the entire string to be received, meanwhile writing the characters to memory until the CR character is received. Then the checksum of the received string is calculated and compared to the checksum present in the string. If they match, the string is considered to be valid.

After this the routine calculates the offset (1 or 2 hours) that must be added to the received time to arrive at the correct summer or winter time, since GPS does not have a flag for this. The algorithm for this determines the current hour of the year and then uses tables to see whether the calculated hour is in the summer time period or the winter time period. The tables extend as far as 2020.

Finally, the time received from the GPS module in ASCII format is converted to hex format with separate variables for hours, minutes and seconds. An ASCII to BCD conversion is also performed to keep the code compatible with previous versions of the author's software.

In theory we now have a valid GPS time, but there is still a little problem if we want to display this time with six digits (i.e. with seconds). The GPS strings are dependent on the reception conditions, which are better in some places than in others. Although the module has a lot of embedded intelligence and high sensitivity, strings can sometimes get lost or corrupted. That will rarely be noticeable if only hours and minutes are shown (as in the present design), but if seconds are shown the display will hiccup when a string is lost. Then the seconds display will skip over one or two seconds, or even worse. To remedy this problem, we decided to not show the GPS string directly on the clock, but instead to implement a real-time clock using a separate timer. For this we made an interrupt loop that is called at exactly 50-ms intervals. This must be carefully adjusted according to the actual microcontroller clock frequency, since only a small deviation is allowed.

The software includes variables such as PIC_HR and GPS_HR, which contain the hour generated by the PIC microcontroller and the hour from the GPS string. The 50 ms pulse signal can be seen on pin 6 (RA4) of the microcontroller.

The internal clock starts at a time of 00:00(:00) After startup, the software therefore waits for that time to be received, which may take a good while. Once a matching valid GPS string is received, the PIC time (microcontroller time) is synchronized and the clock can start running, controlled by the interrupt loop.

A dual checking mechanism launched at this point. To make this as iron-clad as possible, GPS reception is constantly monitored for stability.

Each GPS string is analyzed, and if it is found to be correct it is stored in a buffer with room for ten time strings. Each time string consists of three bytes containing the binary representation of the number of elapsed seconds since midnight. This has a maximum value of 24 x 60 x 60, which is unfortunately greater than 65,536.

When a new time is received the entire buffer shifts up by one position; the oldest time in the first position is deleted and the newest time is placed in the tenth position. If the GPS data stream is perfectly stable, the difference between the oldest and newest times will always be exactly nine seconds. When this is true, the flag GPS_STABLE is set. We can therefore assume that the GPS signal is valid and can be used to synchronize the PIC time.

This synchronization occurs at least once per hour. When the PIC time is xx:00:30, the interrupt routine issues a synchronization request. If the GPS signal is stable, the PIC time is set equal to the GPS time. The synchronization time of xx:00:30 was chosen so that any corrections that may occur are only visible in the seconds, not in the minutes or hours.

In any case, in the various clocks that the author has built the observed deviation has never exceeded 1 second if the interrupt loop is adjusted to exactly 50 ms.
Unfortunately, the time synchronization is not

always adequate. During the start-up phase, it is fairly common for the GPS module to stubbornly output an incorrect time. A valid string, usually without complete coordinate data but with time data totally different from the correct time, may be output after startup, and the module may persist in this behavior for a fairly long time. This is naturally very irritating because the GPS time is improperly used to set the PIC time, and in the worst case it may take nearly an hour before it is adjusted.

To prevent this and at the same time eliminate very imprecise interrupt loops, a second checking

Component List

Resistors
R1,R2,R4,R5,R6,R7,R9,R10,R11,R14,R15,R18,R19, R20,R21,R22 = 10kΩ
R3,R13 = 470kΩ
R8 = 3.3kΩ
R12 = 1kΩ
R16 = 150Ω
R17 = 5.6kΩ
P1 = 500kΩ trimpot, horizontal

Capacitors
C1 = 2.2µF 25V, 2mm pitch
C2,C4,C5,C6,C7,C13,C14, C15 = 100nF
C3 = 470pF, Y5P, 0.1" pitch
C8,C9 = 22pF, C0G/NP0, 0.1" pitch
C10 = 10µF 250V, radial, 5mm pitch (Panasonic ECA2EHG100)
C11,C12 = 100µF 25V, radial, 3.5mm pitch

Inductor
L1 = 330µH 900mA, radial (10mm diam., 15mm height)

Semiconductors
D1 = BYV26 (ultrafast diode, 600 V/1 A)
D2 = 1N4007
LED1 = LED, red, 3mm
T1,T2,T3,T4 = MPSA42 (300V/ 500mA)
T5 = IRF820 (N-MOSFET, 500V/2.5A)
T6,T7 = 2N7000 (N-MOSFET, 60V/200mA)
IC1,IC2,IC3 = K155ID1 (74141)
IC4 = PIC18F2480-I/SP (programmed, # 140013-41)
IC5 = LP2950-33LPE3
IC6 = MC7805
IC7 = MC34063

Miscellaneous
V1,V2,V3,V4 = IN-14 nixie tube
LA1 = wired neon lamp
X1 = 22.1184 MHz quartz crystal
MOD1 = A2035H GPS module met internal antenna (Maestro Wireless Solutions)
JP1,JP2,JP3 = 2-pin pinheader, 0.1" with jumper (JP3 optional)
K1 = 6-pin pinheader, 0.1" pitch
K2 = 2-way PCB screw terminal block, 0.2" pitch
S1 = pushbutton, PCB mount, 6x6 mm (optional)
PCB # 140013-1, see [1]
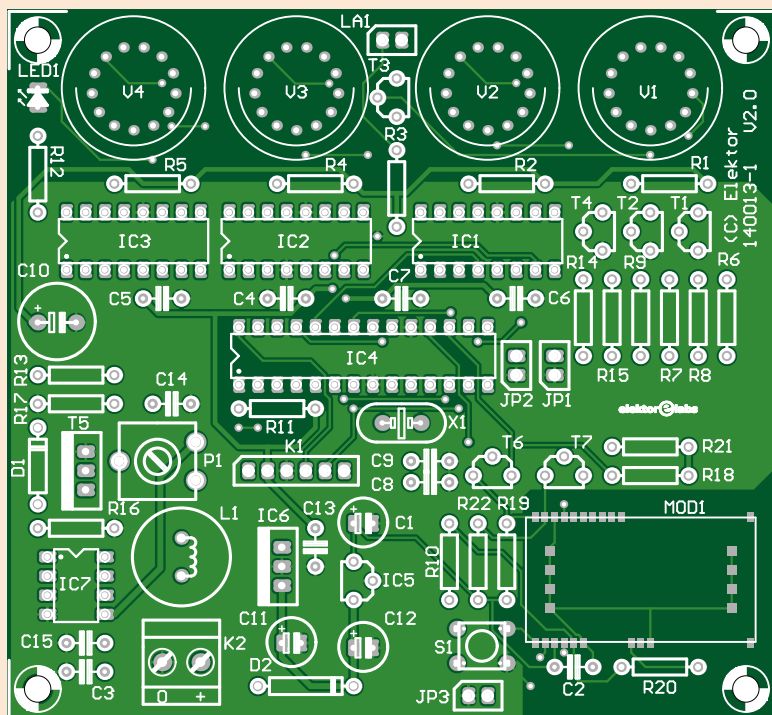
Figure 2.
All of the clock components, including the GPS receiver, are mounted on the PCB. Simply connect an external AC adapter for power and you're ready to go.

mechanism has been implemented. The elapsed seconds are also calculated from the PIC time every second, as a three-byte value. As long as a stable GPS signal is present or as soon as a stable signal becomes available, the difference between the GPS seconds and the PIC seconds is calculated. If the difference is greater than three seconds (a figure simply pulled out of the hat), the PIC time is resynchronized to the GPS time. Although this method may appear complicated (and it actually is), practical experience shows that it works well for extended periods. The neon lamp between the hours and the minutes indicates the reliability of the signal. If it blinks or is constantly lit (depending on the JP2 setting), everything is okay, but if it is dark the clock is running entirely on its internal signal because no valid GPS string has been received for a good while (approximately 30 seconds), and it may not be entirely correct.

Construction

Figure 2 shows the PCB layout designed for the Nixie clock. The PCB can be ordered from the Elektor Shop [3], and of course you can download the layout from the Elektor website free of charge. Everything except the external power supply (AC adapter) is mounted on the PCB. Fitting the components is not difficult, since they are all leaded components with the exception of the GPS module. It has small solder pads that must be soldered to the pads on the PCB. With a bit of patience and a fine-tip solder iron, that is not overly difficult. There are also several ground pads on the bottom of the module. The only way to get them properly soldered is to use a reflow oven. Because most of our members do not have a reflow oven available, we left the ground pads unsoldered on our prototype, and as far as we could see the module still worked fine.

You should start by fitting the low-profile components and gradually work up to the high-profile components. That's always the most convenient method. The voltage regulator ICs do not need heat sinks. Be careful when handling the Nixie tubes—they are fairly fragile and the lead wires are quite thin. Bases are also available for these tubes; they help to keep the tubes stable on the PCB. We usually recommend that you trim the lead wires stepwise (but not too short) before inserting them into the holes in the board, since that makes insertion easier. First solder one lead

in place, then align the tube precisely perpendicular to the board, and finally solder the other leads in place.

Once you're done, connect an AC adapter (for example, 12 V at 1 A) and wait until the GPS module has good reception. Now you can start enjoying your elegant and precise time display. It's also advisable to fit the PCB neatly in a suitable enclosure with only the Nixie tubes protruding. That way you avoid the risk that someone involuntarily comes in contact with the 180 V supply voltage.

(140013-I)
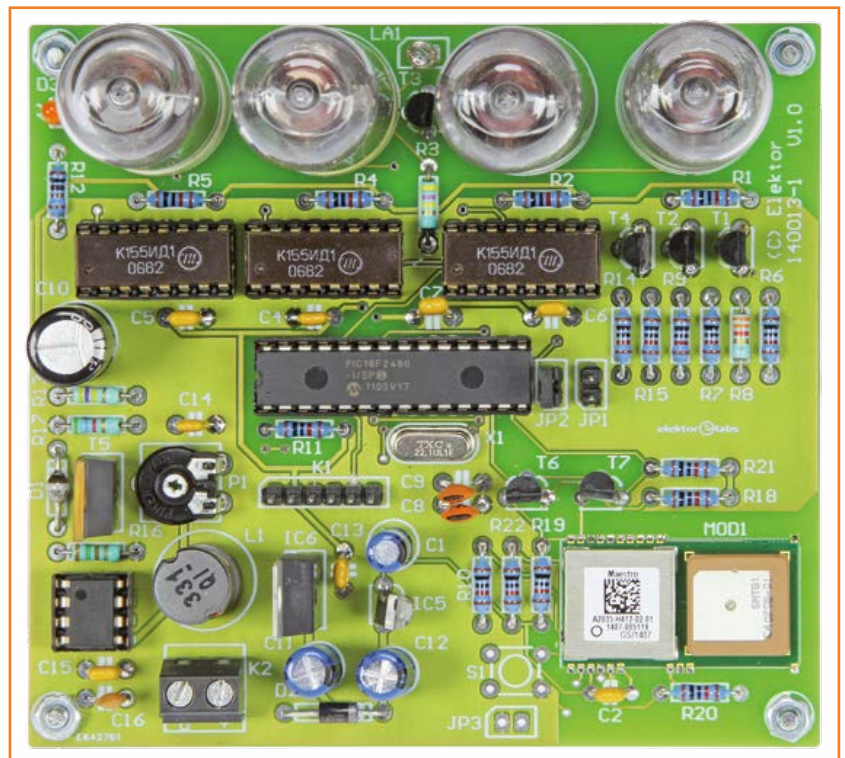
Web Links

[1] Sputnik Time Machine: www.elektor-magazine.com/050018

[2] Nixie Thermometer/Hygrometer: www.elektor-magazine.com/110321

[3] www.elektor-magazine.com/140013

C Modules

Software for Elektor Extension Shield, Relay Board and more

By **Jens Nickel** (Elektor Germany)

ExtensionEFL_ECC_RS485.c

BoardEFL_ArduinoUnoCore.c

ExtensionEFL_Arduino_
ElektorExtensionShield.c

ExtensionEFL_
EEC_Relay8.c

Modular hardware enables us to create plug-and-play prototypes rapidly; all we need do is assemble predesigned building blocks. Exactly the same principle can be applied to software, and the C programming language is well suited to this approach. This article showcases some compatible software modules and demo applications for our Elektor Extension Shield and three expansion boards.

Elektor's 2014 Project Generator double edition introduced a compact plug-in board for the Arduino Uno that contained a Display, two user-definable LEDs, two pushbuttons and a potentiometer or 'pot' [1]. Using this board, newcomers can get cracking immediately and make their first steps in programming microcontrollers, for example adding digital outputs (LEDs) and digital inputs (pushbuttons). More advanced users will make use of two additional expansion connectors on the board. The 10-pin Embedded Communication Connector (ECC) provides TX/RX UART signals and two GPIO pins. Using a flatcable you can hook up an RS-485 module for example, enabling you to send and receive bytes down longer c cables. Already developed is an NFC gateway, enabling simple ASCII commands to read and characterize NFC cards or communicate with NFC-ready smartphones. Further ECC modules are planned as well.

