

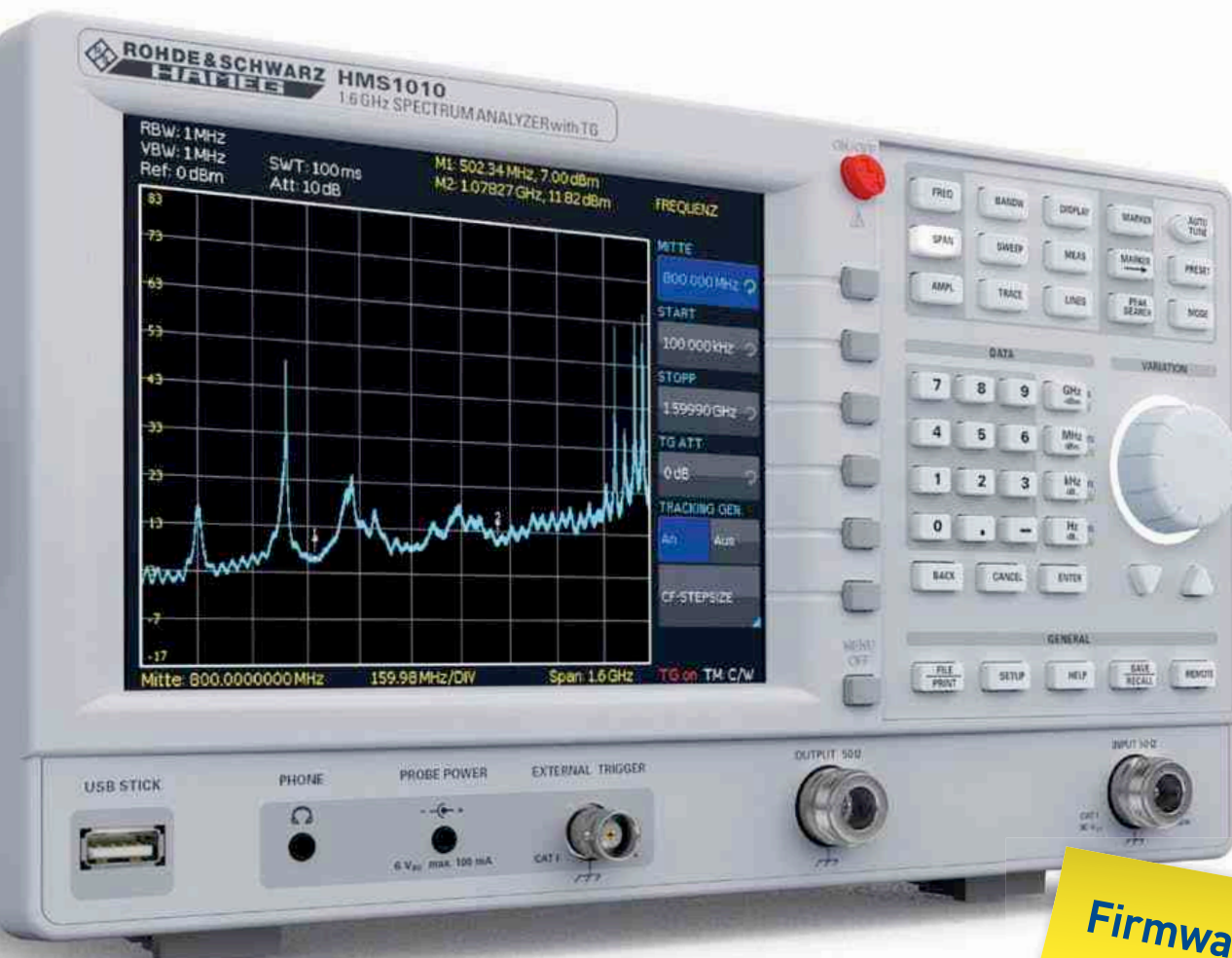
elektor

EMBEDDED FIRMWARE LIBRARY

Schnell zur eigenen Mikrocontroller-Anwendung

I/O LEDs LCD
ADC PWM SD MUX TCP/IP

- Einführung in Python | **Prototyping-Tool** | Dreieck-Generator
- FPGA-Simulation** | Menüs mit Tastern und Display
- BaroStick ● **Neues aus dem Labor**
- **Retronik** ● Rund um die Elektronikwelt



HAMEG Instruments hat den Frequenzbereich bei allen Spektrumanalysatoren der 1000er Serie von 1 GHz auf 1,6 GHz erhöht.

Somit erhalten HAMEG Kunden beim Kauf eines HMS1000, HMS1010 und auch beim HMS1000E ab sofort 60% mehr Frequenzbereich ohne Aufpreis.

+60%

MEHR FREQUENZBEREICH

HMS1010 | HMS1000 | HMS1000E

von 1GHz auf 1,6GHz

NEU!!
Firmware-Upgrade
unter www.hameg.com
kostenfrei verfügbar





DAS ORIGINAL SEIT 1994

PCB-Pool

Beta LAYOUT

Der neue Standard

ENIG statt Chemisch Zinn für Ihre Leiterplatte

Gold!

Ohne Aufpreis!

Hochwertigste Oberfläche: ENIG

Multilayer

Preise bis zu

50%

 gesenkt

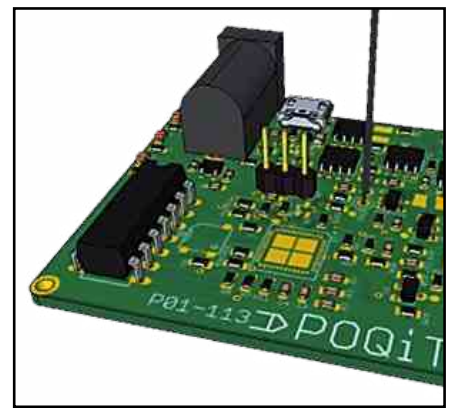
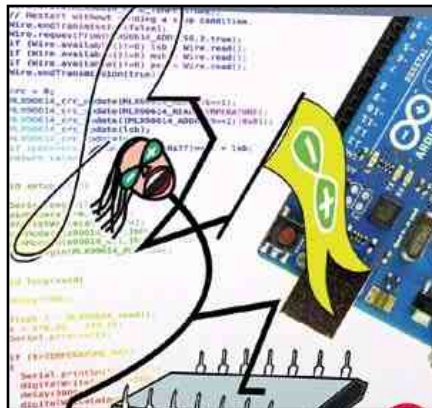
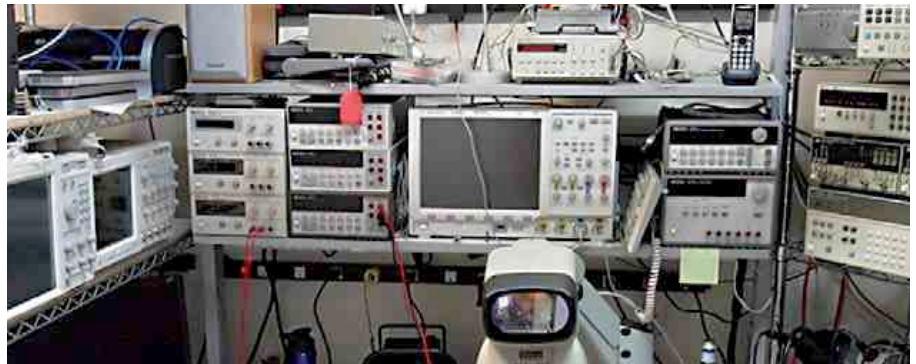
bei 2-5 Multilayer-
Leiterplatten

www.pcb-pool.com

Beta

LAYOUT

create : electronics



Community

- 6 Impressum
- 8 Elektor World
Neues aus der Elektor-Community
- 10 Aktuell
Neue Produkte

Industry

- 38 Tools für
Vektor-Motorsteuerungen
Hier wird gezeigt, wie Software dabei hilft, BLDC-Motoren effizient, präzise und leise bei einstellbaren Geschwindigkeiten zu betreiben.

Labs

- 42 Sorry about my English
Interessante Projekte auf der .Labs-Website
- 44 Fallstricke in der
Platinenproduktion
- 46 **Keramikkondensatoren**
voll daneben!

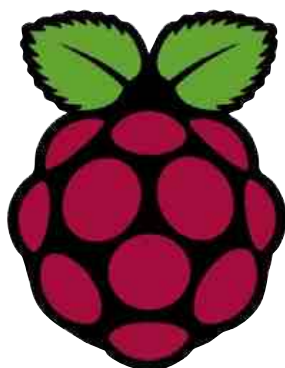
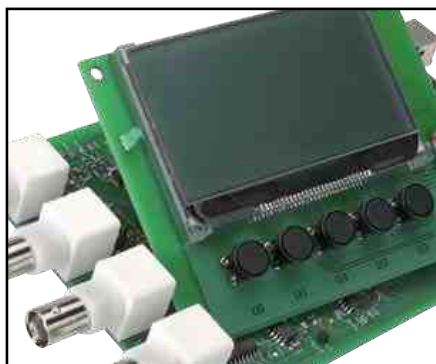
Projects

- 12 **Embedded Firmware Library**
Die modular aufgebaute „EFL“ ist nicht nur eine weitere Bibliothek für das „Rapid Application Development“. Eigene Anwendungen und Module lassen sich dabei hardware-unabhängig programmieren, so

dass sie leicht von einem Board auf ein anderes portiert werden können. Dabei ist man nicht auf einen Controllerhersteller beschränkt; und sogar ein Betrieb „auf“ Embedded Linux ist prinzipiell möglich. Dank diverser Funktionen für ADC, PWM, Display, TCP/IP, SD-Karten und vielem mehr eignet sich die Bibliothek auch für (C-)Einsteiger.

20 Barostick

Module in USB-Stick-Bauform gibt es für viele Anwendungen, von WLAN und Bluetooth bis zum Daten-Logging. Das gleiche handliche Format hat der Luftdruck- und Temperaturmesser Barostick, der die Umweltgrößen Luftdruck und Temperatur mit einem Sensor von Bosch-Sensortec misst. Ein Programm, das unter Windows auf dem PC läuft, stellt die Messwerte grafisch auf dem Bildschirm dar.