The 14-pin Embedded Extension Connector (EEC) is also known as a Gnublin connector, as it enables you to link up with the Gnublin modules from Benedikt Sauter and his team. Benedikt has also defined the specifics of the connector,
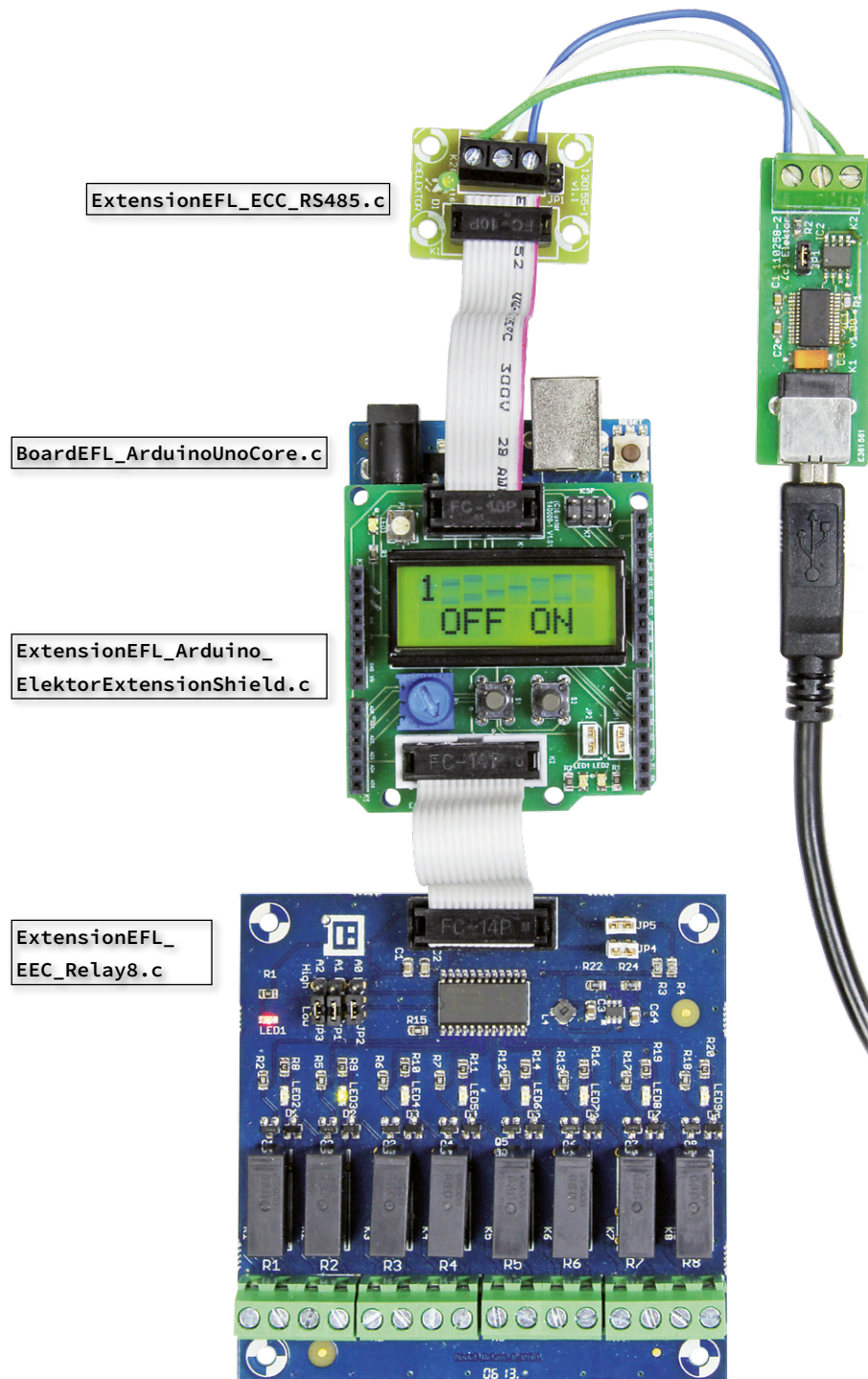
although we should point out that the Elektor Extension Shield employs only four of the 14 pins (namely 3.3 V, GND and the two I²C lines). However, most Gnublin boards [2] use only the I²C pins, for example the Port Expander Board, the Temperature Sensor Board, a board with eight relays [3] and the ADC Board from the September 2014 edition [4].

Software modules

In the BASCOM-oriented Microcontroller Boot-Camp course that concludes in this edition veteran author Burkhard Kainka has developed some interesting applications for the Elektor Extension Shield. More of these will appear in Elektor from time to time.

One thing that was lacking till now was software support for C. This language is particularly well suited for modular software projects. Once developed and tested (either by yourself or by third parties), software modules can be used again and again in your homebrew projects. This shortens development time enormously and in extreme cases you can achieve a major, functioning application in minutes. And should those project requirements alter, for instance if an additional RS-485 interface becomes necessary, you can create a compatible software solution just as rapidly.

Hardware-wise we can now devise a functioning system from modules that need merely plugging together. Once you have stacked the Elektor Extension Shield onto an Arduino Uno board, you can plug in your choice out of an RS-485 module, a relay card or an ADC board. We shall now apply the same modular principle to the software; for an application with defined requirements we need simply assemble the corresponding code modules into a software project. Ideally each hardware module will have its associated software module that can be used without alteration, regardless of in which particular set-up the hardware interacts. Take for example the code for the Relay Expansion Board; it remains unchanged regardless of whether we hook up the relays to the Elektor Extension Shield, the Xmega Board or the Elektor Linux Board.

Pretty close to perfection

Using the programming language C and the Embedded Firmware Library (EFL) described in

New file structure

The EFL is now adapted additionally for larger modular projects, in which several expansion boards are connected to (or chained in series with) a Controller board. For this purpose the Library needs to be extended somewhat and restructured compared with its previous state [5].

- The hierarchical system of the Includes has been disentangled. Only Header files from the `Common` folder are now integrated throughout. These files are identical in all EFL projects, regardless of which boards and Controllers are used.
- To this end a new Header file is placed in the `Common` folder (`ControllerDefinesEFL.h`). It now includes all Function definitions of the Controller API, which of course are always identical. The Controller-specific Header file (previously `ControllerEFL.h`) now includes only Controller-specific definitions, such as those for the Register.
- Instead of `ControllerEFL.h/.c` and/or `BoardEFL.h/.c` we now provide meaningful names for the Controller and board-specific code files. The file that implements the Controller API for the ATmega328, is now called `ControllerEFL_ATmega328`. The file extracted from the wiring connection to the Arduino Uno board is called `BoardEFL_ArduinoUnoCore`.
- We can now integrate several extension boards into our project, since the relevant code files all have differing names instead of `ExtensionEFL.c/.h`. For instance:
  - ExtensionEFL_Arduino_ElektorExtensionShield.c/.h
  - ExtensionEFL_ECC_RS-485.c/.h
  - ExtensionEFL_EEC_Relay8.c/.h

  The type of expansion connector is always shown between the underscore symbols.
- The `Board-Init` Functions that incorporate the onboard wiring connection in the Tables and make ready the Peripheral units are also given more specific names, for example `ExtensionEFL_Arduino_ ElektorExtensionShield_Init(0)` in place of `ExtensionEFL_ Init()`. To make this consistent, the `Controller-Init` Function is now, for example, `ControllerEFL_ATmega328_Init()` instead of `ControllerEFL_Init()`.

  An initialization of the Controllers and all boards could now look like this for example:
  - ControllerEFL_ATmega328_Init();
  - BoardEFL_ArduinoUnoCore_Init();
  - ExtensionEFL_Arduino_ElektorExtensionShield_Init(0);
  - ExtensionEFL_ECC_RS-485_Init(2);
  - ExtensionEFL_EEC_Relay8_Init(3);
- The Init Functions of the Expansion boards are now augmented with the consecutive Block number of the expansion connector to which they are attached. In this way the wiring connection (from the Controller pins through to the furthest Peripheral) is represented correctly in the EFL-internal Tables. It is now also possible to 'chain' Expansion boards in series as in the present case (for this see also the main article and Figure 1).
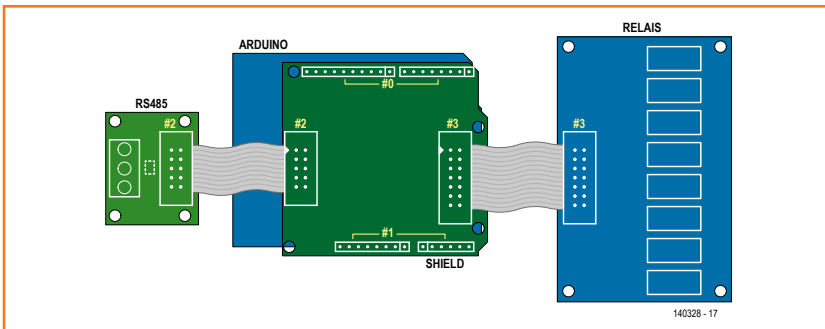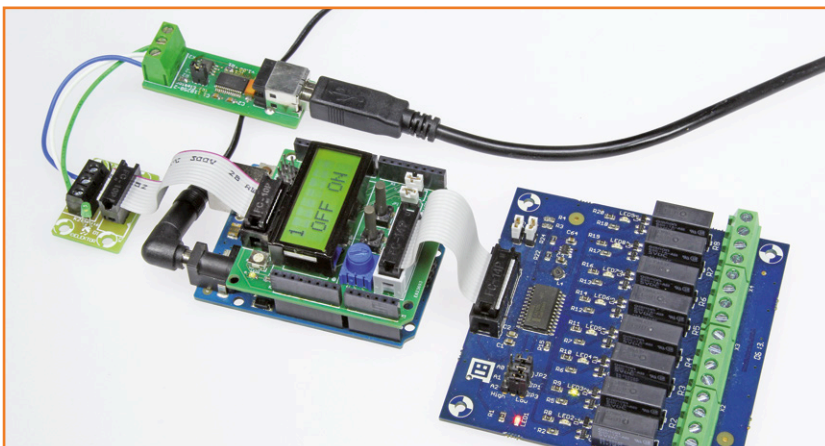
Figure 1.
Chaining expansion boards: the Extension Shield is plugged into two connector strips on the Arduino Uno (#0 = digital pins, #1 = analog pins). Connectors #2 (ECC) and #3 (EEC) are still available for use, being linked through and repeated on the Shield.

Figure 2.
Application 1: Eight relays can be operated either locally via a user interface or remotely via RS-485.

Elektor [5] we can get extremely close to the ideal just described, as we shall now show. Up till now you could use the EFL only for a Duo equipped with a Controller Board and an Extension Board. So that other expansion boards can be used in a project, the Library had to be extended and restructured somewhat. To avoid boring the hands-on practitioners among our readers with all the details, we have placed the full explanation in a separate text box. Here in brief are just the most important alterations:

● Previously a file pair by the name of `ControllerEFL.h/.c` contained Controller-specific source code that made standardized Functions available for activating the inputs and outputs of each Controller, for example `IO_SetPinLevel(…)`. A file pair, always called `BoardEFL.h/.c`, displayed the onboard wiring connections (which the application developer no longer needs to know). Instead of the uniform designations these files now have meaningful names. The file responsible for the ATmega328 is now called `ControllerEFL_ATmega328`. The file belong-

ing to the Arduino-Uno board has the name `BoardEFL_ArduinoUnoCore`.

● We can now use several extension boards in our project, since each of the corresponding code files has a different name, for example:

```
ExtensionEFL_Arduino_
ElektorExtensionShield.c/.h
ExtensionEFL_ECC_RS-485.c/.h
ExtensionEFL_EEC_Relay8.c/.h
```

● The Init Functions called up initially are also named specifically. This is the initialization of the Controller and all the boards for the first sample application discussed next:

```
ControllerEFL_ATmega328_Init();
BoardEFL_ArduinoUnoCore_Init();
ExtensionEFL_Arduino_
ElektorExtensionShield_Init(0);
ExtensionEFL_ECC_RS-485_Init(2);
ExtensionEFL_EEC_Relay8_Init(3);
```

The number of the first expansion Port connected is given together with the Init Functions of the expansion board; this number is incremented during the initialization of the board (**Figure 1**). The Extension Shield is installed on the Arduino Uno, which defines two expansion Ports; these are assigned the numbers #0 (digital pins) and #1 (analog pins). The Extension Shield is connected to #0 and #1 and has onboard two additional expansion Ports, #2 (ECC) and #3 (EEC). To the first of these we connect the RS-485 module and to the second the relay module.

(Remote) Control of relays
We have prepared three demo projects for Atmel Studio 6 that you can find in the download data for this article [6]. Incidentally a Configurator for the PC is already in development; this automatically assembles and integrates the files necessary for an EFL project in Atmel Studio. This leaves you only the task of indicating which boards you plan to use in the project.

Let's begin with the first project, a small control application. Eight relays can be switched locally using a user interface; they can also be controlled remotely by a PC using a Terminal program. An Elektor Extension Shield is plugged on top of an

Arduino Uno; the Gnublin PCB with eight relays is connected to this with some flat cable (**Figure 2**). The RS-485 ECC module is an optional extra, also connected to the Shield; the connection then goes over RS-485 (two wires plus ground) to an RS-485-to-USB converter and the PC. The application will also work if you link the Arduino Uno direct to a PC using USB.

The software can be found in the download package in the `ElektorShieldRelay` folder. After clicking on `ElektorShieldRelay.atsln` the project opens in Atmel Studio (**Figure 3**). On the right, in Solution Explorer, you can see the integrated files. In all cases you should be able to see a folder called `Hardware`. As well as the files mentioned above for the Controller, the Controller board and the expansion board you will also find the new BlockEFL files with the Low Level Functions for the RS-485 driver, the digital outputs and inputs and the Display (see box-out 'BlockEFL files'). Advanced users should check out the code in the file `ExtensionEFL_Arduino_ElektorEx-tensionShield.c` (Figure 3). This code records the peripheral Blocks located on the Shield in the EFL-internal Block Table (**Figure 4**). From now on the buttons on the board respond to the Block number #0 and the LEDs to Block number #1, as there is already an LED mounted directly on the Arduino board (addressed using Block number #0). For the pot an ADC Block #0 is provided, then comes the Display with the number #0. Finally the wiring to the ECC and EEC connectors is displayed, with two new Blocks provided for the connectors (#2, #3). The entries are then used once more by the code of the ECC and EEC modules. Last of all, check out each peripheral unit in the Table to see which Controller pin it is connected to.

Let's now examine the main program in the file `ElektorShieldRelay.c`. The routine `ApplicationSetup()` contains all the initializations from the Controller via the boards as far as the Libraries. Assuming your hardware setup does not change, you can leave the whole bunch untouched.

Short code
The application itself can be viewed in **Listing 1**. In the Function `ApplicationLoop()` you will see all the commands that need to be reiterated each time. First the pressbuttons are polled, then con-

verted into the character strings in Block Protocol format [7] that are received via the UART. The Function `ADCSimple_GetRawValue(0, 0)` returns the value of the ADC input 0 in ADC Block #0 (the pot is connected here). The value can amount to 0…1023 (10 bits); we can reduce this to 7 bits by shifting them to the right, in order to obtain a figure between 0 and 7. This enables us to select one of the eight relays from the setting of the pot. The selected value is shown in line 0 of the Display.

The Function `ButtonEventCallback(…)` contains code that is carried out when a pushbutton is pressed. Which of the two buttons was pressed is determined using the variable `ButtonPosition` (0 or 1). If the left-hand button (0) is pressed, we deactivate the selected relay, whilst pressing the right-hand button (1) activates the relay. As
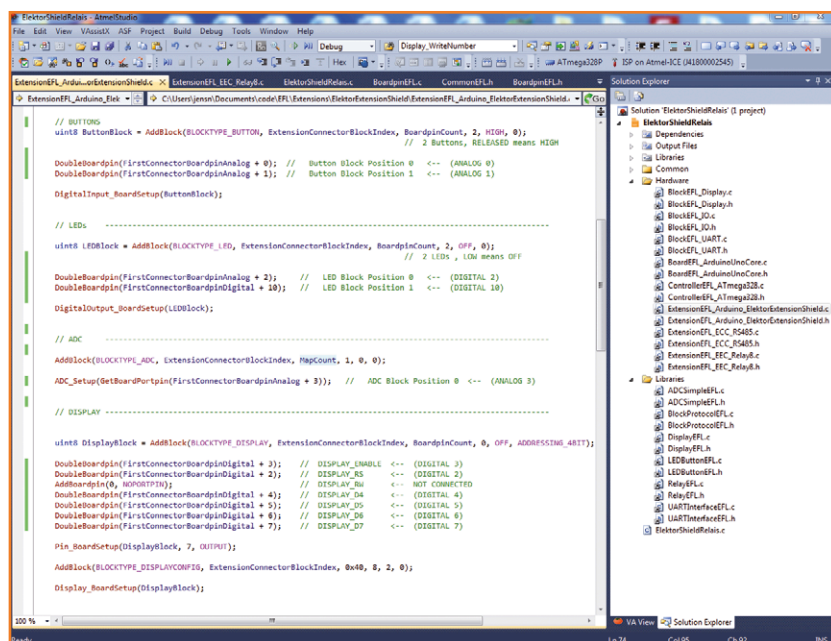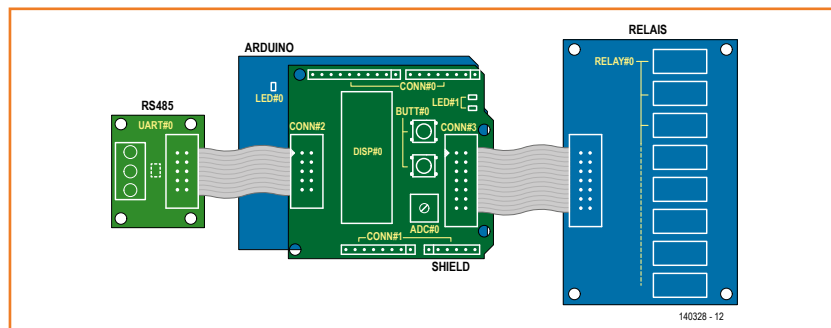
Figure 3.
Relay control operation in Atmel Studio 6. The code indicates the initialization of the Extension Shield.

Figure 4.
All peripherals can be addressed conveniently and uniformly using their Block numbers.

BlockEFL files

Up till now BoardEFL and ExtensionEFL files have still included the so-called Low Level/Block Functions, such as for example the Function

```
void Display_SendByte(uint8 DisplayBlockIndex, uint8 ByteToSend, uint8 DATABYTE_COMMANDBYTE)
```

which is based on the wiring interconnection between the Controller and Display (4-bit or SPI as appropriate). A Display Library such as `DisplayEFL` then needs only to ensure that the correct bytes are sent to the Display. The code there is independent of which particular route the bytes take to reach the Display. This Block Function must be defined once in the project if a Display is provided on any of the boards. Logically we therefore provide a dedicated file pair in the `Hardware` folder for the Low Level/Block Functions required by the Display. If a Display is used anywhere in a project, you need to integrate `BlockEFL_Display.h/.c` in addition. A different BlockEFL file is responsible for the RS-485 Functions. Block Functions for digital outputs and inputs, such as for instance `SwitchDigitalOutput(uint8 BlockIndex, uint8 Position, uint8 ON_OFF)` are also extracted from the BoardEFL file and relocated into a new file `BlockEFL_IO`.

The Board files now contain normally only the Init Functions that define wiring and integrated Peripheral/Blocks like, for example, making the Display ready. For this purpose the Function

```
void Display_BoardSetup(uint8 DisplayBlockIndex);
```

is called up there, now located in `BlockEFL_Display.h/.c`.

Overall, you now have to deal with a greater number of files but the modularity of the EFL has been further improved. Already in development is a Project Configurator, which assembles and integrates the files necessary for a project automatically.

140328 - 14
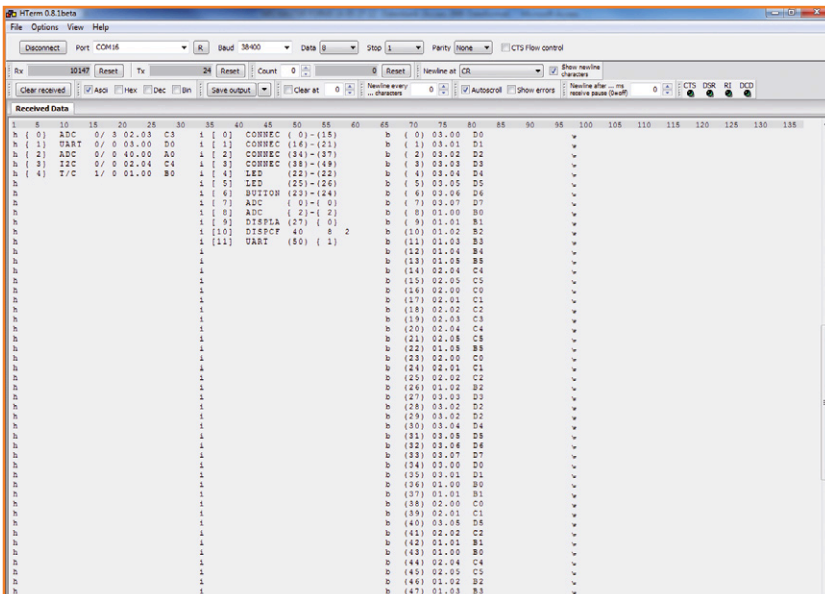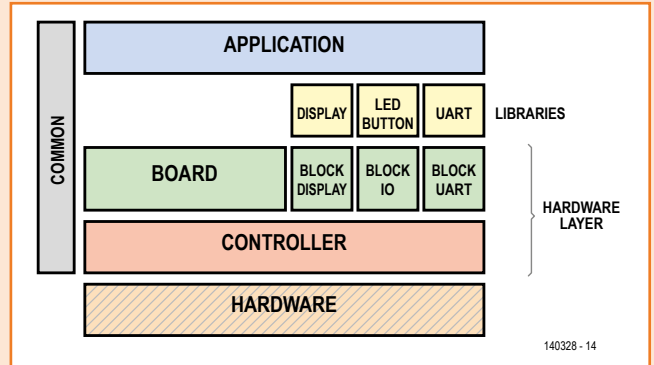
the Function `SwitchRelay(…)` directly demands a 0 = OFF or a 1 = ON as its third parameter, the code turns out agreeably short and to the point. Info on these and the other EFL Functions is in the Doxygen documentation (click on `Index.htm` in the download).

Using the Block Protocol [7] you can control the relays remotely from a Terminal program. The command `R 0 2 + <CR>` activates the third relay in Relay Block #0 (as always we count up from zero, 0, 1, 2…). `R 0 2 – <CR>` deactivates it again. With `L 0 0 + <CR>` we can switch on

Figure 5.
You can display the EFL Tables using a Terminal program. The Blocks are set out centrally.

the LED located directly on the Arduino. `L 1 0 + <CR>` and `L 1 1 + <CR>` are the corresponding commands for the LEDs on the Extension Shield. Do also try the command `x <CR>`; you will then have on your screen the content of the EFL Table. At the centre you will see all the relevant Blocks (**Figure 5**).

Precise measurement

For our second application we remove the relay board and substitute the 16-bit ADC board that we featured in the September issue [4]. As mentioned in that article, we link the output A3 on the lower Arduino connector strip (replicated on the Shield) using a flying lead to the input `AIN0` of the ADC board (**Figure 6**). Our task now is to digitize the setting of the pot also using the accurate external ADC.

Double clicking on `ElektorShieldADC.atsln` reveals the source code. As seen in Solution Explorer, instead of the two files `ExtensionEFL_EEC_Relay8.c/.h` we now have two files `ExtensionEFL_EEC_ADCModule16bit.c/.h` in the project. In addition the files `ADC_ADS1x1xEFL.c` and `BlockEFL_DeviceRegister16.c` have appeared now. `BlockEFL_DeviceRegister16.c` in the `Hardware` folder contains Low Level Functions for addressing an I2C chip, featuring 16-bit registers. `ADC_ADS1x1xEFL.c` takes care of the correct compilation of the bytes, which are written in these registers (the Library of my colleague Clemens Valens is adapted to the EFL here [4]).

Application developers don't have to be concerned with all this. The only thing they need to know is that after initialization of the boards a further ADC Block with the number #1 is created (**Figure 7**). You can now access the external ADC exactly as you would the internal ADC of the Controller. The application library `ADCSimpleEFL.c` for instance makes available the Function `ADCSimple_GetMillivoltValue(..)`, which can provide you with the voltage in millivolts on an analog input of a particular ADC Block.

The actual application code is located in the file `ElektorShieldADC.c`, the critical function being printed in **Listing 2**. Next up we need to find out what the application actually does: the voltage on pin A3 of the Arduino is digitized by both the internal and external ADCs and displayed on-screen. If you check out the voltage with a

---

Listing 1. Relay (remote) control.

```
uint8 RelayPosition;

void ApplicationLoop()
{
  ButtonPoll(0);
  BlockProtocol_Engine();

  RelayPosition = ADCSimple_GetRawValue(0, 0) >> 7;
  Display_WriteNumber(0, 0, RelayPosition);
}


void ButtonEventCallback(uint8 BlockType, uint8
ButtonBlockNumber, uint8 ButtonPosition, uint8 Event)
{
  if (Event == EVENT_BUTTON_PRESSED)
  {
    ToggleLED(1, 0);
    SwitchRelay(0, RelayPosition, ButtonPosition);
  }
}
```

---

good multimeter you will discover the external ADC measures to a high level of accuracy.

This application also provides remote access using the Block Protocol. The command `A 0 0 # <CR>` causes the Arduino to return the value just sampled by the internal ADC in hex characters. `A 1 0 # <CR>` returns the last value of the external ADC. The indications are quite different because we are now dealing with raw values and

Figure 6.
Application 2: External 16-bit ADC on the Arduino Uno. The voltage on the pot is returned across the cable, enabling the readings of the internal and external ADCs can be compared.
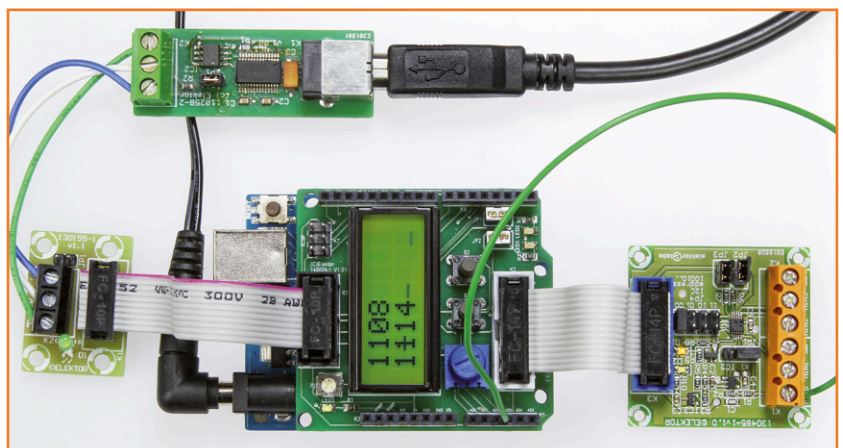
Virtualization

Port expanders and external A-to-D converters are two sample applications for chips that extend the capabilities of a Controller. Frequently these chips are addressed via an I2C Bus or some other serial interface.

A modular prototyping library should, as far as possible, be based on the hardware used. For someone developing an application (or a Library that activates Peripherals) it should be immaterial how the wiring tracks run on the board or to which Controller pins the Peripheral is connected. Ideally the developer should also not need to know whether an input or output is connected to a genuine Controller pin or merely to a Port expansion chip. We can solve that problem by assigning to the Controller in addition to its Ports 0, 1, … etc. some extra 'virtual' Ports that begin with the number 0x40 = 64, so as to be able to differentiate these from 'real' Ports.

If for example you wish to activate a relay connected to a Port expansion device, then access to the relay from the `RelayEFL` Library will be forwarded perfectly normally to the Function `SwitchDigitalOutput(…)`, located in the file `BlockEFL_IO.c`. This Function refers in the EFL Tables which Controller Port and pin the relay belongs and calls up the Function `IO_SetPinLevel(…)` in the Controller file. In the Tables a Port 0x50 is recorded for the relay and normally this would be the end of it, as the Controller is unaware of any Port 0x50. However, for such situations when the relay board is initialized, a special Function of the relay board code is passed to the Controller code that is called up in cases like this. This same Function then sends the corresponding I2C commands to set the output pins of the Port expansion unit located on the board.

The concept of 'Virtualization' is now broadened to analog inputs. In our case the external ADC is located on the EEC/Gnublin ADC Expansion board, so the code in file `ExtensionEFL_EEC_ADCModule16bit.c` must take over the activation of its ADC. When the board is initialized with the Function `ExtensionEFL_EEC_ADCModule16bit_Init(…)` the external ADC is integrated into the EFL Tables like an internal ADC (in our case with the Block number #1), where, however, it is assigned to the virtual Port 0x40. In addition the I2C interface is made ready. Furthermore the Function `Virtual_ADC_GetValue(…)` is notified to the Controller code.

Using the Function `ADCSimple_GetRawValue(uint8 ADCBlockNumber, uint8 ADCPosition)` users now have access to an ADC pin in a specific ADC Block, independent of the connection to the Controller. The Function directly calls up the Controller Function `ADC_GetValue(…)`. The Controller refers to the EFL Tables and on account of the high Port number recognizes that an internal ADC is not intended but instead it should call up the Function `Virtual_ADC_GetValue(…)`, located in the code file of the Expansion board. This results in giving access to the external ADC over I2C.

As well as the Function `ADC_GetValue(…)` our application also virtualizes the Function `ADC_GetParameter(…)`, with which the resolution and voltage range of an ADC can be read off. In this way ADC values can be calculated in millivolts.

not numbers of millivolts. To avoid inflating the Block Protocol library unduly, we have foregone having a Function for converting ADC values into millivolts.

Mini Protocol
Given that we can already calculate the millivolt readings in the application, couldn't we simply call them up using the UART?

Absolutely! We just need to concoct a dedicated mini Protocol. It we transmit `0 <CR>`, then we receive back the value of the internal ADC in millivolts. Typing `1 <CR>` should arrange that
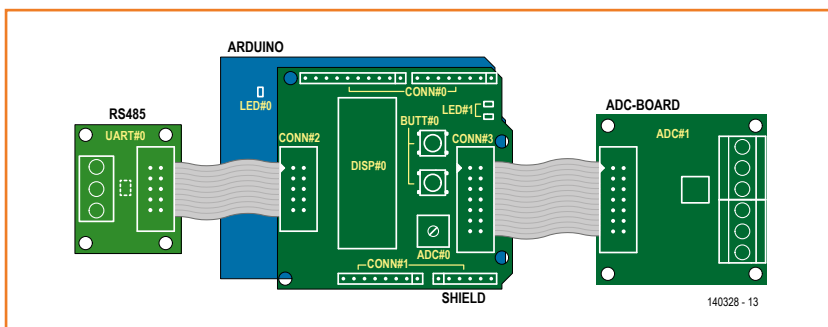

Figure 7.
Internal and external ADCs can be addressed using the same Functions, with the Block numbers #0 and #1 used for differentiation.
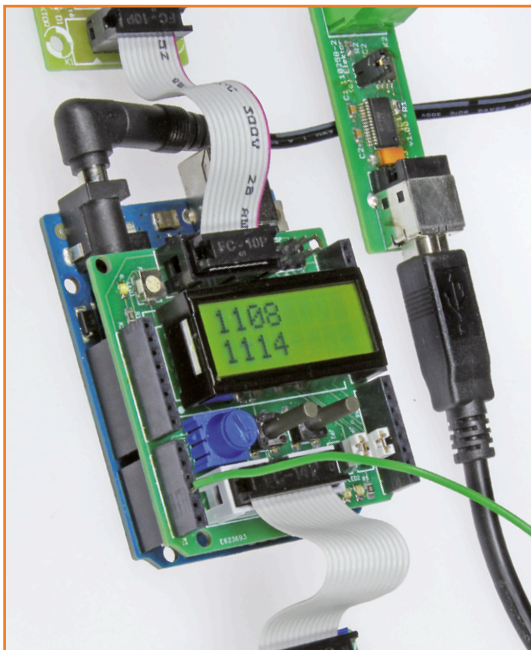
the Arduino send back the value of the external ADC. Now we can compare both values directly in the Terminal program.