26 LCR-Meter 2013 (3)

In die dritte und letzte Runde geht unser LCR-Meter mit folgenden Themen: Platinenlayout und Stückliste, Schaltungsaufbau und Gehäuse, Inbetriebnahme und Einstellungen.

34 PICO PROto

Wie testet man schnell einen neuen Sensor? Und wie probiert man eine neue Idee aus, ohne jeweils ein komplexes und teures Entwicklungs-Board konfigurieren zu müssen? Hierfür wurde PICO PROto entwickelt, ein minimalistisches Prototyping-Tool...

48 Von BASIC nach Python (1)

Die Sprache Python eignet sich gut zur hardwarenahen Programmierung kleiner Rechnersysteme wie z.B. Raspberry Pi.

56 Bauen Sie Ihren Chip! (4)

Auf dem Weg zum einsatzfähigen FPGA ist die Simulation eine wich-

tige Zwischenstation. In diesem Teil der Serie zeigen wir, wie die Anwendung aus der letzten Folge simuliert werden kann.

65 OTA-basierter Dreieck-Generator

Viele Anwendungen von OTAs wurden schon beschrieben, eine Basisschaltung zur Erzeugung eines nicht-sinusoidalen Signals dürfte aber schwer zu finden sein. Dieser Beitrag beschreibt einen kombinierten Dreieck/Rechteckgenerator, der auf zwei OTAs (Operational Transconductance Amplifier) beruht.

68 Flexible Bedienung

Auch auf einem kleinen Display lassen sich viele Parameter und Einstelloptionen eines Projekts anzeigen, wenn man geschickt eine Menüführung programmiert. Voraussetzung für eine bequeme

Bedienung sind Taster, die sinnvoll neben dem Display angeordnet sind. Nun benötigt man nur noch passenden Code.

72 Interview

Raspberry Pi: Ein Jahr und eine Million Computer später...

● Magazine

76 Hexadoku

Sudoku für Elektroniker

78 Retronik

Der Fluch des Sammelns

82 Vorschau

Nächsten Monat in Elektor

Impressum

44. Jahrgang, Nr. 509 Mai 2013
Erscheinungsweise: 10 x jährlich
(inkl. Doppelhefte Januar/Februar und Juli/August)

Verlag

Elektor-Verlag GmbH
Süsterfeldstraße 25
52072 Aachen
Tel. 02 41/88 909-0
Fax 02 41/88 909-77

Technische Fragen bitten wir per E-Mail an
redaktion@elektor.de zu richten.

Anzeigen (verantwortlich):

Irmgard Ditgens
ID Medienservice
Tel. 05 11/61 65 95-0 | Fax 05 11/61 65 95-55
E-Mail: service@id-medien-service.de
Es gilt die Anzeigenpreisliste Nr. 43 ab 01.01.2013

Vertriebsgesellschaft:

IPS Pressevertrieb GmbH
Postfach 12 11, 53334 Meckenheim
Tel. 0 22 25/88 01-0 | Fax 0 22 25/88 01-199
E-Mail: elektor@ips-pressevertrieb.de
Internet: www.ips-pressevertrieb.de

Vertrieb Österreich

Pressegroßvertrieb Salzburg/Anif
Niederalm 300
Tel. +43/62 46/37 21-0

Der Herausgeber ist nicht verpflichtet, unverlangt eingesandte Manuskripte oder Geräte zurückzusenden. Auch wird für diese Gegenstände keine Haftung übernommen. Nimmt der Herausgeber einen Beitrag zur Veröffentlichung an, so erwirbt er gleichzeitig das Nachdruckrecht für alle ausländischen Ausgaben inklusive Lizenzen. Die in dieser Zeitschrift veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen einschließlich Platinen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Die veröffentlichten Schaltungen können unter Patent- oder Gebrauchsmusterschutz stehen. Herstellen, Feilhalten, Inverkehrbringen und gewerblicher Gebrauch der Beiträge sind nur mit Zustimmung des Verlages und ggf. des Schutzrechtsinhabers zulässig. Nur der private Gebrauch ist frei. Bei den benutzten Warenbezeichnungen kann es sich um geschützte Warenzeichen handeln, die nur mit Zustimmung ihrer Inhaber warenzeichengemäß benutzt werden dürfen. Die geltenden gesetzlichen Bestimmungen hinsichtlich Bau, Erwerb und Betrieb von Sende- und Empfangseinrichtungen und der elektrischen Sicherheit sind unbedingt zu beachten. Eine Haftung des Herausgebers für die Richtigkeit und Brauchbarkeit der veröffentlichten Schaltungen und sonstigen Anordnungen sowie für die Richtigkeit des technischen Inhalts der veröffentlichten Aufsätze und sonstigen Beiträge ist ausgeschlossen.

© 2013 elektor international media b.v.
Druck: Senefelder Misset, Doetinchem (NL)
ISSN 0932-5468

Lang oder kurz?

Wenn Sie dieses Heft durchblättern, könnte Ihnen der stark unterschiedliche Umfang der Artikel auffallen. Einige sind recht lang geraten - acht bis zehn Seiten sind nicht ungewöhnlich. Gleichzeitig hat auch die Zahl der Serien in unserem Heft etwas zugenommen. Sicherlich liegt das in der Natur der Sache. Wenn wir komplexe Themen wie die Entwicklung von FPGA- oder Embedded-Linux-Anwendungen auf wenigen Seiten abhandeln würden, hätten nur die Spezialisten unter den Elektor-Lesern etwas davon. Und ein Projekt wie unser LCR-Meter enthält so viele clever gemachte Details, so dass wir ihm ebenfalls unrecht tun würden, wenn wir so etwas auf vier bis sechs Seiten komprimierten. Das Titelprojekt in dieser Ausgabe ist ein weiterer dicker Kaliber. Mit der Bibliothek lässt sich Firmware prinzipiell unabhängig vom Controller-Typ und dem verwendeten Board programmieren. Auf acht Seiten können wir in dieser Ausgabe nur eine Einführung geben, die vermutlich weder die Einsteiger noch die Fortgeschrittenen richtig satt macht. Wir haben uns deshalb entschlossen, viele wichtige Hintergrundinformationen auf ein Zusatz-Dokument auszulagern, das man von unserer Website herunterladen kann. In den nächsten Ausgaben werden Sie dann noch einige Anwendungen der „Embedded Firmware Library“ finden, so dass sich Stück für Stück ein Bild ergeben dürfte, wie man mit dem System gewinnbringend arbeiten kann.

Natürlich sprechen wir auch auf unseren Redaktionskonferenzen darüber, wie wir solch komplexe Artikel am besten unseren Lesern näherbringen. Schätzen diese lieber einen überlangen Artikel, eine Serie oder die angesprochene Kombination aus einer Veröffentlichung im Heft und im Web? Schreiben Sie uns unter redaktion@elektor.de!

Jens Nickel



Unser Team

Chefredakteur:	Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)
Ständige Mitarbeiter:	Dr. Thomas Scherer, Rolf Gerstendorf, Klaus Boda
Internationale Redaktion:	Harry Baggen, Thijs Beckers, Jan Buiting, Eduardo Corral, Wisse Hettinga, Denis Meyer, Clemens Valens
Elektor-Labor:	Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser
Grafik & Layout:	Giel Dols, Mart Schroijen



Germany

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



United Kingdom

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



Netherlands

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



France

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



USA

Hugo Van haecke
+1 860-875-2199
h.vanhaecke@elektor.com



Spain

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



Italy

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



Sweden

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



Brazil

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



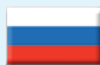
Portugal

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



India

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



Russia

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



Turkey

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



South Africa

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



China

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

Unser Netzwerk



VOICE COIL

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR IMMEDIATE ELECTRONICS ENGINEERING INFORMATION



audioXpress



Die Elektor-Community

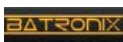


Unsere Partner und Sponsoren



AudioXpress

www.audioamateur.com 77



Batronix

www.batronix.com/go/23 67



Beta Layout

www.pcb-pool.com 3



Distrelec

www.distrelec.de 11



Eurocircuits

www.elektorpcbservice.com 63



HAMEG

www.hameg.com 2



Intelligent Soc s.l.

www.solutions.com 41



LeitOn

www.leiton.de 41



Linx Technologies

www.linxtechnologies.com 25



National Instruments

www.ni.com/multisim/d 23



Pico

www.picotech.com/PS208 25



Reichelt

www.reichelt.de 84



Schaeffer AG

www.schaeffer-ag.de 47

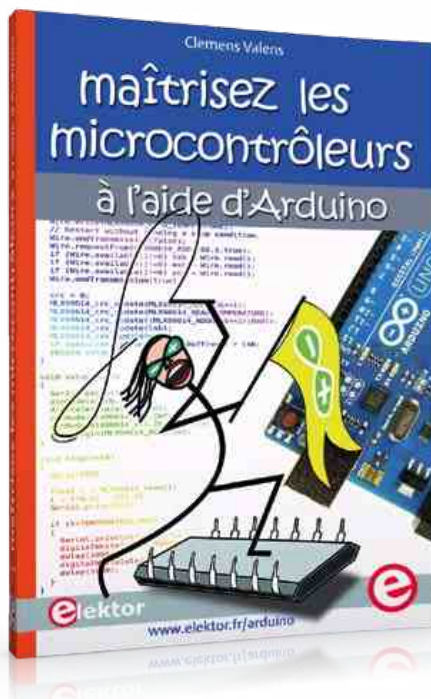
Sie möchten Partner werden?

Kontaktieren Sie uns bitte unter service@id-medienservice.de (Tel. 0511/616595-0).

Elektor World

Zusammengestellt von
Wisse Hettinga

Jeden Tag, jede Stunde, jede Minute, ja, in jedem Augenblick schaffen, optimieren, rekonstruieren und entwickeln Ingenieure und Enthusiasten neue Elektronik. Hauptsächlich zum Spaß, aber manchmal wird aus Spaß (beruflicher) Ernst.



■ ■ Schwing die Peitsche!

Wenn Sie unseren Abenteuern folgen, wissen Sie, dass Elektor-Mitarbeiter nicht an nur einem Ort arbeiten. Die meisten von uns arbeiten in Holland, aber wir haben auch Kollegen in Spanien, den Vereinigten Staaten, Italien, Indien, Frankreich - es ist eine lange Liste. Clemens Valens arbeitet von Frankreich aus und verbindet seine Pflichten als Manager der Elektor-Labs-Website mit ... Schreiben! In seinen ruhigeren Stunden hat er an einem Buch über Arduino gearbeitet. In dem Buch erklärt er, wie man die Arduino-Plattform verwenden kann, um seiner Nachbarschaft auf die Nerven zu fallen, während man lernt, mit dieser interessanten Hardware-Plattform umzugehen. Zugegeben, als wir das Buch sahen, überraschte uns das Cover mit einem Mikrocontroller (so weit, so gut) und einer (männlichen oder weiblichen) Figur mit einer Peitsche. Vermutlich, um den Controller richtig anzutreiben. Also, das sieht wirklich wie ein interessantes Buch aus und wir beschlossen, es in Deutsch und Englisch zu übersetzen. Aber die Peitsche!? Was sollen wir uns hierbei bloß denken?

🇬🇧 Die Rückkehr des Tapirs

Letztes Jahr haben wir ein kleines E-Smog-Meter namens Tapir vorgestellt. Die Konstruktion war einfach und bemerkenswert mit Leiterplatten als Gehäuse. Dieses Jahr bekamen wir die Bestätigung von Eurocircuits, dass sie weitere 1.000 Stück dieser netten Platinen produzieren werden. Das E-Smog-Messgerät kann Ihnen helfen, die Welt des Elektromagnetismus zu entdecken. Es fängt die Streusignale verschiedenster Quellen auf und verwandelt sie in Audiosignale. Mit einem simplen Kopfhörer hört man die versteckten Sounds von Smartphones oder Tablets. Die Platinen und der Bausatz des Tapirs werden Ende Mai 2013 wieder im Elektor-Webshop verfügbar sein. Verfolgen Sie dies unter www.elektor.com/Tapir.



In der Höhle des Löwen

Es gibt sie auf der ganzen Welt - Dachböden, Keller, Garagen, kleine Zimmer, angefüllt mit elektronischen Geräten. Sie sind die Schatzkammern der Elektronik! Viele unserer Leser haben an solchen Orten viele Stunden gearbeitet, nachgedacht und neue Anwendungen entwickelt. Vergessen Sie die High-Tech-Labors mit Millionen-Euro-Sponsorverträgen - denn nicht dort wohnt der wahre Geist der Elektronikentwicklung. Unser Schwester-Magazin *Circuit Cellar* (aha!) zeigt Fotos solcher Höhlen Elektronik-Besessener. Hier sehen Sie den Arbeitsplatz von Vincent Himpe, einem Entwickler-Pionier und Autor der Elektor-Buchreihe *LabWorX*.



Elektor-Labor in Mumbai eröffnet

Jahr für Jahr benötigen wir rund 200 Projekte für unsere Publikationen. Traditionell haben wir ein Team von Ingenieuren, das mit den Elektor-Redakteuren ortsnah Hand in Hand arbeitet. Aber die wachsende Zahl von Projekten verlangt mehr Kapazität. Mit einer neuen Elektor-Labs-Abteilung in der Lamington Road in Mumbai (ehemals Bombay), mitten im Herzen des größten indischen Elektronikmarkts, haben wir hier investiert. Die Lamington Road ist mit mehr Elektronik-Läden vollgestopft, als Sie sich erträumen können. Wenn unsere Mumbai-Entwickler irgendein Bauteil benötigen, gehen sie einfach in den Shop nebenan oder auf der anderen Straßenseite - was für ein großartiger Ort, um zu arbeiten! Das Elektor-Labs-Team ist hier auf dem Foto zu sehen: Von links nach rechts Krishna Chandran, Sunil Malekar, Clemens Valens, Nandini Singh und Shreenivas G M Shree.



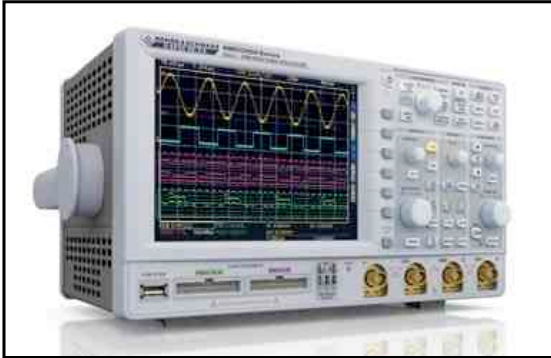
Spark your Design!

Elektor und RS Components arbeiten ab sofort in punkto *DesignSpark* zusammen. Zuerst geht es hier um das DesignSpark-Magazin. Dies ist der neue Titel der bekannten Zeitschrift *eTech* und wird der gesamten Kundschaft von RS Components zugesandt. Die neue Zeitschrift berichtet von den neuesten Entwicklungen auf neuen Hardware-Plattformen und wird auch Elektor-Projekte enthalten. Weiterhin geht es um die Einführung einer speziellen Design-and-Build-Sektion auf der Webseite von *DesignSpark.com*. Die Projektplatinen werden mit DesignSpark PCB, einer kostenlosen CAD-Software (www.designspark.com) geroutet. In Elektor und auf unserer Website www.elektor-labs.com wird es hierzu spezielle Projekt-Beiträge geben.

(130051)

Neue MSOs von HAMEG

Die neue Serie HMO3000 von HAMEG Instruments besteht aus sechs neuen Modellvarianten in den Bandbreiten von 300 MHz bis 500 MHz und mit 2 oder 4 Kanälen. Durch die Funktionalität eines Mixed-Signal-Oszilloskops können zusätzlich zu den Analogkanälen bis zu 16 Digitalkanäle



analysiert werden.

Signifikantester Unterschied zur Vorgängerreihe ist neben der erhöhten Bandbreite die mit 8 MB verdoppelte Speichertiefe – jetzt sowohl manuell als auch automatisch einstellbar. Neu ist auch, dass alle bisherigen Modelle der Serie HMO3000 mit 300 MHz bzw. 400 MHz bei Bedarf über ein Bandbreiten-Upgrade auf

500 MHz erweitert werden können:

300-MHz-Modelle mit Option H00352 (2-Kanal) oder H00354 (4-Kanal),

400-MHz-Modelle mit Option H00452 (2-Kanal) oder H00454 (4-Kanal).

Als Einführungs-Angebot erhalten Käufer der neuen Mixed-Signal-Oszilloskope den H03508-Logik-Tastkopf ohne Aufpreis dazu. Diese Aktion läuft bis zum 31. Oktober 2013.

www.hameg.de

Preiswerter LED-Strahler

Reichelt hat einen LED-Flutlichtstrahler mit einer Anschlussleistung von 10 W im Programm, der nur 18,95 € kostet. Das ist für einen kompletten Scheinwerfer für den Außeneinsatz sehr preisgünstig. Zusammen mit einem Bewegungsmelder kann man damit zu geringen Kosten zum Beispiel den Eingangsbereich eines Hauses beleuchten. Der Strahler vom Typ HEIT 37009 bringt es immerhin auf eine Helligkeit von 870 lm, was in etwa der Helligkeit einer 75-W-Glühbirne entspricht. Gegenüber dieser ist die LED-Technik aber bezüglich Lebensdauer (ca. 30.000 h) und der Anzahl an Schaltzyklen weit überlegen, was für Installationen, bei denen man zum Auswechseln des Leuchtmittels eine Leiter oder mehr benötigt, besonders relevant ist. Den aus Aluminium gefertigten LED-Strahler gibt es in warmweiß mit einer Farbtemperatur von 3.000 K und einer Anschlussspannung von 230 V. Er ist für den gleichen Preis auch in kaltweiß mit 6.700 K zu haben. Außerdem gibt es auch noch Ausführungen in 20, 30, 50, 70 und 100 W.

www.reichelt.de

Micro-WLAN-Geräteserver

Lantronix stellt sein neues WLAN-Modul xPico vor, bei dem es sich um einen kompakten, Embedded-WLAN-Geräteserver handelt, der schnelle Verbindungen zwischen seriellen Anschlüssen (bis zu 921 kbit/s) und WLAN ermöglicht. Dabei muss keine Software entwickelt und keinerlei Code geschrieben werden. xPico misst nur 24 mm x 16,5 mm und ist das neueste Exemplar in der xPico-Produktfamilie. Muster sind schon zu haben und im Laufe des ersten Halbjahrs 2013 sind auch größere Stückzahlen zu Preisen ab rund 25 US-Dollar erhältlich.

xPico bietet Wi-Fi-Funktionen auf kleinstem Raum, einen vollständigen IP-Stack plus erweiterte Sicherheit und ist daher kosteneffektiv und robust.

www.lantronix.com



Neuer Steckverbinderkatalog

Auf 464 Seiten zeigt der neue Katalog von W+P Steckverbinder ab einem Rastermaß von 0,5 mm, darunter erstmalig wasserdichte Ausführungen. Das Artikelprogramm besteht aus RJ-45, D-SUB- und USB-Steckverbindern, Kabelkonfektionen und Zubehör. Weitere Produkte, wie z.B. IEEE 1394 (FireWire),

DVI-Steckverbinder und Rundsteckverbinder sind auf Anfrage auch als wasserdichte Ausführung erhältlich. W+P bietet auch kundenspezifische Produkte an.

Erweitert wurde das Produktspektrum um zusätzliche Miniaturisierungslösungen mit geringen Bauhöhen und kleinen Rastermaßen im Stift- und Buchsenleistenbereich. Der Anteil an rationell bestückbaren und automatengerecht verpackten Artikeln wächst zunehmend. Hierzu werden Tape-and-Reel-Verpackungen mit Bestückungshilfen angeboten, die leicht in bestehende Abläufe integriert werden können.

Der Katalog ist in deutscher und englischer Sprache gehalten und kann kostenlos angefordert werden. Alle Produkte sind jeweils mit Foto, Maßzeichnung, Leiterplattenlayout, technischen Daten und Polzahlen abgebildet.