Listing 3 shows the relevant code (project `ElektorShieldADCMiniProtocol.atsln` in the download). `ReceiveRingbuffer` contains the address of the receive ring-buffer, where the characters are written. These then reach the Arduino via the UART (UART Block #0).

The millivolt value is converted into hex digits in the application routine `SendMillivolt(…)` and sent forward via the UART #0.

In an upcoming issue we will introduce the Configurator, with which you can generate an EFL project yourself. We'll also provide simple instructions for writing your own board file. Stay tuned!

(140328)

Listing 2. Measurement using internal and external ADCs.

```
void ApplicationLoop()
{
  ButtonPoll(0);

  BlockProtocol_Engine();

  uint16 ADCValue1 = ADCSimple_GetMillivoltValue(0, 0);
  Display_WriteNumber(0, 0, ADCValue1);

  uint16 ADCValue2 = ADCSimple_GetMillivoltValue(1, 0);
  Display_WriteNumber(0, 1, ADCValue2);
}
```

Listing 3. Reading values with a mini Protocol.

```
void ApplicationLoop()
{
  ButtonPoll(0);
  //BlockProtocol_Engine();

  uint16 ADCValue1 = ADCSimple_GetMillivoltValue(0, 0);
  Display_WriteNumber(0, 0, ADCValue1);

  uint16 ADCValue2 = ADCSimple_GetMillivoltValue(1, 0);
  Display_WriteNumber(0, 1, ADCValue2);

  while (Ringbuffer_IsEmpty(ReceiveRingbuffer) == FALSE)
  {
    uint8 ReceivedChar =
Ringbuffer_GetByte(ReceiveRingbuffer);
    if (ReceivedChar == '1')
    {
      SendADCValueOverUART(ADCValue2);
    }
    if (ReceivedChar == '0')
    {
      SendADCValueOverUART(ADCValue1);
    }
  }
}


void SendADCValueOverUART(uint16 ADCValue)
{
  uint8 sd[3];

  sd[0] = (ADCValue & 0xFF00) >> 8;
  sd[1] = ADCValue & 0x00FF;
  sd[2] = 13;

  UARTInterface_Send(0, sd, 3);
}
```

Web Links

[1] www.elektor-magazine.com/140009

[2] www.elektor.com/development/gnublin/

[3] www.elektor-magazine.com/130157

[4] www.elektor-magazine.com/130485

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

[6] www.elektor-magazine.com/140328

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

Minuscule Prototyping and Stonehenging

By **Thijs Beckers** (Elektor Labs)

Prototyping! Hours of mindless gazing at the CAD system screen, the slight waver when ordering parts and PCB and finally 'fingers crossed' when flipping the On switch of the lab supply, powering up your newly designed and freshly built-up project. Often followed by a 'back to the drawing board', or, if you're in luck, back to the solder station to check solder joints and potential shorts between a couple of SMD IC pins.

If this project had been designed using through-hole components, the correction to the circuit would have been a lot easier. But Ton persevered and got it to work using '0805' on the resistors and '0603' on the capacitor, the latter now 'seated' between the pins of the SOIC-8 IC. For reference: the diameter of the enameled copper wire in the photograph is 0.2 mm (7.9 mil). But of course there are more examples. The right
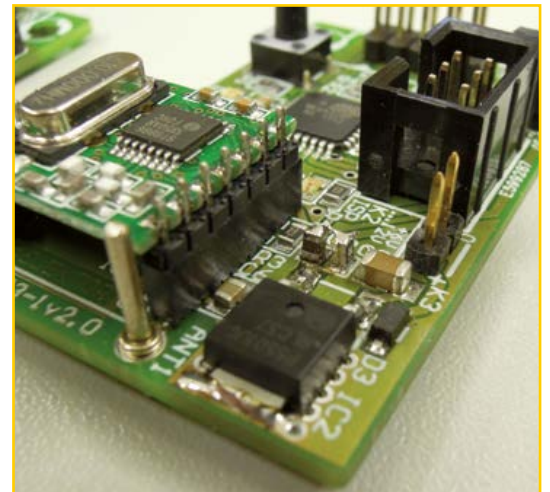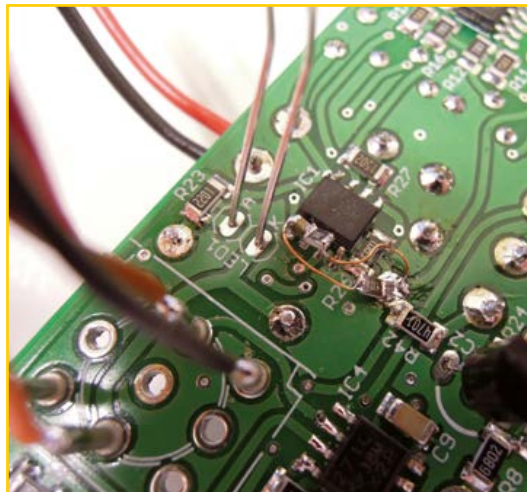
At Elektor Labs none of this is out of the ordinary. All designers are in this loop: design, test, correct if necessary, test again. Sometimes a completely new PCB design is called for. Although it may seem easier to tweak the prototype at hand, there too you can always run into convolutions. I followed what lab worker Ton Giesberts concocted as part of a recent project (see left photograph). The output of IC1, a TLC272 dual opamp, was found to oscillate. To remedy this, a 100 Ω stopper resistor had to be connected in series with the output of IC1, at pin 1. For this pins 1 and 2 had to be desoldered and lifted off their solder pads. But that wasn't all. As you can see with some effort, a capacitor had to be connected between pin 1 and 2 of IC1, and a 10 kΩ feedback resistor between pin 2 and the 100 Ω series resistor at the output on pin 1.

hand photograph shows a fine case of intentional tombstoning*. Here, instead of a single normal capacitor, a version with a high ESR (Equivalent Series Resistance) has been created using a standard capacitor and two paralleled 13 Ω resistors to give 6.5 ohms worth of ESR. A 'wire bridge' then connects the tombstoned components and creates a genuine miniature dolmen.

(140290)

Tombstoning, Stonehenging, and the Manhattan Effect refer to the tendency of small leadless components to tilt like a tombstone during the soldering process. The effect is due to the surface tension of molten solder.

.Labs

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

# Post on Elektor.Labs and get a job. Really!

Get Elektorized!

Do you remember this one? You say you do. But did you really? I did. And I now know of at least one other person who did too. This person posted some projects on Elektor.Labs and got a job. *Say what?* That's right—people get hired after publishing good stuff on our projects website. It happened to Labs user lux36. And what is even better, he told me about it. Congratulations lux36! Keep up the good work.
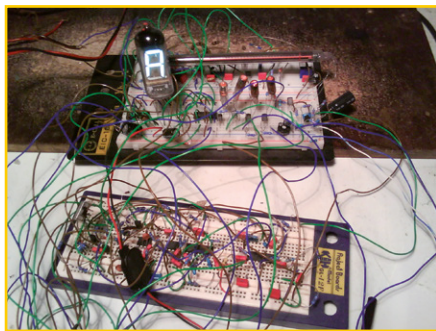
Surely other people out there have been elektorized in some way, but unfortunately never told us about it. That is the problem with people. Whenever there's reason for complaining they know how to find you, but when all is fine & dandy—not a dicky-bird. Oh well, that's life.

Anyway, lux36 now makes good money and all he did was putting a project of his online. Actually, he published several. And he can make even more money from this if we decide to work out and publish his projects in Elektor magazine. But for that he should provide Labs with a little more information, because the schematics are missing and the PCB design looks incomplete. So lux36, if you are reading this, please add some details to your project and maybe you will get elektorized once again through a publication in Elektor Magazine.

www.elektor-labs.com/node/3169

Elektor.Labs Preferred Parts (ELPP) library moved to GitHub

In the September 2014 edition we launched the Elektor.Labs Preferred Parts (ELPP) program, a component library meant to simplify circuit design by publicizing a list of frequently-used parts with tested footprints and popular distributor order codes saving you the trouble of looking them up each time you need one. Not only does this library contain electronic parts, it also includes electromechanical devices like connectors, relays and switches. The announcement generated a lot of positive response which has strengthened us in our belief that such a library is not only interesting for us, but for you too.
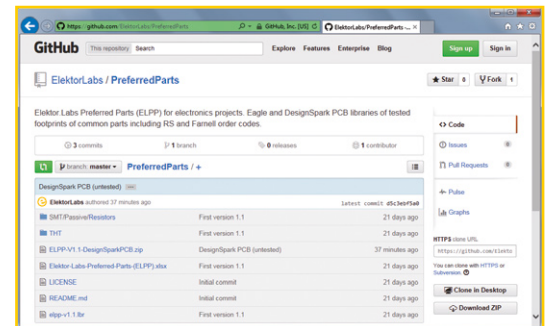
GitHub, ELPP's new host.

The ELPP library was initially posted on our FTP server, but we felt it would be better manageable if we moved it to a well-established online platform where it's easily accessible for all of us. We chose GitHub. The advantage, besides its accessibility, is the version control system it provides. This system not only makes available the most recent version of the ELPP library 24/7, it also keeps track of the changes made to the library. BTW, we just added the ELPP for DesignSpark PCB library.

https://github.com/ElektorLabs/PreferredParts

(140305)

Spaghetti breadboarding can get you far.

Add 3D Sensing to your Micro or PC

with the Microchip/Elektor development kit & 3D touchpad

Microchip Technology Inc. and Elektor International Media jointly prepare the launch of a product bundle for all design engineers and programmers keen on implementing 3D sensing and gesture control on their embedded systems and PCs. And touch control too.

Key to the project is the **MGC3130**, Microchip's first product based on GestIC® technology designed to detect changes in a transmitted E-field corresponding to capacitive changes in the femtofarad range (1 fF = 10^{-15} F).

The bundle consists of an **MGC3130 Hillstar Development Kit** and a **3D TouchPad**. The dev kit comprises an MGC3130 Module, an I2C to USB Bridge Module, a 4-layer Reference Electrode (95 x 60 mm sensitive area), a 'Hand Brick' set (self-assembly, 4 foam blocks, 1 copper foil), and a USB cable for PC connection. The kit enables the MGC3130 to be parameterized and the associated Gestic® technology to be explored in great depth. The downloadable **Aurea software tool** provides a graphic aid to viewing signals, perform logging, and set up the MGC3130. The kit also provides advice and tools to designing electrodes specifically for your application.

The **3D Touchpad** in the product bundle is a ready manufactured 3D Tracking and Gesture controller with mouse functionality included. Out of the box it adds a large sensitive touchpad to any PC, tablet or embedded system with USB. Internally, a single PCB implements both the electrodes for the 3D sensing as well as the touch sensor matrix for the touchpad.

The 3D Touchpad and the Hillstar MGC3130 Development Kit together provide a solid basis for making 3D control a reality on embedded systems including Arduino, RPi, BBB, T-Boards and others. In a number of follow-up publications, Elektor will describe simple and advanced applications of the Gestic® technology (TuxRacer is perfect to get started...).

Specially for this promotion, Elektor Labs have secured technology and applications backing from Microchip Technology product developers in Germany and the US.

The dev kit and touchpad product bundle will be retailed exclusively by Elektor at a special reduced price to be announced.

Further details and announcements on product availability will appear in the Elektor.POST weekly newsletter, in Elektor magazine, at the Elektor-Labs website and at Elektor Maker Space Munich 2014.

(140408)

MGC3130 Hillstar Single Zone Development Kit (Part # DM160218)

3DTouchPad DM160225

Neon Bulbs
Weird Component #9

By **Neil Gruending**
(Canada)

Figure 1.
Neon bulbs.

Figure 2.
Neon bulb schematic symbol.

Figure 3.
Neon bulb as relaxation oscillator, and typical output signal.
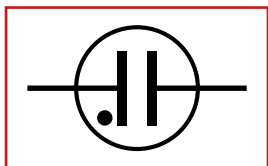

Figure 4.
Neon bulbs used in cascade acting as frequency dividers in an electric organ.

This installment was inspired by an old Tektronix power supply I saw that used a neon bulb. I originally thought it was a power indicator but a little probing proved that it was actually part of the power supply protection circuitry which surprised me. Let's take a look at how neon bulbs work and what makes them useful in applications like this.

Neon bulbs—or neon glow lamps as they're sometimes called—are pretty simple devices. They're made by filling a small glass tube with a low-pressure neon gas mixture with two electrode wires coming out of it like in **Figure 1**. When they are powered with a DC source only the negative electrode will glow but an AC source will illuminate both electrodes. The schematic symbol in **Figure 2** also resembles the neon bulb structure.

A neon bulb behaves like an open circuit when it is off and it won't turn on until its operating voltage reaches the gas breakdown voltage. Once the bulb is illuminated its operating voltage will decrease significantly and remain constant over a pretty wide current range. This behavior is why you will usually see a large 100+kΩ resistor in series with the bulb. The breakdown voltage is usually around 90 volts for a common NE-2 bulb and once it's illuminated the voltage will decrease to around 60 volts. Since the voltage thresholds can be affected by ambient light levels a small amount of radioactive gas is usually added to make sure that the bulb will illuminate as expected.

But how are neon bulbs used in a power supply? The bulb's high trigger voltage and built in hysteresis makes them useful as overvoltage protection devices, especially since they can give a visual indication of a fault condition. These days MOVs or TVS diodes are better choices as protection devices but neon bulbs are still a great high voltage indicator for power supplies.

The negative resistance characteristic of neon bulbs makes them useful in other applications as well. For example, a neon bulb can be used as the active element in a relaxation oscillator along with a resistor and capacitor (**Figure 3**). Two bulbs can also be used as an astable multivibrator, and cascading several of them together can make more complex circuits like ring counters, dividers (**Figure 4**) and other simple logic circuits.

Neon bulbs have been around for a long time and they have a wide variety of applications. It's even possible to see them in unexpected places like in power supplies as I discovered (and in a 1960s electric organ, *Ed*.). But one of the best things about them is that they're still available in a wide variety of styles so they're easy to find and experiment with. The only thing to remember is that the newer style devices typically have a narrower breakdown-to-holding-voltage range so some circuits may need to be adjusted before they will work properly.
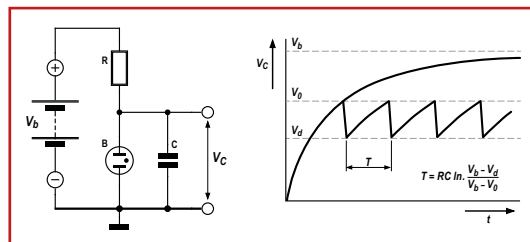
(140291)
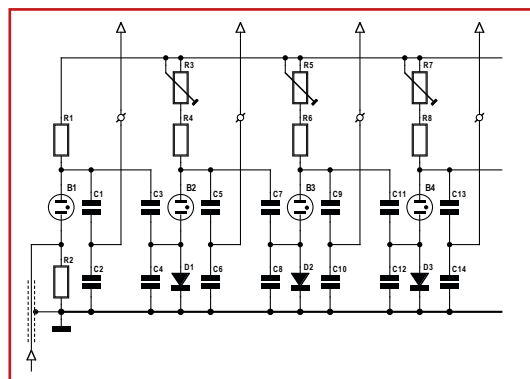
SMD Feedthrough Capacitors

BOM: up 1 component, EMI: down 60 dB

Even old hands at electronics may have a moment of discovery when they stumble on feedthrough capacitors. Originally from the Bob Pease & Radio Age these devices are now available in SMT offering brilliant specifications in terms of noise suppression.

By **Kirk Sceviour**
(Canada)

That very discovery occurred to me recently while upgrading an old communications project. The new design contained several new components and a dual polarity supply, all of which had to fit in the original board dimensions. It doesn't take long to run out of space on a small board, even

Figure 1.
Insertion loss for several feedthrough chip capacitors values. (source: Murata)
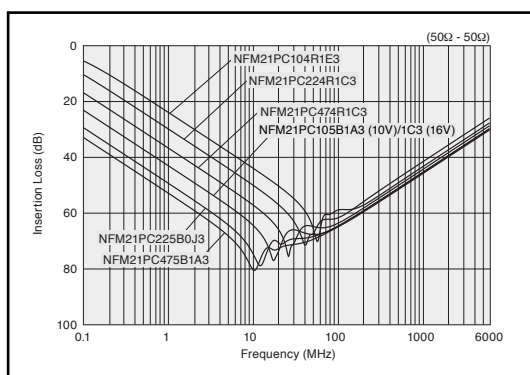
Figure 2.
Typical dimensions of SMT feedthrough capacitors (source: Murata).

with exclusively surface mount devices (SMDs). My eye fell immediately upon a glaring space consumer: the relatively large LC filter network of a DC/DC converter. Could this filter network be reduced in size, even replaced? Are the traditional combinations of resistors, inductors and capacitors the only way to filter signals with discrete components? My researches lead to a remarkable new (to me!) part: the feedthrough capacitor.

Not surprisingly the device comes in a myriad of package designs and a quick Internet search revealed a variety of options, from very large panel (chassis) mount to 0402 surface mount. Some examples of typical feedthrough capacitors are shown in the head illustration.

Further study revealed that a feedthrough capacitor will pass an AC or DC signal while shunting EMI (at the resonant frequency) to ground. The more common use of these devices seems to be in high current, high frequency situations where a signal passes through a shielded enclosure. Some filter devices (known as tubular filters) may use a single feedthrough capacitor or several inductor and feedthrough capacitor combinations to provide the desired response. The grounding points of the internal components are connected to a ground terminal or the body of the device itself, forcing any EMI nasties riding on the data, audio signal, or supply voltage to see a low-reactance path to ground or the equipment chassis. While

more common, the uses of feedthrough capacitors are not limited simply to heavy duty chassis mounted applications. My real interest lay with the surface mount variety, which is available in the familiar multi-layer ceramic X7R and NP0 flavors. With some skills these chips can be placed using normal soldering techniques. Regardless of the package they come in, feedthrough capacitors are noise filters contained in a single component and they have some impressive performance characteristics.

Small device, great specs

As an example; the data sheet for the Murata .1µF 0805 NFM21PC series feedthrough capacitor indicates an insertion loss exceeding 60 dB centered at 55 MHz (**Figure 1**). That's a significant bit of filtering in a very small package (**Figure 2**).

These devices are easy to get, available in a reasonable capacitance range and they're not lightweights when it comes to power ratings. Surface-mount feedthrough chip capacitors are available with ratings up to several hundred volts and 5 amps or more. They really shine at the higher frequencies; **Figure 3** shows a graph obtained from Murata's web site [1] comparing insertion loss characteristics of standard chip and leaded ceramic capacitors to a 3-terminal flow through chip filter.

For your application too?

The obvious question is: what makes a feedthrough capacitor different from any other conventional capacitor—or filter network for that matter? The differences lie in how they are applied to a circuit, their internal structure and electrical characteristics. In addition to the usual electrode and ceramic layers, feedthrough capacitors sport extra printed layers that run perpendicular. It is this extra layer that makes them 3-terminal devices.

Shown in **Figure 4** is a typical feedthrough capacitor application (a) compared to the same circuit using a conventional capacitor (b). Both circuits are filters.

The different wiring application can be seen in the diagram. A normal surface mount capacitor has two terminals only and no 'preferred ground side'. The feedthrough C essentially has three terminals: two 'signal' and one 'ground'.

The crux

As usual (at least for old hands @ radio) the real difference lies in what a signal 'sees' when it arrives (or wants to exit from a circuit). The feedthrough capacitor has a reduced inductance to ground compared to the conventional capacitor. The difference can be one order of magnitude when using a feedthrough design.

Equivalent electrical circuits are shown in **Figures 5a** and **5b** for both the filter types. From the EECs it's obvious the electrical performance of both devices will be remarkably different. The feedthrough capacitor has minimal parallel inductance and an increased series inductance com-
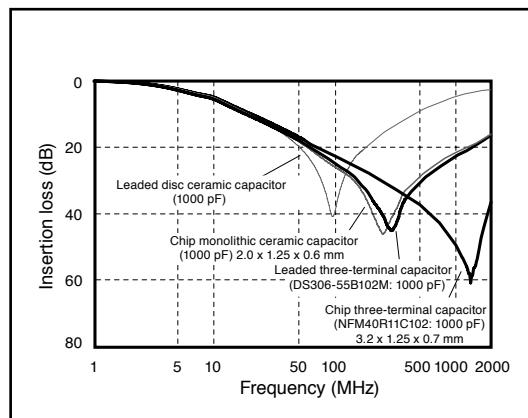

Figure 3.
Insertion loss of feedthrough chip versus regular capacitor models.
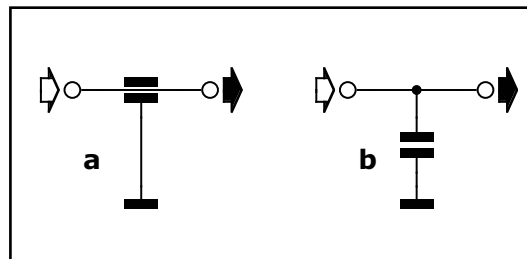(source: Murata)

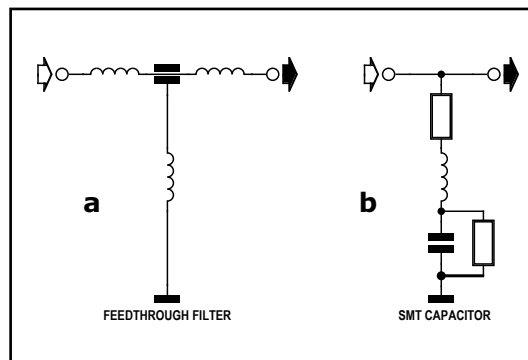Figure 4.
Feedthrough filter (a) and conventional filter (b).

Figure 5.
Equivalent electric circuits for feedthrough filter (a) and conventional filter (b).
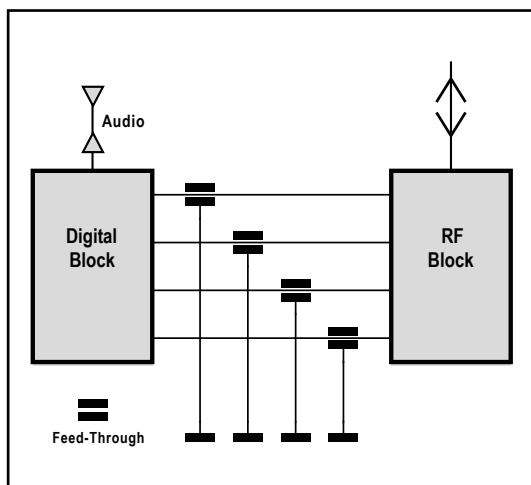
Figure 6.
Budding IoT and WiFi hardware designers, use feedthrough capacitors on all data and supply lines between your digital blocks and RF blocks. (source: AVX)

pared to the usual SMT capacitor. Most manufacturer graphs indicate a larger frequency response curve as a result.

Perfect for Data-to-RF interfacing

So why aren't these devices found in more circuit designs, particularly in RF clocking and high speed digital circuits of the sort we often see here in Elektor? I'll leave that discussion to the experts. For my part, a bit of exploration revealed a wide range of potential applications varying from V_{CC} conditioning to clock, data line and PA filtering. The SMD feedthrough capacitor (a.k.a. flow through capacitor) seems particularly well suited for the rapidly advancing Internet of Things (IoT) and ever expanding use of wireless devices (**Figure 6**). A prime example of this is the Digital to RF interface essential to any wireless system. Significant noise introduced between the digital base and the radio transceiver will corrupt or render useless any transmitted data. Properly selected feedthrough capacitors on the data lines could provide useful EMI protection particularly in a noisy environment. The advantages gained from the added EMI filtering should be well worth the extra cost and board space.

Any digital output line that exerts precision control over another device is susceptible to EMI and a possible place for feedthrough filtering. One situation that springs to mind is in radio control systems. An RC device can fall victim to its own transmitter if it's operated in close proximity to the board. A feedthrough filter centered at or close to the transmitter's frequency and placed on the microcontroller input pin should be very effective at eliminating EMI caused by the transmitter itself. I would be very interested to see

how this solution would work for Jan Lichtenbelt's *RC Speed Control for DC Motors* published in Elektor July & August 2014 that had some high frequency components at 40 MHz due to this problem [2].

At time of writing I was unable to obtain a suitable variety of feedthrough devices to play with. However, I suspect that the 2.7 MHz switching frequency of my converter is too low for a feedthrough filter to really be effective. This does not necessarily eliminate their use with low frequency supplies; if filtering acceptable to application can be obtained, then it might be worth it to lower the parts count and free up board space.

The existence of feedthrough capacitors may be old news to many, but they have been an interesting discovery for this enthusiastic amateur. In fact, my next project may very well be picked with their use specifically in mind… no telling where it might lead.

(140147)

Web Links

[1] Murata:
www.murata.com/~/media/webrenewal/
products/emc/emifil/knowhow/20to22.ashx

[2] www.elektor-magazine.com/130259

Ordering Information

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.

Microchip Extends Spotify® Connect Support in New JukeBlox®

Microchip Technology Inc., released Spotify® Connect in the standard Microchip JukeBlox® Platform. This release extends support to more than 8 million audio products based on Microchip's network audio processors and includes a number of key improvements on the initial release. Audio brands can easily add Spotify Connect to existing products through firmware upgrades and new designs based on all of Microchip's CX870 Wi-Fi® modules and DM860 Ethernet processor. To speed product and firmware availability to consumers, Microchip's APT Lab is the first to offer pre-certification services to Microchip customers using Spotify Connect.

Spotify Connect does not require the mobile device to continuously stream content to a wireless speaker or AV receiver. The key benefit of Spotify Connect is that once a track has been selected, the audio stream is delivered directly from Spotify's servers to the wireless speaker using the local network. This frees the mobile device for use during music playback which greatly reduces battery depletion and allows the mobile device to move anywhere in the network without interrupting music playback. Audio devices powered by Microchip's JukeBlox platform, such as wireless speakers, AV receivers, Internet radios, home theater systems, wireless speakers and portable music player docking stations, will be part of the Spotify Connect experience.

Spotify Connect is an award-winning digital music service that provides on-demand access to more than 20 million tracks. Spotify makes it easier than ever to discover, manage and share music. Spotify Premium users can control and play their music through their phone, tablet and audio devices simply and effortlessly, at the touch of a button. Spotify Connect is available in 28 markets including USA, UK, Sweden, Finland, Norway, Denmark, France, Switzerland, Germany, Austria, Belgium, The Netherlands, Spain, Australia, New Zealand, Ireland, Luxembourg, Italy, Portugal, Singapore, Hong Kong, Malaysia, Poland, Estonia, Latvia, Lithuania, Iceland and Mexico, with more than 24 million active users, and over 6 million paying subscribers.

Spotify Connect is available today, via a free download (link below). Spotify offers a 30-day free trial period and thereafter there is a $9.99 monthly fee for Premium Service.

http://www.microchip.com/get/NBWE   (140335-II)

Nanodiamond Cost Reductions of up to 70 percent for Electronics and LED Applications

Finnish company Carbodeon claims to achieve a 20 percent increase in polymer thermal performance by using as little as 0.03 wt.% nanodiamond material at 45 percent thermal filler loading, enabling increased performance at a lower cost than with traditional fillers.

Last October, Carbodeon published its data on thermal fillers showing that the conductivity of polyamide 66 (PA66) based thermal compound could be increased by 25 percent by replacing 0.1 wt.% of the typically maximum effective level of boron nitride filler (45 wt.%) with the company's application fine-tuned nanodiamond material. The latest refinements in nanodiamond materials and compound manufacturing allow similar level performance improvements but with 70 percent less nanodiamond consumption and thus, greatly reduced cost.

The samples were manufactured at the VTT Technical Research Centre in Finland and their thermal performance was analyzed by ESK (3M) in Germany.

The active surface chemistry inherent in detonation-synthesized nanodiamonds has historically presented difficulties in utilizing the potential benefits of the 4-6nm particles, making them prone to agglomeration. Carbodeon optimizes this surface chemistry so that the particles are driven to disperse and to become consistently integrated throughout parent materials, especially polymers. The much-promised properties of diamond can thus be imparted to other materials with very low, and hence economic, concentrations.

For more demanding requirements, conductivity increases of as much as 100 percent can be achieved using 1.5 percent nanodiamond materials at 20 percent thermal filler loadings.

www.carbodeon.com   (140335-III)

ams: NFC for Microcontroller Systems

ams' new AS3911 NFC development kit and interface software stack provides a blueprint for an NFC implementation in any microcontroller-based system.

The new AS3911 NFC development kit from ams eliminates the need for the OEM designer to implement a complete, proprietary software interface between a host microcontroller, its operating system and the NFC reader IC.

The software in the AS3911 development kit includes an NFC Controller Interface (NCI) stack, a standard-based modular firmware/software solution, operating from the hardware level up to the operating system. Developed in collaboration with Stollmann E+V GmbH, it manages the interaction between a microcontroller and any NFC/HF reader in the AS391x family from ams.

Fully compatible with the Android, Linux, Windows 7 and Windows 8 operating systems, the AS3911 development kit lets developers quickly and easily create NFC applications for multiple microcontrollers. The standard interface is suitable for any kind of NFC-enabled device, including routers, set-top boxes, automotive infotainment systems, consumer electronics devices and home appliances. Because the NCI stack has a modular design, users can optimize it for their system, selecting only the features and functions required by their application. This means that designers can optimize for a minimal microcontroller processing overhead and memory usage, or for high performance and multi-protocol support.