<http://wppro.com>



Experimentier-Board für Raspberry Pi

Das Gertboard ist ein besonders interessantes Add-on für Raspberry Pi, das via Farnell/element14 erhältlich ist (Bild: Vorführung des Gertboards am Messestand auf der Embedded World). Das von Gert van Loo entwickelte Experimentier-Board kann direkt

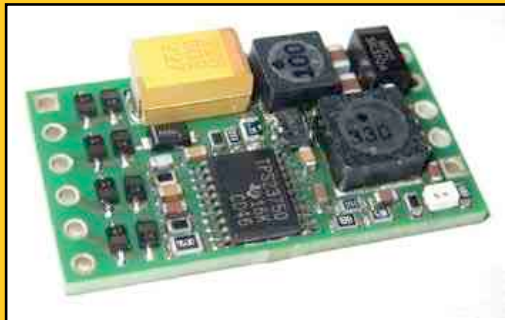
an Raspberry Pi gesteckt werden und ermöglicht diesem den Zugriff auf die physikalische Welt. Analoge Spannungen lassen sich erfassen und generieren. Hinzu kommt eine Steuerung für Motoren, LEDs und Relais sowie die Registrierung von Tastendrücken. Mit Jumper-Kabeln kommt man an alle Stellen der Platine und erhält so maximale Flexibilität. Das Board kann direkt auf den GPIO-Sockel von Raspberry Pi gesteckt werden und ermöglicht so die Steuerung von Motoren bis zu 18 V/2 A. Integriert sind ein zweikanaliger D/A-Konverter und ein zweikanaliger A/D-Konverter mit jeweils 8-bit-Auflösung sowie ein ATmega328 mit 10-bit-ADC. Programme können auf dem Raspberry Pi geschrieben und entwickelt sowie dann auf das Gertboard übertragen werden. Das Add-on-Board verfügt darüber hinaus über sechs Open-Collector-Ausgänge, 12 LEDs und 3 Taster.

www.farnell.com

PoE für Embedded-Anwendungen

Mit dem CrumbX1-NET Ethernet-Modul bietet die chip45 GmbH eine Plattform für Embedded-Anwendungen und ergänzt diese nun durch eine kompakte Power-Over-Ethernet-Stromversorgung Power-PoE-1. Durch PoE reduziert sich der Verkabelungsaufwand für ein Ethernet-Gerät auf ein Netzkabel. Dabei wird die Versorgungsspannung von einem PoE-Switch oder -Injektor über Ethernet-Datenleitungen oder ungenutzte Aderpaare des Ethernet-Kabels übertragen.

Der angeschlossene Verbraucher benötigt einen PD-Controller, der die Gegenstelle beim Power-Up über seine Leistungsklasse informiert. Außerdem ist in der Regel ein Schaltregler erforderlich, der die hohe PoE-Spannung von 36...57 V auf eine gängige Versorgungsspannung von 5 V oder 3,3 V bringt.



Diese komplette Funktionalität ist im kompakten Steckmodul Power-PoE-1 mit Abmessungen von nur 28 x 15 mm integriert. Es kann z.B. direkt auf das Embedded Ethernet-Modul CrumbX1-NET von chip45 aufgesteckt werden und stellt als Kombination wohl eine der kleinsten PoE-versorgten Ethernet-Lösungen dar. Selbstverständlich kann das PoE-Modul auch alleine verwendet werden.

Das PoE-Modul hat die PoE-Klasse 1 und liefert eine Ausgangsspannung von 5 V bei einem maximalen Dauerstrom von 500 mA. Optional kann die Ausgangsspannung auch auf 3,3 V eingestellt werden. Anschluss-Pins im Rastermaß 1/10" vereinfachen die Verwendung des Moduls auf Lochrasterplatinen oder lötfreien Steckboards.

www.chip45.com

APPKÜRZUNG

ZU ÜBER
500.000
PRODUKTEN



HEUTE AUSSUCHEN, MORGEN AUSPACKEN!



Ob Onlineshop oder Katalog:

Wir liefern deutschlandweit innerhalb von 24 h – ohne Mindermengenzuschlag. Lieferung ab 1 Stück. Die Distrelec-Gruppe: Ihr Partner für elektronische Bauelemente, Automation, industrielle IT und Zubehör.

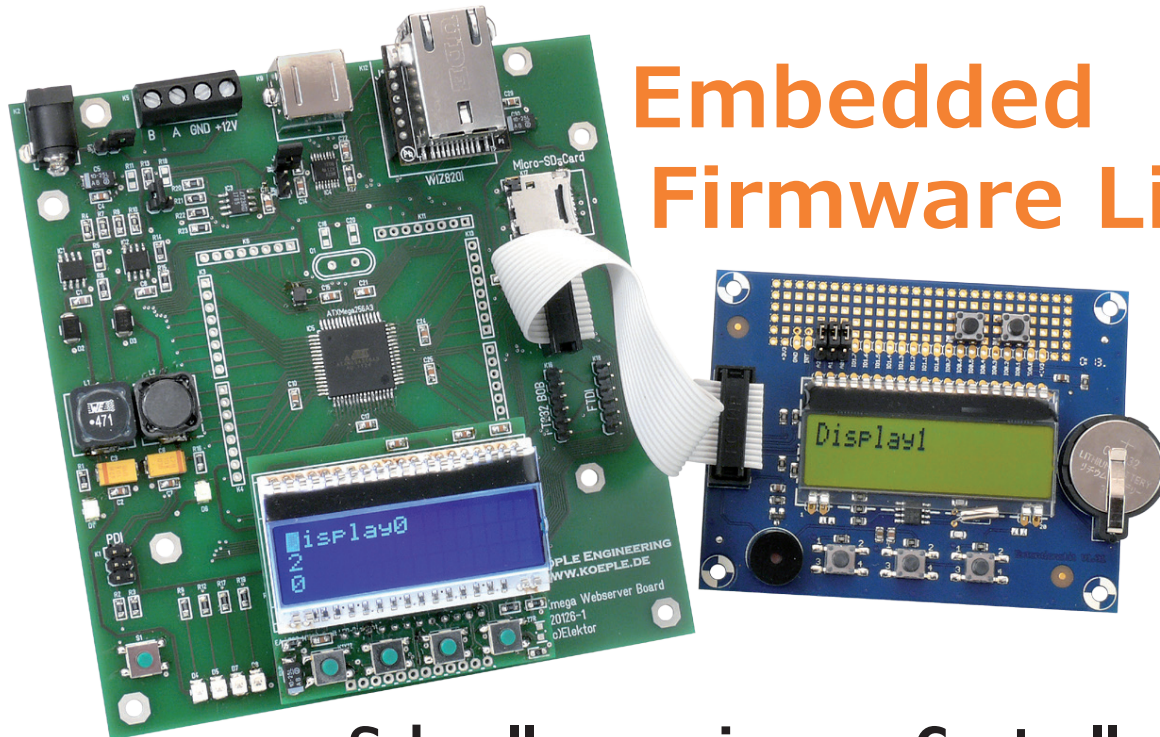
WWW.DISTRELEC.DE

Bestellhotline 0180 5223435*

*14 Ct./Min. aus dem Festnetz der Dt. Telekom AG, Mobilfunk kann abweichen



DISTRELEC
A Datwyler Company



Embedded Firmware Library

Schnell zum eigenen Controller-Projekt!

Von **Jens Nickel**

Die modular aufgebaute „EFL“ ist nicht nur eine weitere Bibliothek für das „Rapid Application Development“. Eigene Anwendungen und Module lassen sich dabei hardwareunabhängig programmieren, so dass sie leicht von einem Board auf ein anderes portiert werden können. Dabei ist man nicht auf einen Controllerhersteller beschränkt; und sogar ein Betrieb „auf“ Embedded Linux ist prinzipiell möglich. Dank diverser Funktionen für ADC, PWM, Display, TCP/IP, SD-Karten und vielem mehr eignet sich die Bibliothek auch für (C-)Einsteiger.

Eigenschaften der Embedded Firmware Library

- Prinzipiell an alle Controller anpassbar, für die ein ANSI-C-Compiler besteht
- Empfehlung ab 32 KB Flash und 2 KB RAM
- Gut dokumentierte API
- Bisher unterstützte Controller: ATmega328/324/644 und Xmega256A3. Weitere in Vorbereitung
- Bisher unterstützte Boards: ElektorBus-Experimentalknoten, Xmega-Webserver-Board [9], Linux-Extension-Board [8], Arduino Uno Rev.3. Weitere Boards in Vorbereitung!
- Einfach an andere Boards anpassbar
- Bisher verfügbare höhere Libraries: LEDButton, ADCMux, StepperMotor, ElektorBus, Display, WizNet-TCP/IP-Modul, SD-Karte (Rohdaten)
- Open-Source-Lizenz LGPL

Im Herbst hatten wir eine C-Bibliothek für Elektor-Bus-Mikrocontroller-Knoten vorgestellt [1]. Eines der Kennzeichen war die Trennung in einen hardwareabhängigen Teil sowie ein Codemodul, das für die Protokolle zuständig ist.

So ist es möglich, die Bibliothek schnell an andere Controllertypen und Boards anzupassen – man muss nur den Hardware-Layer austauschen, die eigentliche ElektorBus-Bibliothek bleibt immer gleich.

In einer typischen Bus-Anwendung müssen aber nicht nur Nachrichten codiert und über RS485 versandt und empfangen werden.

Es werden auch LEDs und Relais bedient, analoge (Sensor-)Werte eingelesen und Motoren angesteuert. Darüber hinaus wäre es schön, wenn der Controller auf einfache Weise mit anderen

Chips kommunizieren könnte, etwa zur Anbindung eines TCP/IP-Moduls oder einer SD-Karte.