The NCI supports all NFC protocols specified by the ISO standards organization, as well as providing extended functionality for proprietary card systems. It also supports the automatic antenna tuning feature provided by ams' AS391x family of NFC readers. 'The broad adoption of NFC in mobile phones is leading to a new wave of interest in NFC from manufacturers of
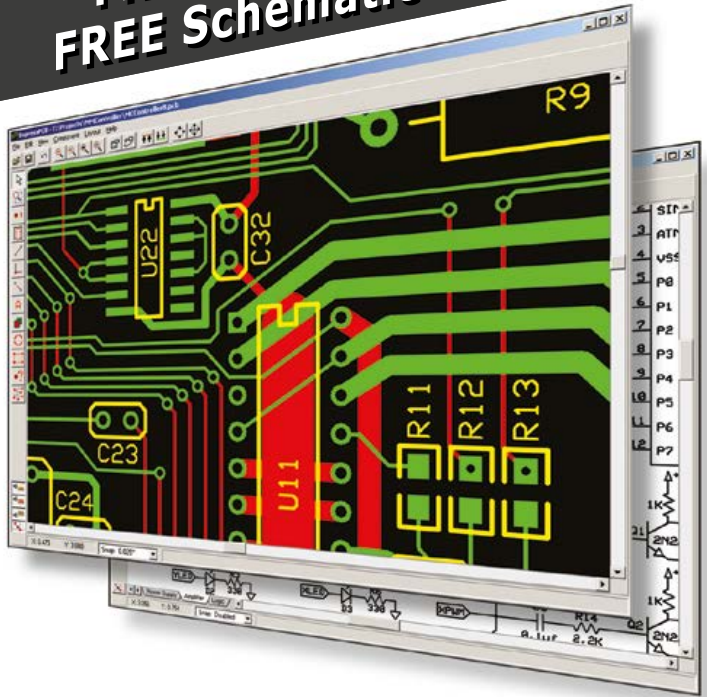
devices which can make a contactless bridge to mobile phones. Because of its AS391x family of NFC readers, ams is the ideal partner for Stollmann, which is positioned perfectly for the substantial market growth we expect,' said Juergen Schick, CEO of Stollmann E+V GmbH. 'Our easy to integrate software library complements the AS3911's unique features, helping OEMs to achieve a faster time to market.'

The NCI software stack is compatible with various physical interfaces including UART, SPI and I^2C.

The AS3911 development kit and NCI software stack are available immediately.

www.ams.com/NFC-HF-Reader/AS3911   (140386-I)

Autonomous Robotic Vehicle Kit is C Programmable

Global Specialties' new RP6V2 Robot Kit with RC5 remote and battery charger is an economical autonomous mobile robot system which provides an introduction to the fascinating world of robotics. It is designed for beginners as well as experienced electronics and software developers. Programmable in C, the RP6V2-C has many possibilities for expansion as your programming skills grow. The RP6V2-C is ideal for educational curriculum at universities, trade schools, high schools and of course hobby users. With an extensive manual, lots of example programs, and a huge C function library, programming is easy and you can instantly start experimenting with your robot.

Features:
- ATMEGA32 8-bit RISC microcontroller with 8 MIPS and 8MHz clock
- Delivered fully assembled (no soldering needed)
- CD with software, 138 page manual, and many extras
- AVR-GCC and RobotLoader open source software for use with Windows and Linux
- Programmable in C
- Receives IR codes in RC5 format from the included remote control
- USB Interface for easy programming and communication
- Modular I2C bus expansion system
- Expansion boards may be stacked as needed
- Sample C programs and huge C function library
- Powerful tank drivetrain can negotiate steep ramps and obstacles
- Large payload capacity
- Light, collision, speed and IR-obstacle sensors integrated
- Two 7.2 V DC motors
- 625 CPR encoder resolution for precise speed regulation
- Six PCB expansion areas

RP6V2-C comes with the following items: RP6V2 robot; CD with software, user manual, and sample programs; 10-pin connector; USB connector cable; USB programmer interface; High speed battery charger; RC5 remote control. Available immediately, the RP6V2-C has a list price of $269.

www.globalspecialties.com   (140386-II)

Altium Blog on Open Source Hardware Design Products

Written by Altium field applications engineer Petr Tosovsky, a new blog page published by Altium describes a number of exciting Open Source Hardware (OSHw) projects designed with Altium Designer, including the first totally OSHw laptop and an electric vehicle. For all designs, there are links to the source files.
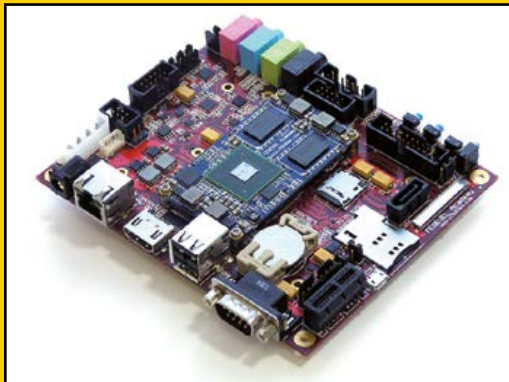
Two projects on the blog are shown here: RHINO, the Reconfigurable Hardware Interface for computiNg and RadiO, and iMX6 REX, a processor module sporting a 1.2 GHz Freescale iMX6 CPU (dual- and quad core). In addition to "formally open source" designs are those which have been publically developed utilizing Altium tools. Among them, for example, are Red Pitaya or the ZedBoard. To make it easier to design these projects to popular and capable form-factors, the Altium Content Store offers to its users a relatively large library of template projects, including standard board outlines (with connectors) like PCI, PCIe, SODIMM, VME, QSeven etc. It is widely known that Elektor Labs are long term users of Altium Designer. At the url below, find: Open Source Hardware.

http://blog.live.altium.com   (140386-III)
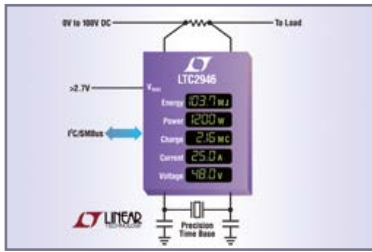
0 V to 100 V, 1% Accurate Energy Monitor has 12 Bit Output

Linear Technology Corporation introduces the LTC2946, a high or low side charge, power and energy monitor for DC supply rails in the 0 V to 100 V range. An integrated ±0.4% accurate, 12-bit ADC and external precision time base (crystal or clock) enables measurement accuracy better than ±0.6% for current and charge, and ±1% for power and energy. A ±5% accurate internal time base substitutes in the absence of an external one. All digital readings, including minimums and maximums of voltage, current and power, are stored in registers accessible by an I²C/SMBus interface. An alert output signals when measurements exceed configurable warning thresholds, relieving the host of burdensome polling for data. The LTC2946 provides access to all the necessary parameters to accurately assess and manage board level energy consumption. In addition its wide operating range makes it ideal for monitoring board energy consumption in blade servers, telecom, solar and industrial equipment, and advanced mezzanine cards (AMC).

The LTC2946 can be powered from 2.7 V to 5 V directly, from 4 V to 100 V through an internal linear regulator, or beyond 100 V through an internal shunt regulator. Two of the three general purpose input/output (GPIO) pins are configurable as an accumulator enable and alert output. The internal ADC operates in either a continuous scan mode or a snapshot mode. In shutdown mode, the device current consumption drops from 900 µA to 15 µA.

The LTC2946 is available in two options: the LTC2946 I²C interface has separate SDA input and output pins for standard or opto-isolated I²C, whereas the LTC2946-1 has an inverted SDA output for inverting opto-isolator configurations. Specified over the commercial, industrial, automotive and military temperature ranges, the LTC2946 is offered in 16-pin MSOP and 4mm x 3mm DFN packages. 1,000-piece pricing starts at $3.95 each. Device samples and evaluation circuit boards are available online or from your local Linear Technology sales office.

www.linear.com/products/power_monitors   (140386-V)

A Sensor Pro in Every Discipline

Sensirion is taking sensor technology to a new level this year. At the electronica event in Munich, they will present the most advanced platform for humidity and temperature sensors: the Platform3x with the powerful SHT3x sensor series. Sensirion is launching an innovation that excels across the board and a sensor that outperforms all previous models.

The versatile Platform3x consists of a group of humidity and temperature sensors with different precision levels and features. The Platform3x is thus optimally designed for individual applications on the market. Whether for the cost-conscious, the groundbreaking or the high-end product that demands the best humidity and temperature sensor, the Platform3x impresses in every discipline and provides the ideal solution for all precision classes and various interfaces. The SHT3x combines the strengths of the established SHT1x, the revolutionary SHT2x and the advanced SHTC1 series in a single, unique product. But that's not all by a long way. The SHT3x includes a user programmable alert function, where the sensor can be used as a humidity and temperature sensor. Moreover, Sensirion's latest innovation contains another world premiere, an analog ratiometric voltage output. This is the first fully calibrated and linear digital/analog humidity and temperature sensor. The SHT3x series thus combines multiple functions and various interfaces (I²C, voltage out) with a user-friendly, very wide operating voltage range (2.4 to 5.5 V). Like all sensors from Sensirion, the SHT3x is based on the unique CMOSens® Technology, which allows a high production volume at an exceptional price/performance ratio. In addition, the technology enables a small footprint of 2.5 x 2.5 mm with a height of 0.9 mm.

www.sensirion.com   (140386-IV)

Heathkit AA-100 Tube Amplifier (1960)
$85 DIY kit challenges commercial amps

By **Jan Buiting**,
Editor-in-Chief

First a boatanchor, now a fossil, the Heathkit AA-100 tube power amplifier shone when it came out in 1960: 25 watts stereo, cheap, 'sweet' output tubes, and a ton of inputs. Plus, the fun of building it yourself from parts. An AA-100 surfaced a while ago and landed on the Retronics workbench.

The Heathkit company stands monumental in the history of electronics. Practically every engineer in the age range line frequency to slightly above ½√2 recognizes the brand in connection with a kit they once built, owned—and sold on— or simply admired be it for ham radio, test and measurement, or power supplies. Personally I recollect working with a "Heath" VTVM and a grid dipper in the mid-1970s, though I had not assembled these. The Heathkit manuals are probably as good as it gets in terms of utter clarity and tech illustrations, and they still make fantastic reading if only for wanting want to learn about e-terminology specifically with a US slant, and e-didactics.

Our man in Belgium
In was contacted by a longtime[2] Elektor reader wanting to respectfully dispose of "some of his equipment" on condi-tion of a few of the more prominent items being described in *Retronics*. I visited Raymond in the spring of 2013. Now an emeritus Professor of Computer Technology, he studied and taught at various US universities including the famous Californian ones. Raymond returned to Holland where he taught IT-CT at Nijmegen University. He now enjoys retirement in Belgium, just 40 miles across the border from Elektor Castle, and still works on electronics concentrating on com-puter-to-radio interfacing.

Among the vast amount of unpackaged equip-ment collected from Raymond's deluxe dwellings was a very heavy cardboard storage box.

It's gotta be American
Having unpacked the box and seen "Heathkit/Day-strom" I did what most restorers and collectors do: seek assistance from the original manufacturer, though long since "folded". Google passed the word though: "The Heathkit AA-100 Deluxe Ste-reo Amplifier stands heads and shoulders in value above any other stereo amplifier in the kit industry. Packed full of exciting features, this high-fidelity rated 50 watt amplifier outperforms units selling for twice its price. It's a mere $84.95 in kit form, that's just $1.70 a watt for two 25-watt channels of distortion-free power to handle every amplify-ing task in your stereo system."
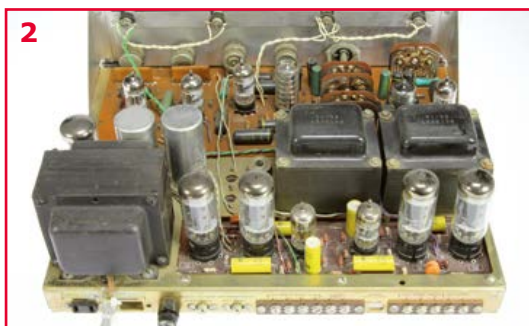
"While the electronics of the AA-100 are enough to prove this stereo amplifier the best buy on the market today, the wonder styling features of this beauty make it an even greater value. The luggage tan vinyl-clad steel cabinet looks and feels like real leather, but is practically indestructible because of its mar, scratch and stain-resistant qualities. The fabulous AA-100 is the "stereo-twin" of the famous Heathkit AJ-30 Stereo AM/FM Tuner ... they look alike, and will fit in any room decor in your home." [1]

I also looked into some audio and vintage electronics forums and found good advice on must-do enhancements and restoration issues. I ignored the pointless debates on tube replacements, transparent sound, violin solos, super-sonic capacitors and oxygen starved cables.

X is for...

With the AA-100 on the bench my first observations were: that's not real leather; this thing is (real) heavy (man); it has no back cover (wow!)—and no power switch either. Also, it's been modded at the back in the AC power area (yuk!); the Heathkit/Daystrom label is at the wrong side; the signal inputs are at the underside (!); and these loudspeakers are pathetic considering the grand appearance of the amp.

The inside of this AA-100 looks neat with the usual dust of course (**Figures 1, 2**) and all soldering is clean as are the wiring and parts positioning. The controls look a kind of white, though all images on the net suggest they are transparent golden. Intriguing.

I always resist the temptation to power up old equipment straight out of storage. In this case the hesitation was strengthened by the apparent mod in the AC-in area at the back panel (**Figure 3**). Suspiciously the original 2-way SWITCHED socket was not fitted at all and the NORMAL socket was labelled "Remote Switch" in sharpie handwriting. These sockets, one switched, one unswitched, are originally intended to power ancillary equipment like the beautiful AJ 30 tuner. Sure enough the storage box contained the Remote Switch—nothing more than a length of 2-wire cable with a US style AC plug at one side and a plastic in-line slide switch at the other.

Having come to know Heathkit as a very American manufacturer, meaning 117 VAC / 60 Hz when in Europe we are on 230 V 50 Hz generally, I was particularly weary about the AC line

AA-100 Specifications	
Power output:	2 x 25 watts stereophonic or 50 watts monophonic
Music Power Rating:	30 watts stereophonic (.7% THD at 1 kc)
	60 watts monophonic (.7% THD at 1 kc)
Input Sensitivity	(V_{RMS} for 25 watts output per channel)
	Monophonic PHONO* (L channel only): 1.5 mV
	*for magnetic cartridges
	Stereophonic PHONO*: 1.5 mV
TAPE HEAD:	1.0 mV
TUNER:	0.2 V
AUXiliary 1:	0.2 V
AUXiliary 2:	0.2 V
Input Impedances:	PHONO: 47 kΩ as supplied, adaptable to cartridge
	TAPE HEAD: 470 kΩ
	TUNER and AUXiliary: 250 kΩ each
Output Impedances:	4, 8 and 16 Ω each channel
Tape Recorder Output:	Approx. 0.5 V max. at 600-Ω source resistance from cathode follower.
	Min. 150 kΩ.
Frequency Response:	± 1 dB 30 – 15,000 Hz at 25 W, from AUX inputs.
Channel Separation:	42 dB min. at 1 kHz
Damping factor:	15
Harmonic Distortion:	<0.5% at 25 watts, 1 kHz. <2% at 25 W, 30 – 15,000 cps
Intermodulation Distortion:	<1 % at 25 watts, 60 cps and 6000 cps mixed 4:1.
Hum and Noise:	PHONO: 55 dB
	TAPE HEAD: 35 dB
	TUNER and AUXiliary Inputs: 70 dB
Equalization:	PHONO: RIAA curve
	TAPE HEAD: NARTB tape playback curve
Tone Controls:	Bass 15 dB boost /17 dB cut. Treble 12 dB boost / 20 dB cut
Tube Complement:	2x EF86, 4x 12AX7, 2x 7199, 4x 7591, 1x GZ34
Size:	15-1/4" (w) x 4-3/8" (h) x 13-1/2" (d) (max., feet included)
Net weight:	28-3/4 lbs.

aspects of the amp. Digging deep in the box I found another homebrew item, a hard plastic case (**Figure 4**) with a switch, a power light and an IEC power socket on the top panel, and immediately thought this would contain the 230 V / 115 V transformer. Wrong, there was an auto transformer inside to reduce the AC line voltage by about 9 volts or 18 volts, strapable. For good reasons tube & vintage radio lovers in Continental Europe have taken precautions against the

effects of the AC line voltage being raised gradually from 220 V nominal to 230 V nominal (and 240 V maximum). It's currently 228 volts @ Elektor Castle.

So I thought this power transformer in the AX-100 surely has double 115 V windings with a center tap and the primaries are series connected for 230 V operation. Wrong again—having inspected the dead simple transformer AC line-side wiring I found there are just two primary wires connected exactly as shown by the Heathkit schematic with the exception of the On/Off switch. Only the color of one primary wire wasn't black but green & black. This is a different transformer than mentioned in the manual! And indeed the type code reads: 54X-89 (**Figure 5**), not 54-89. eXport? Heathkit people, please assist here.

The wiring to the Power On/Off switch was installed inside, but not the switch. It was not in the box either. Covering the rectangular hole for it in the front panel was the Heathkit/Daystrom label.

At this point I was confident the AA-100 amp was a 230-V AC type and I could connect it to my faithful variac for a soft wakeup call. Later.

Just to recap…

The AA-100 combines PCBs with discrete wiring, though no wire looms are found. Just a few parts are seen mounted in the air or fitted on a solder lug. Remarkably, two thick-film devices are used in the L and R tone control circuitry. These are called "packaged electronic circuit" (PEC) by Heathkit.

Although I have never been a fan of tubes mounted on PCBs (especially not the thin Perti-nax stuff) in the AA-100 there are no apparent problems even with the final tubes which run very hot. Those look like good quality tube sockets with large, spring-loaded terminals.

Raymond had supplied a bunch of notes indicating the final amp board had been overhauled with new capacitors and resistors in 1992. That's foresight & good practice. Polypropylene capacitors to replace leaky "black bombs", and improved resistors for the old carbon-something ones gone high-R. The original caps supplied by Heathkit have flimflam names on them like: PYRAMID, SANGAMO, MICAMOLD TROPICAP (!) which is enough to ruffle audiophile feathers. Many capacitors were left over after the recap operation, I found them in a separate box (**Figure 6**). Sadly all of the brand new electrolytic capacitors also pictured are too tall to close the equipment case. Even sadder the blue ones are 180 µF single electrolytics which is way too much for the GZ34 rectifier tube to handle. At 450 V they're also too low on voltage spec.

As part of the 1992 restoration job a new tube complement was fitted, all voltages were measured and recorded, and the original tubes got stored in boxes they do not belong in (**Figure 7**). Heathkit apparently had Mullard UK rebadge 12AX7's (ECC83) for them. I checked all original tubes on my uTracer tube tester and found only one 7591 a bit low on emission though not unusable. I liked seeing the GZ34 rectifier tube (now fetching $$$ if original Philips) and another Philips Holland gem the EF86! In fact only the driver and final tubes are US designed types.

I plan to overhaul the tone/preamp board in the AA-100 in the future. I expect some flaky C's and R's there too.

Pre-HiFi stereo with bells & whistles

The driver and final sections of the AA-100 schematic are reproduced in **Figure 8**, the full schematic is available from many places on the net, including Vintage Radio [2].

This is a classic push-pull power amplifier with a slightly unusual choice though of the final tubes. Also remarkable is the double bass/treble control—yes separate for the left and right channels, not forgetting the curious SEPARATION function and the CENTER SPEAKER connections, all provided to—Heathkit says—reduce the "hole in the middle" effect found in some stereo program

material. With the L and R loudspeakers spaced too far even Phil Spector's "Wall of Sound" may have needed center speaker support. Another gimmick, phase shift on the left channel was available on the A-100. The switch and wiring were removed by the previous owner.

Looking at the schematic the pentode section of each 7199 (V7 and V8) serves as a voltage amplifier. It is direct coupled to the triode section configured as a phase splitter, delivering anti-phase (push-pull) signals to the 7591 final tubes (V9/V10 and V11/V12) adding their output power in the output transformer, where the total power is coupled to the loudspeaker load (4, 8, or 16 ohms). The tubes are operated as straight pentodes with plate, screen and bias potential chosen for maximum undistorted power output. A fixed bias (input grid voltage) of −16 V is derived from a half-wave selenium rectifier.

The value of the grid leak resistors R91/R92, R85/R86 got widely criticized for being too high—the 7591 datasheets specify a maximum of 300 kΩ where Heathkit uses 470 kΩ. A must-do mod for the AA-100 is to drop the resistors to 220 kΩ or 270 kΩ. In my AA-100 the selenium rectifier got replaced by a silicon diode and the associated resistors were modified to maintain a −16 V bias.

Plastic fantastic

Back to the cosmetics, I found that all front controls on the AA-100 and even some of the internal wiring had fine specks on them not unlike mold or paint sputtering. The worst affected were the six large plastic knobs, which looked off white instead of translucent gold. The black knobs were covered in light brown specks. **Figure 9** shows the effects of a quick clean of one knob pair (right) with no more than a mild detergent and a toothbrush. Fortunately the plastic was not degraded—the growth is superficial. If you can explain the cause, please write.

Cleaning the other knobs and the faceplate was deferred for the purpose of this article, in *Ret-*

ronics we prefer to show equipment as it arrives— or *dans son jus* as the French say.

Next step: The XX

With the AC side of things okay and the driver/ PA board recapped the AA-100 was connected to my variac and gently powered up by increasing the supply voltage in 20 V increments up to the nominal value over a full day. Nothing untoward happened.

I banned the pitiful little loudspeakers that came with the AA-100 to the attic and instead connected a pair of refurbished Philips 22RH427's [3] (1973, closed box, 35 liters, 3-way, double 8" woofer). This big American amp I figured being so far from its 60 Hz home turf in Benton Harbor, MI, deserved US indie sound, vocals through a 1946 microphone and the warm, rich sound from a Gretsch semi-acoustic guitar. So I played *16 Horsepower Live [4].* In comparison with my fully overhauled Philips AG9015 (1966) the sound is dark and punchy. The AA-100 is much louder than the AG9015 though and should do well at parties. Although it has the X factor (54X) my AA-100 still needs work on that preamp PCB and a good leather wax shine.

(140333)
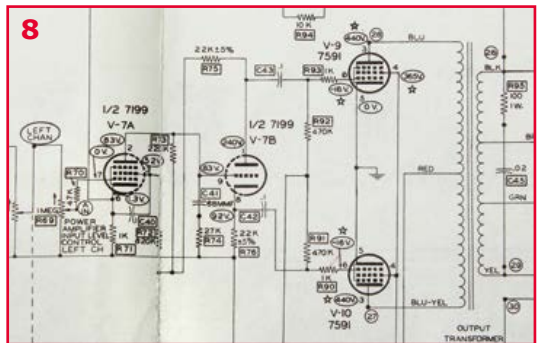
Web Links

[1] Heathkit Museum: www.heathkit-museum.com/hifi/hvmaa-100.shtml

[2] AA-100 schematic: www.vintage-radio.info/heathkit/

[3] 22RH427: www.oudio.nl/speakers/22rh427.htm

[4] 16 HorsePower live @ Montreux 2010: http://youtu.be/G5pV2aQdf3o

Hexadoku The Original Elektorized Sudoku

Research indicates that conundrums and puzzles keep the brain active. That alone should encourage you to partici-pate in this month's Hexadoku. 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 December 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 September 2014 Hexadoku is: **E80F4**.
The €50 / £40 / $70 book vouchers have been awarded to: Ulf Claesson (Sweden),
Larry Burns (Canada), A. van Maris (Netherlands), Jairo Rotava (Brasil) and Zvi Herman (Israel).
Congratulations everyone!

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

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

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

Say What?

By **Gerard Fonte** (USA)

Speech generation has been around for decades and works quite well. DARPA subsidized research in speech recognition in the early 1970's and even with the astounding advances in computing hardware and software, there hasn't been much success to talk about. Apple has been promoting speech recognition for some time. But Siri's performance is still quite limited. It fails with accents and head colds. And if you've ever read an e-mail translated from voice-mail, you know that speech recognition still has a long way to go. But why? Isn't it just decoding sound?

Running On

First, speech is not like text at all. It's not separate words with a single font. Speechisallruntogether. Separating the sounds into words, or parsing, is not at all trivial. With this example it is easy to see that the middle could be parsed as "his all runt". These are perfectly good English words. Of course, you say that parsing should start from the beginning. So, is the first word "speech" or "speeches"? How do you know? The unfortunate answer is you can't know until you parse the whole sentence. So, in order to know any part of the sentence, you have to know the whole thing. (Which is a recurring problem in Artificial Intelligence or AI.) One method is to parse it into all possible sentences and then work backwards into something that makes sense. But in order for it to make sense to a computer, the computer has to know what "makes sense" actually is. Now you have to teach the computer the rules of grammar and, at the least, define words as nouns, verbs, etc. Things get complicated real fast. (You can simplify speech recognition considerably by having the speaker pause after every word. But that's cheating.)

Human Nature

That's not the way I hear speech, you say. Which is very true. Humans hear speech as discreet words and translate the sounds into meanings from start to finish. There are only about 31 distinct and meaningful sounds in the English language (called phonemes). On the surface it seems like it should be easy to identify just 31 sounds. Except that doesn't work either. Humans perceive many very different sounds as the same phoneme. We hear the same word regardless if it's spoken by a man or woman or child. We hear it when it's distorted by the telephone or with a lot of background noise, if it's sung or chanted, accented or slurred. (Stick those into an FFT and try to have your software try to find the identity.) It's only when we stop and think about the actual sounds do we realize that a woman has a higher vocal pitch than a man. This pitch has no effect on the meaning. Quite simply, the sounds we perceive as speech can be much more different than they are alike. How we easily accomplish this seemingly impossible task is usually ascribed to "pattern recog-

nition". And it is true that there are patterns to speech. But these patterns are also variable according to the speaker. If they were genuine, fixed patterns it would be an easy thing for a computer to find and decode. Human (and animal) pattern recognition is an incredibly powerful mechanism for dealing with the world around us. It apparently integrates top-down analysis with bottom-up synthesis. How this is actually accomplished is probably the biggest question in AI. It seems clear that the speech recognition systems currently in use are not at all related to how people do it. (It's the same for chess programs and other AI systems.)

Signal to Noise

Information theory says that for any transmitter-receiver system there must be a minimum signal-to-noise ratio (S/N) in order to function. The noise can be arbitrarily divided between the receiver and transmitter. With humans, the noise is apparently almost all in the transmitter. As noted above, we understand speech under wildly varying conditions—which corresponds to lots of noise. This means that the receiver must be nearly noise free. It turns out that humans apply all sorts of rules and cues that have nothing directly related to speech in order to reduce the noise. Even before someone starts talking, his or her expression and body language tell us what to expect. The conversation is different in a church than it is in a bar. The age, sex and health of the speaker also give us information. Sometimes the most important things are communicated with silence. In short, our whole life experience can be applied to understanding speech.

AI

Speech recognition is one small aspect of AI. One of the original goals of AI was to understand how humans thought. It became apparent that to try to duplicate human cognition, one had to duplicate a human history as well as a human physiology. That is, to think like a person you had to experience being a person. And, as noted previously, you had to process the whole in order to understand any part. These obstacles pushed AI into emergent-behavior topics like bird-flocking. Modern AI turns the original goal inside out. Instead of trying to mimic human thought, the new goals are to create behaviors that duplicate living systems. Little attempt is made to consider how living systems actually process information. The idea is that similar behaviors may have similar processes. This assumption may not be valid. So, these AI systems may be better described as Alien Intelligence rather than Artificial Intelligence.

(140357)
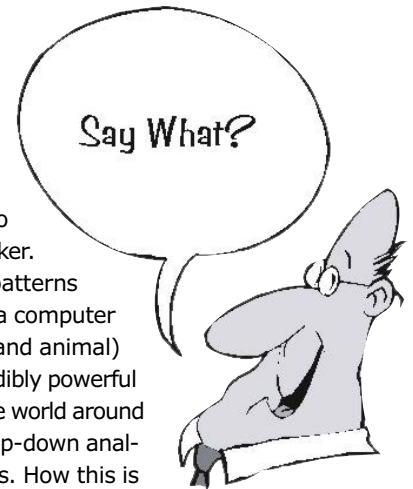
MUST-HAVE
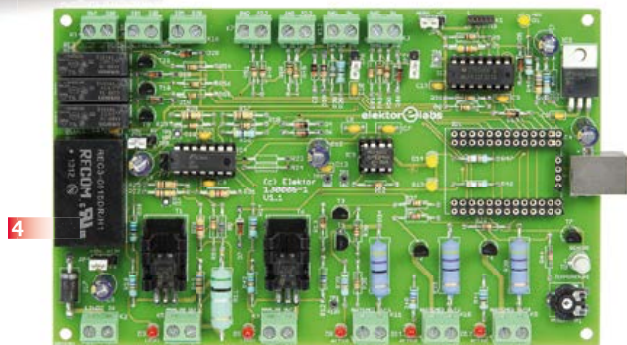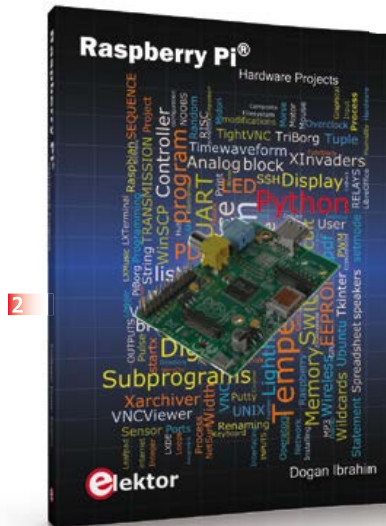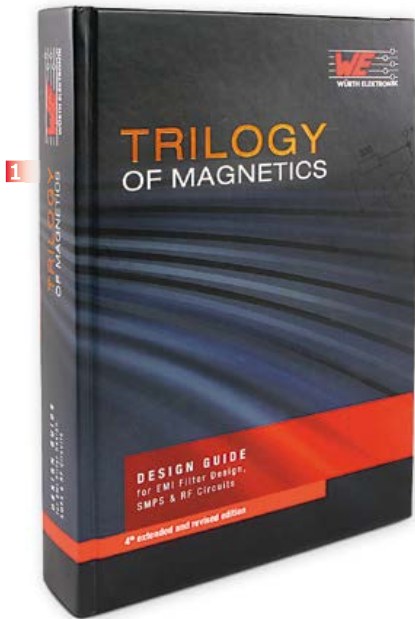
15% DISCOUNT
for GREEN and GOLD Members!
www.elektor.com/trilogy-of-magnetics

The Authority on Magnetics and Inductors

1 Trilogy of Magnetics

A standard work in the hands of 15,000 design engineers, academics and researchers worldwide, Trilogy of Magnetics aims at familiarizing users with the characteristics and applications of inductive components. The Trilogy comprises: Basic Principles, Components, and Applications. Highlights include calculation, dimensioning and construction of customer-specific transformers, an introduction to frequency compensation, Class-D amplifier design and basic principles of Ethernet & Power-over-Ethernet. Furthermore Trilogy of Magnetics includes over 200 practical examples of filter circuits, audio & video circuits, interfaces, RF circuits, motor control units and SMPS. The book closes off with a keyword index and a summary of formulas.