Zusammen programmiert

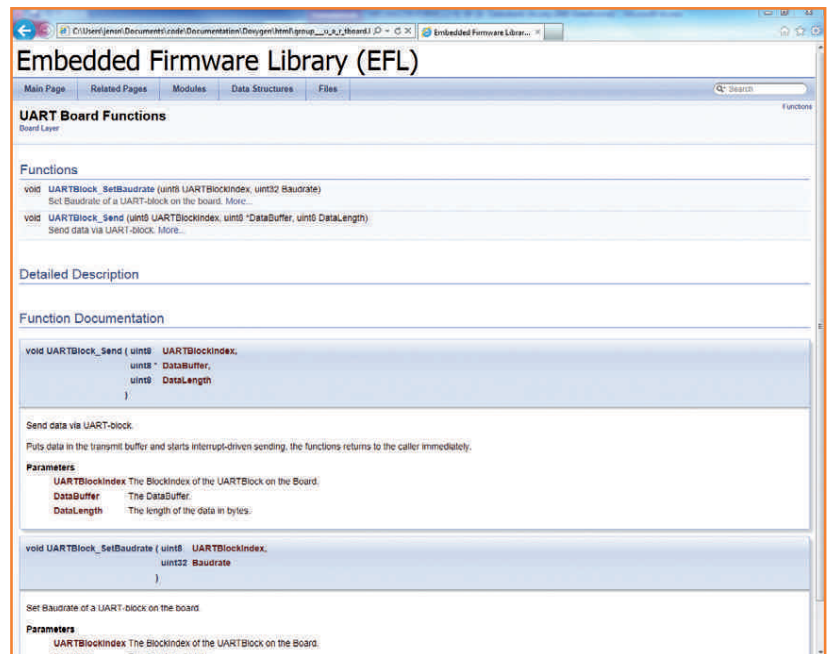
Beim Programmieren kamen weitere Ideen, und so wurde das Ganze Schritt für Schritt flexibler und umfassender gemacht – weit über das ElektorBus-Projekt hinaus. Ich hatte dabei Glück, dass ich mit unserem Leser Michael Busser (einem Mitglied der ElektorBus-Google-Group) einen Mit-Entwickler fand, der clevere Codemodule, Funktionen und Ideen beisteuerte.

Um in freien Stunden zusammen am Code arbeiten zu können, legten wir auf der berühmten Projekt-Plattform „SourceForge“ [2] einen Eintrag namens „Embedded Firmware Library (EFL)“ an. SourceForge ist für Open-Source-Projekte reserviert, was ja gut zu Elektor passt; als Lizenz haben wir die LGPL gewählt [3]. Das wichtigste Feature der Projekt-Plattform ist ein integriertes, SVN-kompatibles Versionsverwaltungssystem [4], das man unter Windows zum Beispiel mit TortoiseSVN [5] bedienen kann. Ein Mausklick genügt, und man kann seine Änderungen hochladen („Commit“) oder die neueste Version des Codes herunterladen („Update“).

Da mit der Zeit eine stattliche Sammlung von Funktionen zusammenkam, haben wir auch zum ersten Mal mit einem automatischen Dokumentationssystem gearbeitet. Wenn man im Code Kommentare mit einer bestimmten Notation hinterlässt, kann das Open-Source-Programm Doxygen [6] automatisch eine übersichtliche Dokumentation aller Funktionen erstellen, welche auf HTML basiert und in einem Browser angezeigt werden kann (**Bild 1**). Wirklich eine tolle Sache! Die Doxygen-Dokumentation der EFL kann man auf der Website zu diesem Projekt [7] herunterladen.

Baukasten-System

Die Embedded Firmware Library ist als Baukasten-System konzipiert, einzelne Module einer Schicht („Layer“) lassen sich dabei gegeneinander austauschen. Es gibt drei unterschiedliche Hardware-Ebenen: Ganz unten ist der Controller-Code angesiedelt, dann kommt der Board-Layer und schließlich ein Extension-Layer (siehe **Bild 2**). Das entspricht den Verhältnissen eines realen Projekt-Designs. Verschiedene (pinkompatible) Controller können ja auf ein- und demselben Board zum Einsatz kommen, ein Controller aber auch auf verschiedenen Boards. Ein schönes



Beispiel hierfür ist der ATmega328, er ist auf verschiedenen Elektor-Platinen, aber auch auf dem Arduino-Uno-Board zu finden.

Bild 1.
Das Tool „Doxygen“ sorgt für eine übersichtliche Dokumentation der Funktionen.

Wenn man einen Erweiterungsstecker mit einer bestimmten Steckerbelegung spezifiziert, lassen sich auch verschiedene Controllerboards und Erweiterungsboards kombinieren. Ein Beispiel hierfür ist das Linux-Extension-Board in diesem Heft [8]. Es passt nicht nur an das Elektor-Linux-Board, sondern auch an das Xmega-Webserver-

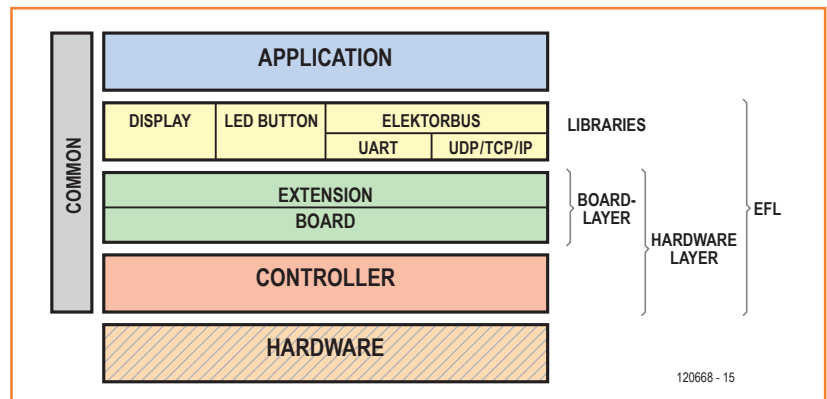


Bild 2.
Dank sauber getrennter Hardware-Layer ist ein EFL-Projekt schnell an andere Plattformen anpassbar. Sehr nützlich ist das, wenn beim Prototyping auch die Hardware noch weiterentwickelt werden muss.

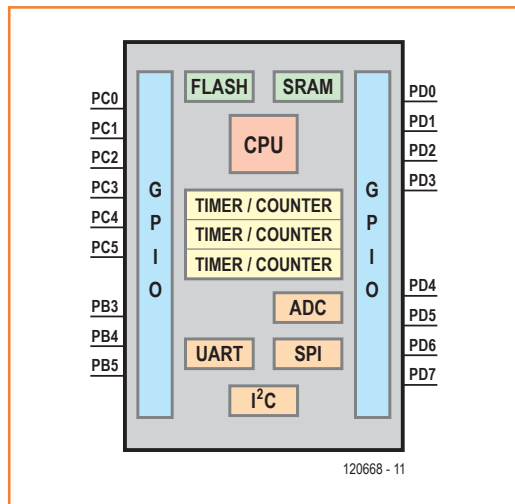


Bild 3.
Stark vereinfachtes
Blockschaltbild eines
Controllers am Beispiel
ATmega328.

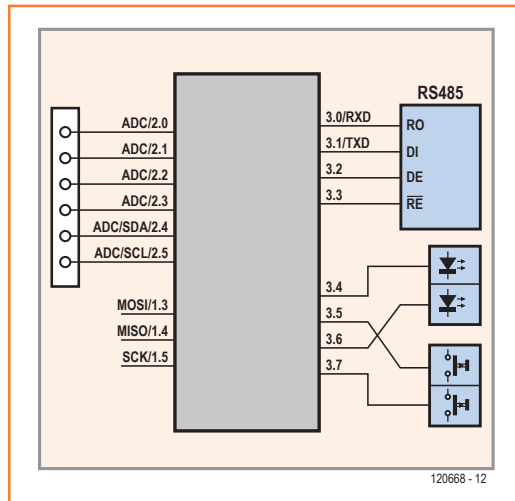


Bild 4.
Für den Board-Code-
Entwickler ist der Controller
eine Blackbox, er muss nur
die Anschlüsse (Portpins)
kennen.

Board [9], das wir im Juni vorstellen werden (großes Bild). Ein Mother- oder Baseboard mit diverser Peripherie, auf das verschiedene Prozessorboards aufgesteckt werden können, ist eine weitere Anwendung der EFL.

Für den Controller, das Controllerboard und das Extensionboard gibt es drei unterschiedliche Codemodule, die jeweils aus einem Headerfile .h und einem Codefile .c bestehen. Man muss also jeweils die passenden Files ControllerEFL.h/.c, BoardEFL.h/.c und ExtensionEFL.h/.c in sein Projekt einbinden. Die Filenamen sind für alle Controller und Boards, die von der EFL unterstützt werden, gleich. Damit es hier nicht zu Namenskonflikten kommt, sind die Files in der unter [11] downloadbaren Codebasis in Unterordnern untergebracht, die den Namen des jeweiligen Control-

lers beziehungsweise Boards tragen. Mehr hierzu findet man in einem Extra-Dokument zur EFL [7].

Das Blackbox-Prinzip

Im Idealfall können alle drei Hardware-Codemodule von drei verschiedenen Entwicklern programmiert werden. Klar ist: Der Controller-Code-Entwickler kann dabei nicht wissen, auf welchem Board sein Controller zum Einsatz kommt, und der Board-Entwickler nicht alle Extensions-Boards, die es geben wird, voraussehen.

Doch auch umgekehrt (von oben nach unten) gilt, dass ein Entwickler sich nicht mit Einzelheiten und dem intern verwendeten Code der nächsten Schicht beschäftigen muss. Der Board-Code-Entwickler benötigt nur den Schaltplan des Boards, auf dem der Controller mit seinen Anschlüssen (Portpins) und deren Funktion verzeichnet ist. Mit den Registern und Details im Inneren des Controllers (**Bild 3**) muss er sich nicht auskennen; der Controller ist für ihn eine „Black Box“ (**Bild 4**). Ähnliches gilt für den Entwickler des Extension-Boards. Ihn interessiert nur die Pinbelegung des Erweiterungssteckers, über den sein Board mit dem Controllerboard und damit auch mit dem Controller Kontakt aufnimmt. Aber die Verdrahtung auf dem Controllerboard (also welche Pins des Steckers an welche Portpins des Controllers angeschlossen werden) muss er nicht kennen (**Bild 5**).

Höhere Bibliotheken

Über der Hardwareschicht folgen Module, die höhere Bibliotheksfunktionen zur Verfügung stellen (in der EFL-Sprache „Libraries“ genannt). Als Beispiele enthält die Codebasis bereits ein Modul zum Ansteuern eines HD44780-kompatiblen Text-Displays, einen Treiber für ein TCP/IP-Modul von WizNet, eine kleine Schrittmotor-Ansteuerung sowie ein Modul zum Lesen und Schreiben von Rohdaten auf eine SD-Karte. Das Blackbox-Prinzip wird hier fortgesetzt. Wer die Display-Library ändern oder erweitern möchte, kann das Display (oder besser gesagt den Display-Controller) als „Block“ im Blockdiagramm des Boards betrachten (**Bild 6**). Er muss nicht wissen, welche Anschlüsse des Displays mit welchen Portpins des Controllers verbunden sind. Dem Display-Library-Entwickler kann sogar egal sein, ob das Display im 4-bit-Modus oder per SPI angesteuert wird. Er kann sich also ganz auf seine Aufgabe konzentrieren, die entsprechenden HD44780-Kommandos und Daten zusammenzustellen.

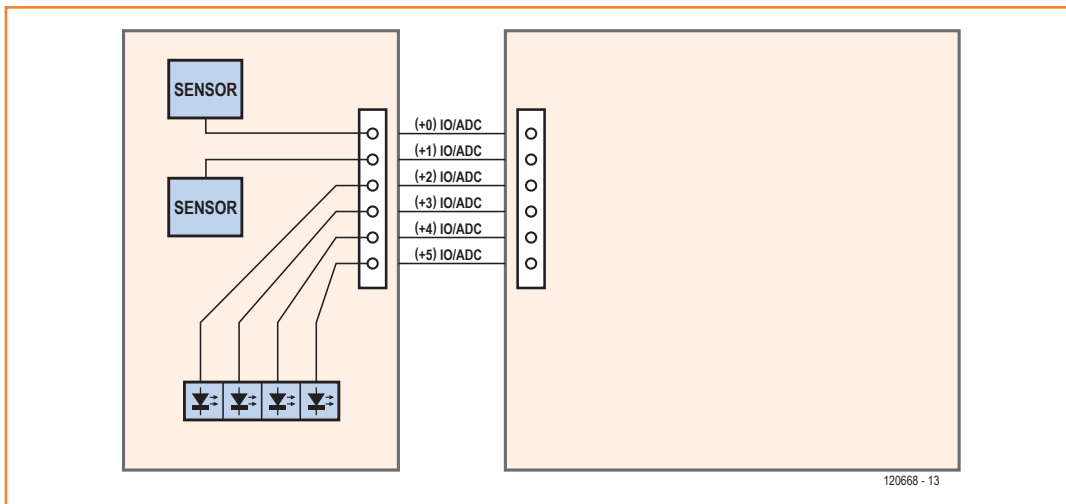


Bild 5.
Der Entwickler des Extensionboard-Codes muss sich nur für die Anschlüsse am Erweiterungsstecker und deren mögliche Verwendung interessieren.

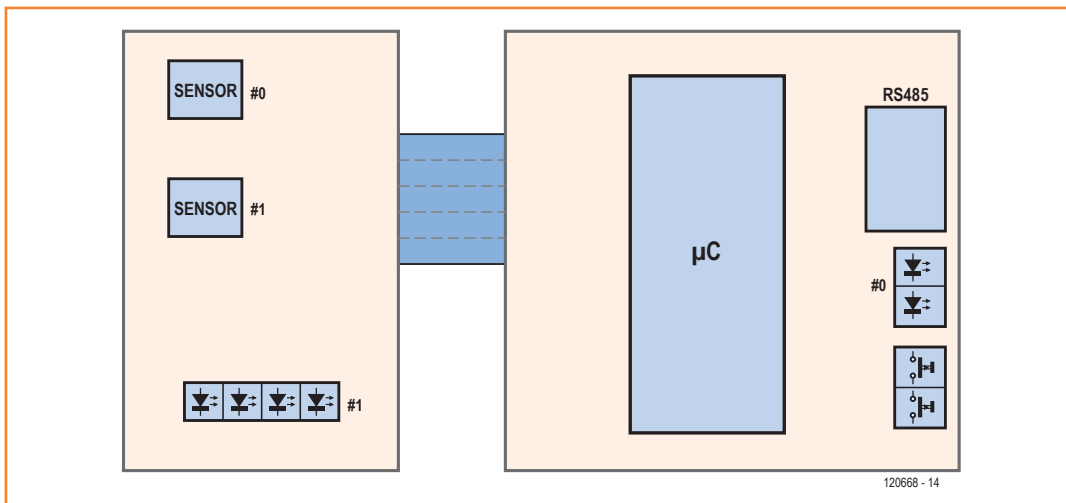


Bild 6.
Der Applikationsentwickler benötigt nur ein Blockdiagramm, die Verdrahtung auf den Boards muss er nicht kennen. Die beiden LED-Blöcke werden über die Blocknummern #0 und #1 angesprochen.

Applikationscode

Der Applikationsentwickler kann bequem auf alle Blöcke des Boards zugreifen, er muss eigentlich nur wissen, dass diese Funktionseinheiten überhaupt vorhanden sind. Es ist sogar möglich, dass sowohl auf dem Controllerboard als auch dem Extensionboard ein LED-Block vorhanden ist, mit beispielsweise einmal zwei und einmal vier LEDs (siehe Bild 6). Der Applikations-Entwickler kann dann zum Beispiel die erste LED im Block 0 und die dritte LED im Block 1 einschalten, ohne sich darum kümmern zu müssen, auf welchem Board die LEDs sitzen.

LEDs, Taster, Relais und weitere digital gesteuerte Einheiten lassen sich auch dann mit derselben Bibliotheksfunktion bedienen, wenn sie nicht direkt an einen Controllerpin, sondern an einen

Portexpander angeschlossen sind. Auch Taster an analogen Eingängen (wie beispielsweise auf dem Linux-Extension-Board) kann man wie echte „digitale“ Buttons ansprechen (siehe großes Bild).

Einheitliches Funktionsset

Wie das Ganze intern funktioniert, kann hier nur ansatzweise beschrieben werden (siehe Textkasten). Im Extra-Dokument [7] findet man viele weitere Infos zur Hardware-Ebene. Diese sind vor allem dann wichtig, wenn man einen eigenen Board-Layer für ein bisher nicht unterstütztes Board schreiben möchte.

Zentrales Prinzip der ganzen Bibliothek ist jeweils ein einheitliches Set von Funktionen (API), das eine untere Schicht den jeweils höheren Schichten zur Verfügung stellt. Die Controller-API ent-

Tabelle: Controller-Funktionen

(Stand März 2013, ohne Setup- und Parameter-Funktionen, Updates siehe [11])

Funktion	Xmega256A3	ATmega 328/324
Interrupts an/aus	X	X
Wait... (Verzögerung)	X	X
Pin setzen	X	X
Pin lesen	X	X
ADC-Wert einlesen	X	X
ADC-Wandlung starten	X	X
DAC-Wert setzen	X	–
UART Daten senden (interrupt-gesteuert)	X	X
Empfang in Ringbuffer	X	X
SPI-Master (Byte senden/empfangen)	X	X
I2C-Master		
Bytes schreiben/lesen (kombiniert)	X	X
Timer starten/stoppen	X	X
Zählen an einem Pin	X	*
Timergesteuert:		
- Hochzählen	X	X
- Funktionsaufruf	X	X
- Pin toggeln	X	X
- Toggeln mit variabler Geschwindigkeit	X	X
- Frequenz messen	X	*
PWM ausgeben	X	*
Software-PWM	*	*
X implementiert, * geplant, – nicht vorhanden		

hält die immer gleichen Funktionen (vorausgesetzt, sie werden vom Controller unterstützt). Hier handelt es sich zuallererst um Setup-Funktionen zur Vorbereitung der Funktionseinheiten („Units“) des Controllers, wie ADC, SPI, UART, Timer/Counter und mehr. Dazu kommen Funktionen, um Parameter einzustellen oder auszulesen (zum Beispiel die Baudrate bei einer UART-Unit) sowie Funktionen für die eigentliche Ansteuerung (Wert einlesen bei einem ADC-Pin, etwas über ein UART- oder SPI-Interface senden und mehr). Einen kleinen Überblick findet man in der **Tabelle**.

Auf der Board-Ebene (Board- und Extension-Modul) findet man Low-Level-Funktionen für die einzelnen Funktionseinheiten wie Display, Schrittmotortreiber und so weiter. Diese Funktionen sind nötig, weil ja die unterschiedliche Verdrahtung und Adressierung (zum Beispiel 4-bit- oder SPI)

gekapselt werden muss. Die höheren Libraries greifen dann auf diese Low-Level-Funktionen zu und stellen dem eigentlichen Applikationscode wieder ein Set von Funktionen zur Verfügung.

Beispiel: Display

Als Beispiel schauen wir uns die Ansteuerung eines Textdisplays an. Der Applikationscode soll eine kleine Begrüßung in die erste Zeile des Displays schreiben und ruft folgende Funktionen der Display-Library auf:

```
Display_LibrarySetup();
Display_WriteString(0, 0, "Welcome!");
```

Die erste 0 bezeichnet dabei die Nummer des Displays (nach C-scher Art sind alle Angaben in der EFL nullbasiert). Zur Ausführung des Schreibbefehls ruft die Display-Library mehrmals die Funktion

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

auf Board-Ebene auf, die jeweils ein Daten- oder Kommandobyte an den Display-Controller übermittelt.

Im Falle des Xmega-Webserver-Boards ist das Display über SPI angeschlossen, die Funktion ist also so implementiert, dass der Aufruf an

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

auf der Controller-Ebene weitergegeben wird. Außerdem werden die zusätzlichen Eingänge CS und RS am Display bedient.