728 pages • ISBN 978-3-89929-157-5
£42.95 • € 49.00 • US $67.00

The RPi in Control Applications

2 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.

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

110 Elektor Editions, Over 2500 Articles

3 DVD Elektor 2000 through 2009

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

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

Measurement and Control Using Your PC

4 IO-Warrior Expension Board

Don't throw out your old PCs and notebooks or leave them gathering dust in the basement! They can be a useful resource: by adding this universal interface card an old PC can be pressed into service as a measurement and control hub. An IO-Warrior module on the I/F board takes care of USB communication, and source code is available that works with the free version of Visual Studio.

Ready-built IO-Warrior56 module
Art.# 130006-91
£34.95 • € 39.95 • $54.00

Fun to Build and Use Projects

5 Create 30 PIC Microcontroller Projects with Flowcode 6

This book covers the use of Flowcode® version 6, a state-of-the-art, all-graphical based code development tool, for the purpose of developing PIC microcontroller applications at speed and with unprecedented ease. Without exception, the 30 projects in the book are fun to build and use. A secret doorbell, a youth deterrent, GPS tracking, persistence of vision (POV), and an Internet webserver are just a few examples of projects in the book waiting to be explored and mastered. This makes the publication
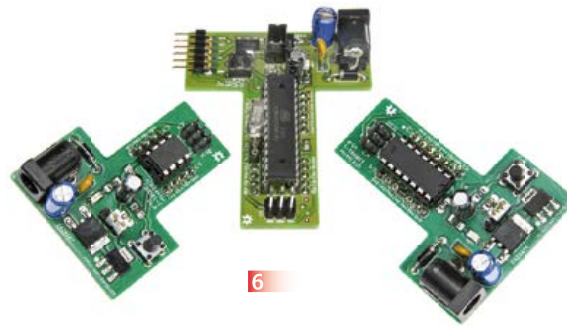
5

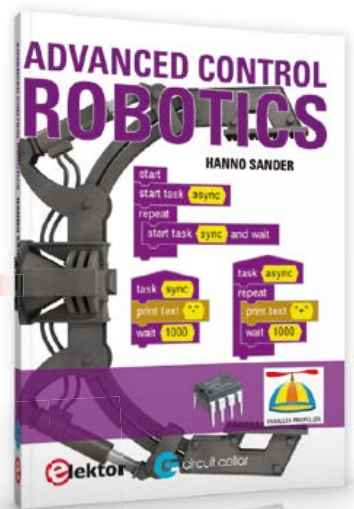**Also Available as E-book**

6

7

8

9

a perfect source of projects constantly challenging your hardware and software skills as you progress, resulting in advanced microcontroller applications you can be proud of. All sources referred in the book are available for free download, including the support software.

226 pages • ISBN 978-1-907920-30-1
£30.95 • € 34.95 • US $48.00

Three Sizes for Cheap & Fast AVR Prototyping
6   T-Boards

In response to the limitations posed by fixed-design, Arduino-style boards, Elektor has designed three mini-development boards that give developers more flexibility while still hosting the microcontroller and its supporting components. We're very excited to present the T-Boards! T-Boards are breadboard-friendly PCBs designed for simple and swift microcontroller proto-typing using an Atmel ATmega328, ATtiny24-44-84 and ATtiny25-45-85 microcontrollers. Additionally, each T-Boards has an integrated 3.3V and 5V-selectable power supply, which assists in reducing the number of required jumper wires and allows for experimenting with lower power usage. Programming the microcontroller can be done in-circuit (ICSP).

Bundle of three boards
Art.# 130581-94
£33.95 • € 39.00 • US $53.00

Advanced Robot Technologies Made Easy
7   Advanced Control Robotics

It doesn't matter if you're building a line-following robot toy or tasked with designing a mobile system for an extraterrestrial exploratory mission: the more you know about advanced robotics technologies, the better you'll fare at your workbench. Hanno Sander's Advanced Control Robotics is intended to help roboticists of various skill levels take their designs to the next level with microcontrollers and the know-how to implement them effectively. Advanced Control Robotics simplifies the theory and best practices of advanced robot technologies. You're taught basic embedded design theory and presented handy code samples, essential schematics, and valuable design tips (from construction to debugging).

160 pages • ISBN 978-0-96301-333-0
£34.95 • € 39.95 • US $54.00

Build Your Own Robot
8   ActivityBot

Learn real-world engineering skills with the friendly, capable, and peppy ActivityBot from Parallax. It's a great option for first-time robot-builders, as well as for an intro to technology and engineering courses in high schools and colleges. Step-by-step web tutorials

take you through programming its multicore Propeller chip in C, wiring circuits on a breadboard, and build-ing sensor systems so your robot can navigate on its own. Following the checkmarks gets you to the fun fast, with optional links for added learning.

Art.# 140191-71
£173.95 • € 199.90 • US $270.00

Nostalgia: The Fascination of 40 Years Ago
9   DIY VHF Retro Radio

For anyone enthusiastic about Very High Frequency (VHF) radio, this all-in-one kit from our partner FRAN-ZIS is ideal. This modern VHF radio in a stylish, retro body receives FM stations in the 87.5 MHz to 108 MHz band with good reception performance. You will mainly hear the powerful local stations in high sound quality. However, the sensitivity of the receiver also allows you to listen to remote stations at times. This radio assem-bly kit contains all required parts, including casing, printed circuit board, rod antenna and all necessary components and can be assembled easily and enjoy-ably. High quality components, high sensitivity and a rod antenna provide the best radio reception possible.

All-in-one kit
Art.# 140260-91
£26.95 • € 29.95 • US $41.00
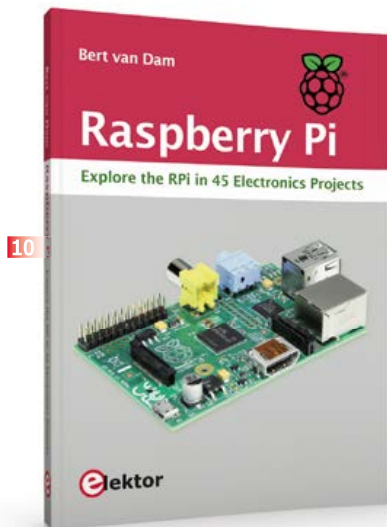
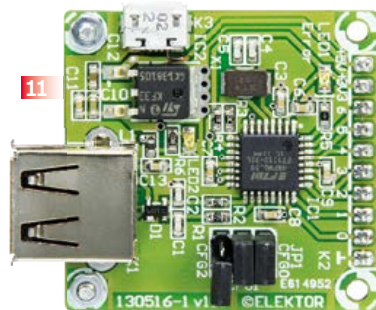**Explore the RPi in 45 Electronics Projects**
10 Raspberry Pi

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

288 pages • ISBN 978-1-907920-27-1
£34.95 • € 39.95 • US $56.40

Android User Interface Builder
11 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

Build Your Own Bat Detector
12 Do-it-yourself Bat Detector

Who says you can't hear bats? With this proper gizmo, you can–and much more! Build your own high-quality ultrasonic bat detector and experience these sounds for yourself! This kit for home assembly makes it an easy and fool-proof process. The printed circuit board comes with many SMD components premounted. All you have to do is solder in a few parts and connect the circuit board to the microphone, the loudspeaker and the controls. State of the art integrated circuits ensure high sensitivity and volume. Once you have assembled the bat detector, you can start exploring right away!

All-in-one-kit
Art.# 140259-91
£26.95 • € 29.95 • US $41.00

**A Small Basic Approach**
13 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

MIFARE and Contactless Cards in Application
14 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
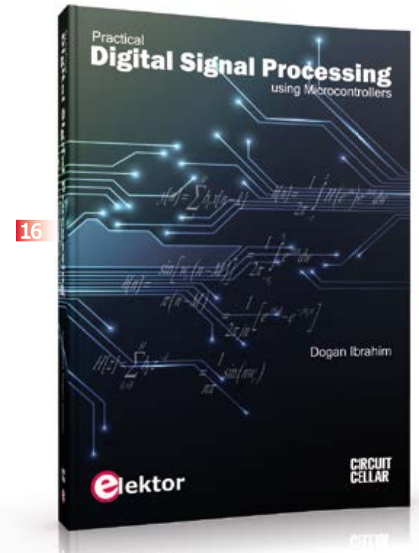
antenna design, security considerations and cryptography. The complete design of a reader's hardware and software is described in detail. The reader's firmware and the associated PC software support programming using any .NET language. The specially developed PC program, "Smart Card Magic.NET", is a 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

LEDs, Buttons, Display and More
15 Arduino Extension Shield

This shield intended to augment the Arduino Uno offers starters a text display, LEDs and pushbuttons to provide a good basis for their projects. For experienced users two extension connectors are provided which can be used to connect relay modules, wireless modules and many other devices. This shield thus ameliorates the Arduino Uno's most significant shortcomings on on-board peripherals.

Ready-built module
Art.# 140009-91
£23.95 • € 26.95 • US $37.00

Ideal Reading For Students and Engineers
16 Practical Digital Signal Processing using Microcontrollers

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

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

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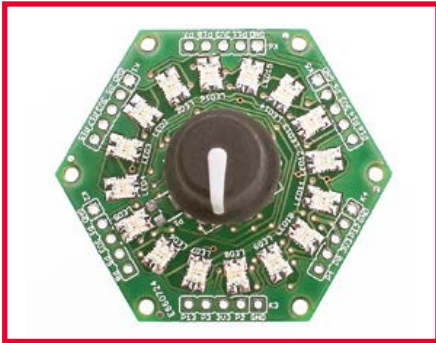
UK / ROW

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

USA / CANADA

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

## NEXT MONTH IN ELEKTOR MAGAZINE

COol COntroller COncept

For many devices one or two buttons suffice for their control. With this in mind we designed a hexagonal little board, several of which can easily be fitted behind the front plate of a DIY project.

A rotary encoder or pushbutton serves as a control, with a ring of 16 two-color LEDs indicating the value or position.

Display Board for VariLab 402

A display board provides simple and straightforward operation of the VariLab 402 PSU described in this issue. On a 4-line LCD various measured and calculated values are shown, the operation being done with two rotary encoders and a pushbutton. The circuit is designed such that it's easily dropped into other measurement and control circuits.
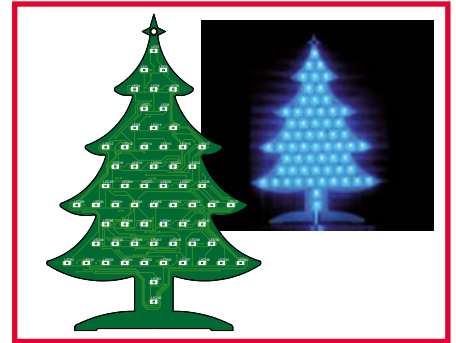
Programmable Christmas Tree

No December edition of Elektor is complete without a Christmas Tree project. This year, in cooperation with Eurocircuits we developed a Christmas tree with 62 LEDs in a matrix encapsulated in PCB material. The flexible tree is special in being customer programmable, meaning you design your own light effects and patterns. Connect the tree to your PC and enter the LED patterns in a browser by mouse clicks.

Article titles and magazine contents subject to change, please check www.elektor-magazine.com for updates.
Elektor's December 2014 edition is processed for mailing to Gold Members starting November 18, 2014.

Please note: as of the October 2014 edition Elektor magazine is not available from bookshops, newsstands and kiosks. Readers not having an Elektor membership can purchase printed or digital copies of individual magazines directly from the publishers at www.elektor.com (click on MAGAZINES).

See what's brewing @ Elektor Labs 24/7

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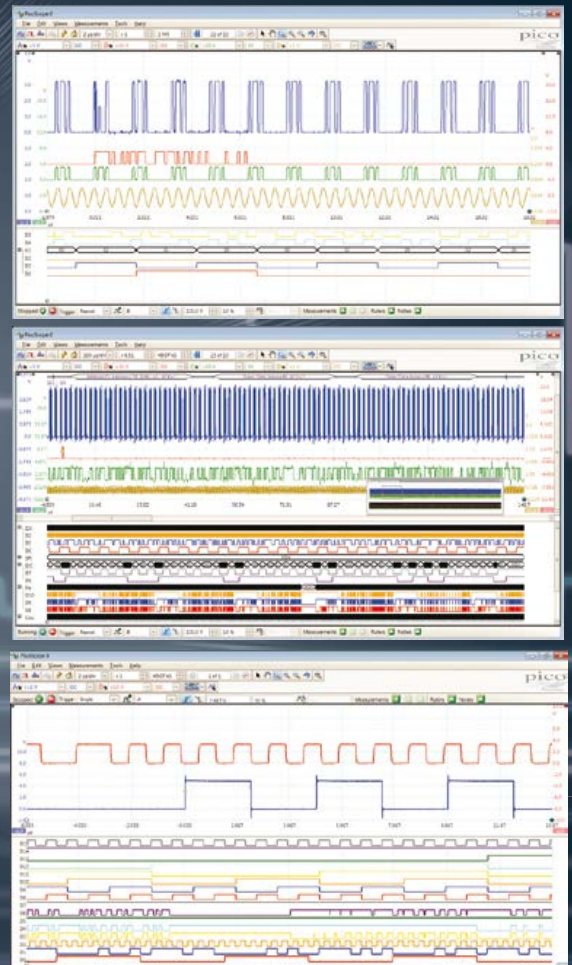# www.elektor-labs.com
and join, share, participate!

Embedded
3D Gesture and Position Tracking

with Microchip's GestIC® Technology

Microchip's patented GestIC® 3D Technology provides a **single-chip**, real-time gesture recognition solution. The MGC3130 enables the next breakthrough in user interface design for any product at a very low cost point.

MGC3130

- The MGC3130 runs the Colibri Gesture Suite, enabling robust recognition of complex 3D gestures without requiring any host processing

- The MGC3130 delivers the results (detected gestures/ XYZ positioning) to the host via an I²C™ bus interface or using GPIOs

Approach Wake-up   Air Wheel   Flick Gestures   Sensor Touch   Position Tracking

microchip DIRECT
www.microchipdirect.com

microchip.com/gestic