Software-Download

Im Software-Download von der Website zu diesem Artikel [7] findet man ein Applikationsbeispiel. Es handelt sich um ein ähnliches Projekt wie in [1]. Der ElektorBus-Experimentalknoten [10] wird mit einem Extension-Board gekoppelt, das mit zwei Sensoren und vier zusätzlichen LEDs bestückt ist (**Bild 7**). Die Erweiterung wurde auf einer halben ELEX-Platine aufgebaut, ein LochMaster-File findet man ebenfalls im Projekt-Download. Wer die Software mit dem Atmel Studio 6 öffnet, sieht, dass die Codefiles in bestimmten (virtuellen) Unterordnern untergebracht sind, nämlich Hardware, Common und Libraries (siehe

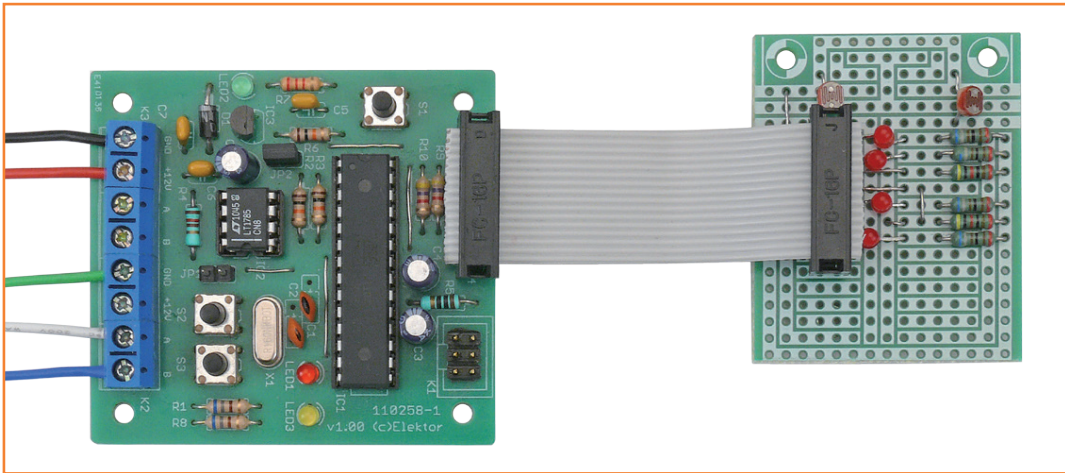


Bild 7.
ElektorBus-
Experimentalknoten mit
Erweiterungsplatine
(zwei Sensoren und vier
LEDs). Der Board-Layer
ist schnell an ein anderes
Board angepasst, wie zum
Beispiel die kommende
Schrittmotorplatine.

Bild 8). Außerdem erkennt man im Hauptfile (ExperimentalNodeEFL.c) das Grundgerüst einer EFL-basierten Anwendung. Unter anderem ist zu sehen, welche Funktionen beim Programmstart aufgerufen werden müssen.

In der ApplicationSetup()-Funktion werden

zuerst die Konfigurationsdaten für den Bus bereitgestellt (ein spezielles Konfigurationsfile ElektorBusNode.h/.c benötigen wir nicht mehr). Danach folgen die Setup-Funktionen für die Library-Module, deren Parameter man im Normalfall nicht verändern muss. Mit SwitchLED(1, 0, 0N); wird die erste LED auf dem Extension-Board

Ein Blick in den „Motorraum“ der EFL

Portpins

Die Digital-I/O-Funktionen (zum Beispiel Pin-Level setzen) im Controller-Code werden mit einem Parameter „Portpin“ aufgerufen, einer Zwei-Byte-Ganzzahl ohne Vorzeichen (uint16):

```
void IO_SetPinLevel(uint16 Portpin, uint8 PinLevel)
```

Das erste Byte von Portpin bezeichnet den Port (die Pingruppe); bei den AVR-Controllern ist dem Port A der Wert 0, dem Port B der Wert 1 usw. zugewiesen. Über diesen Wert rechnet die Funktion dann das richtige Register aus. Das zweite Byte bezeichnet den Pin innerhalb des Ports, bei einem 8-bit-Controller läuft der Wert von 0..7. Die Portpins kann man im Schaltplan mit „0.1“, „0.2“, ... , „1.0“, „1.1“ usw. bezeichnen (siehe Bild 4).

Features, Units und Handles

Die verschiedenen Funktionen, die ein Controller zur Verfügung stellt (ADC, UART, SPI, Timer/Counter usw.), werden in der EFL „Features“ genannt. Von einem Feature wie SPI kann es mehrere „Units“ geben, so besitzt der Xmega256A3 beispielsweise drei SPI-Units. Zu jeder der Units gehören bestimmte Portpins (siehe Datenblatt des jeweiligen

Controllers). Der Board-Code kann die interne Nummerierung der Units und auch die dazugehörigen Register aber nicht kennen. Beim Setup der Unit über eine Funktion auf der Controller-Ebene gibt er stattdessen einen der beteiligten Portpins als Parameter mit, zum Beispiel bei SPI:

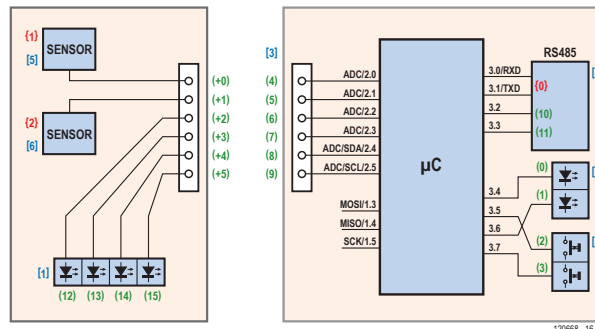
```
int8 SPI_Master_Setup(uint16 Portpin)
```

Die Setup-Funktion im Controller-Code ermittelt nun über den Portpin die Unit-Nummer des gewünschten SPI-Interfaces, mit welcher der Controller die entsprechenden Hardware-SPI-Register berechnen kann. Für spätere Zugriffe wird die Unit-Nummer zusammen mit dem Portpin in einem Array namens „Map“ gespeichert. Danach gibt die SPI_Master_Setup-Funktion den Index dieses Eintrags als „Handle“ (0...) zurück.

Mit dem Handle kann der Board-Code ab jetzt das gewünschte SPI-Interface ansprechen, um ein Byte zu senden und zu empfangen:

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

Genauso funktioniert das auch bei den anderen Features.



Boardpins und Blocks

Die einzelnen Portpins des Controllers sind mit Funktionseinheiten auf dem Board (zum Beispiel Reihen von LEDs oder Buttons, RS485-Treiber) verbunden. Die Portpins für diese verschiedenen „Blöcke“ werden beim Board_Init (im Board-

Code) nacheinander in ein Array eingetragen. Die Indices dieses Arrays heißen Boardpins.

Für den LED-Block in Bild 4 würden beispielsweise die Boardpins (0) und (1) vergeben, für den Button-Block die Boardpins (2) und (3) und für die Pins des Erweiterungssteckers die Boardpins (4)..(9).

Boardpin	Portpin
(0)	3.4
(1)	3.6
(2)	3.5
(3)	3.7
(4)	2.0
(5)	2.1
(6)	2.2
...	
(9)	2.5

Die Extension_Init-Funktion im Extension-Codemodul reserviert für die eigenen LEDs (Bild 5) weitere Boardpins, zum Beispiel (12)..(15). Die zugehörigen Portpins 2.2 bis 2.5 werden dabei indirekt über die Boardpin-Einträge des Erweiterungssteckers (6)..(9) ermittelt:

```
...
(12)  2.2    // = (6)
(13)  2.3    // = (7)
(14)  2.4    // = (8)
(15)  2.5    // = (9)
```

Im Boardpin-Array befinden sich jetzt zwei Blöcke von LED-Boardpins: (0)..(1) und (12)..(15). Diese Zahlen (genauer gesagt der erste Boardpin und die Zahl der LEDs innerhalb des Blocks) werden für jeden Block in ein weiteres Array namens „Block“ eingetragen, zusammen mit dem Typ des Blocks:

BlockIndex	BlockType	First	Count
[0]	BLOCKTYPE_LED	0	2
[1]	BLOCKTYPE_LED	12	4
[2]	BLOCKTYPE_BUTTON	2	2
...			

Will eine LED-Library eine bestimmte LED in einem bestimmten Block ansprechen, ruft sie die Funktion

```
void SwitchDigitalOutput(uint8 BlockIndex, uint8
Position, uint8 ON_OFF)
```

auf Board-Ebene auf.

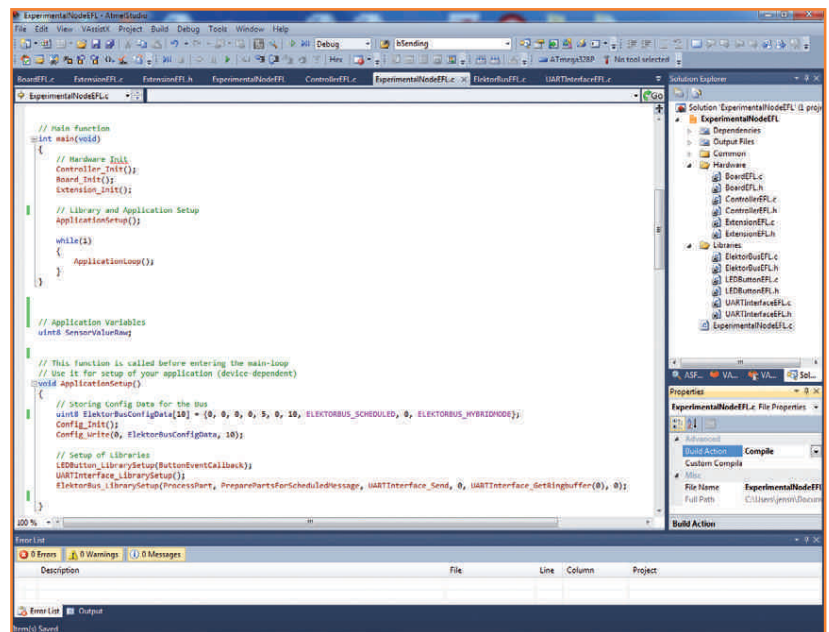
Die Funktion SwitchDigitalOutput(...) ermittelt anhand der Parameter BlockIndex und Position der LED innerhalb des Blocks den entsprechenden Boardpin und darüber dann den Portpin. Im Block-Array wird auch gespeichert, ob eine eingeschaltene LED einen High- oder Low-Pegel verlangt, was ebenfalls von Board zu Board unterschiedlich sein kann.

Ressourcen

Die enorme Flexibilität der EFL hat einen Preis: Neben Flashspeicher für den Programmcode belegt die Bibliothek auch RAM-Speicher (einstellbar wenige bis viele Hundert Byte). Und auch die Performance (die Geschwindigkeit der Programmausführung) sinkt. Letzteres fällt besonders bei einfachen Funktionen wie dem schnellen Toggeln eines Pins ins Gewicht. Daher wurden zu diesem Zweck spezielle Funktionen implementiert, die das timergesteuert erledigen; man muss den Pin nicht fortgesetzt mit einem erneuten Funktionsaufruf „High“ und „Low“ schalten. Doch selbst wenn man dies tun würde, wäre die Performance noch gut ausreichend, um beispielsweise mehrere LEDs zu dimmen.

angeschaltet (zur Kontrolle des Programmstarts). Ein Druck auf den ersten Taster des Experimentalknotens toggelt die rote Test-LED sowie die dritte LED auf dem Erweiterungsboard. Auf Aufforderung des Bus-Schedulers übermittelt der Knoten die Werte der zwei Sensoren zum PC. Die zweite LED auf dem Erweiterungsboard blinkt im Takt dazu; die vierte LED lässt sich vom PC aus schalten.

Das File ExtensionEFL.c enthält übrigens auch gleich den Setup-Code für ein kleines Relaisboard (welcher Code-Abschnitt kompiliert wird, kann man mit dem #define-Ausdruck in ExtensionEFL.h auswählen). Im Projekt-Download findet man auch das erwähnte, ausführlichere Dokument zur EFL und die Dokumentation, die mit Doxygen erstellt wurde. Dazu gibt es dort die gesamte Codebasis inklusive weiterer Libraries (zum Beispiel Display, WizNet-TCP/IP-Modul) als Quellcode.



Ausblick

Dieser Artikel konnte nur eine kurze Einführung sein; speziell für C-Ein- und -Umsteiger ist noch ein Step-by-Step-Artikel geplant. Darüber hinaus wird die EFL auch bei kommenden Elektor-Projekten eingesetzt, zum Beispiel beim Xmega-Webserver-Board.

Vielleicht wird der eine oder andere Leser die EFL nicht nur verwenden, sondern auch erweitern und anpassen wollen (siehe hierzu die Hinweise im ausführlichen Dokument [7]). Wer Vorschläge für Änderungen oder Erweiterungen hat, etwa bei der Controller-API, kann sich gerne beim Autor melden unter redaktion@elektor.de. Natürlich wäre es auch schön, wenn wir Unterstützung bei der Anpassung an weitere Controller, Boards und Peripherie-Blocks bekommen könnten. Auf der Elektor .Labs-Website [11] werden wir regelmäßige Updates veröffentlichen.

(120668)

Weblinks

- [1] www.elektor-magazine.de/120582
- [2] <http://sourceforge.net/projects/embeddedlib/>
- [3] www.gnu.org/licenses/lgpl.html
- [4] <http://subversion.apache.org>
- [5] <http://tortoisesvn.net>
- [6] www.doxygen.org
- [7] www.elektor-magazine.de/120668
- [8] www.elektor-magazine.de/120596
- [9] www.elektor-labs.com/xmegawebserver
- [10] www.elektor-magazine.de/110258
- [11] www.elektor-labs.com/EFL

Bild 8.

Screenshot des Beispiel-Projekts in Atmel Studio 6. Die Codefiles des Projekts sind (virtuell) in bestimmten Unterordnern untergebracht, was die Übersicht erleichtert.

Über RS485, TCP/IP und Funk: ElektorBus und eigene Protokolle

Die Hardware-Unabhängigkeit der ElektorBus-Library wurde weiter verbessert. Das Codemodul `ElektorBusEFL.h/.c` ist künftig nur noch rein für die ElektorBus-Protokolle zuständig. Es ist möglich, damit Nachrichten nicht mehr nur über RS485, sondern auch über RS232, TTL/UART und sogar TCP/IP zu versenden und zu empfangen, was bereits mit dem kommenden Xmega-Webserver-Board und einem TCP/

IP-Modul WIZ820io von WizNet getestet wurde. Die Nutzung von 433-MHz-Funkmodulen wäre eine weitere interessante Möglichkeit. Analog kann man sich natürlich seine eigene Protokoll-Library schreiben, wiederum unabhängig vom verwendeten Übertragungskanal, Board und Controller. Es ergibt sich hier also eine riesige Spielwiese für eigene Experimente und Anwendungen.

Barostick

Luftdruck und Temperatur mit „USB-Stick“ messen



Von
Ruud van Steenis
(NL)

Module in USB-Stick-Bauform gibt es für viele Anwendungen, von WLAN und Bluetooth bis zum Daten-Logging. Das gleiche handliche Format hat der Luftdruck- und Temperaturmesser Barostick, der die Umweltgrößen Luftdruck und Temperatur mit einem Sensor von Bosch-Sensortec misst. Ein Programm, das unter Windows auf dem PC läuft, stellt die Messwerte grafisch auf dem Bildschirm dar. Sogar das Hochladen der Grafiken auf einen FTP-Server ist möglich.

Wenn es in der Welt der PCs um das Speichern und Transportieren von Daten geht, sind USB-Sticks unentbehrlich. Die Vorteile liegen buchstäblich auf der Hand: Die kleinen Speicherriesen passen in jede Hosentasche, und Zugriffe auf die Daten sind mit modernen PCs problemlos möglich. Die notwendigen Treiber sind in den aktuellen Betriebssystemen bereits integriert. Auch Messsysteme lassen sich im Format von USB-Sticks realisieren, solange sie nicht allzu komplex sind. Ein Beispiel ist ein USB-Stick eines chinesischen Herstellers, der Temperaturen und relative Luftfeuchten misst. Diese Größen gehören zu den maßgeblichen Faktoren für das Wohlbefinden des Menschen. Dagegen ist der Luftdruck eine Größe, die von Messsystemen dieses Formats nur selten erfasst wird. Unser Barostick, der diese Lücke schließt, arbeitet mit dem Luftdrucksensor BMP085 [1] von Bosch-Sensortec. Aus **Bild 1** geht hervor, wie der Sen-

sor in ein Mikrocontroller-System eingebunden wird. Im Sensor sind individuelle Kalibrierwerte gespeichert, die der Hersteller seinem Produkt mitgibt. Die Messergebnisse sind erst aussagefähig, nachdem sie über einen vorgegebenen Algorithmus mit den Kalibrierwerten verknüpft sind. Eine Bibliothek [2] fasst alle komplizierten Rechenoperationen zusammen, so dass dem schnellen Programmieren eigener Anwendungen nichts im Weg steht. Die Routinen sind auch im PC-Programm implementiert, das für den Barostick geschrieben wurde. Das PC-Programm stellt die Messwerte grafisch auf virtuellen analogen Anzeigeeinstrumenten dar. Natürlich können die Messwerte auch gespeichert werden.

Hardware

Für den Datenaustausch mit einem Mikrocontroller ist auf dem Chip des Sensors BMP085 eine I²C-

Schnittstelle integriert. Wichtig ist hier, dass die Spannung auf den I²C-Leitungen niemals +3,3 V übersteigen darf. Da ein USB-Port +5 V bereitstellt, ist eine Anpassung der Signalpegel zwingend notwendig. Anders als der Sensor BMP085 arbeitet der Mikrocontroller PIC18F14K50, eine SMD-Ausführung im SOIC-Gehäuse, unmittelbar an +5 V.

Wie die Schaltung in **Bild 2** zeigt, sind außer dem 12-MHz-Quarz, den zugehörigen Kondensatoren sowie einigen Kondensatoren zum Entkoppeln der Betriebsspannungen nur wenige weitere Bauelemente nötig. Ein Low-drop-Spannungsregler AP1117 setzt die Spannung +5 V auf die Betriebsspannung +3,3 V des Sensors BMP085 herab. Die bidirektionale (!) Signalpegelanpassung zwischen +5 V des Mikrocontrollers und +3,3 V des Sensors ist in **Bild 3** dargestellt. Folgende Signalflüsse können auftreten:

- Auf der Sensor-Seite (+3,3 V) wird eine Leitung nicht nach logisch 0 gezogen. Die Spannungen an Gate und Source des FET betragen +3,3 V, so dass die Source-Gate-Spannung die Schaltschwelle nicht erreicht, der FET sperrt. Auf der Mikrocontroller-Seite (+5 V) zieht ein Pullup-Widerstand die Leitung nach logisch 1.
- Auf der Sensor-Seite (+3,3 V) geht eine Leitung auf logisch 0. Die Source des FET ist ebenfalls logisch 0, während die Gate-Spannung auf +3,3 V bleibt. Nun übersteigt die Source-Gate-Spannung die Schaltschwelle, der FET leitet. Auch auf der Mikrocontroller-Seite (+5 V) wird die Leitung logisch 0.
- Auf der Mikrocontroller-Seite (+5 V) liegt eine Leitung auf logisch 0. Zuerst geht die Sensor-Seite (3,3 V) über die interne Drain-

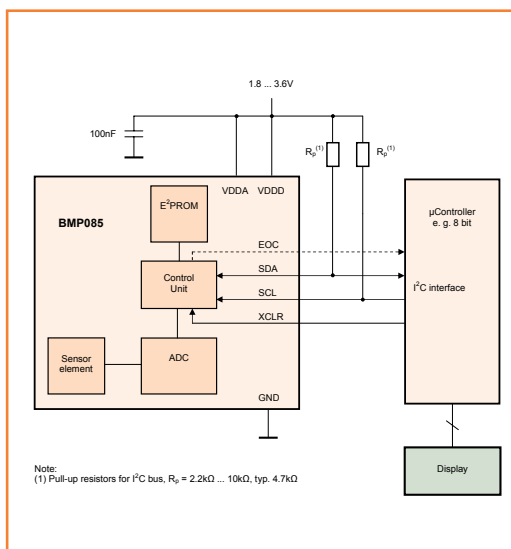


Bild 1.
Im BP085 sind der Sensor, ein Analog-Digital-Wandler, ein EEPROM mit individuellen Kalibrierdaten und eine I²C-Schnittstelle integriert.

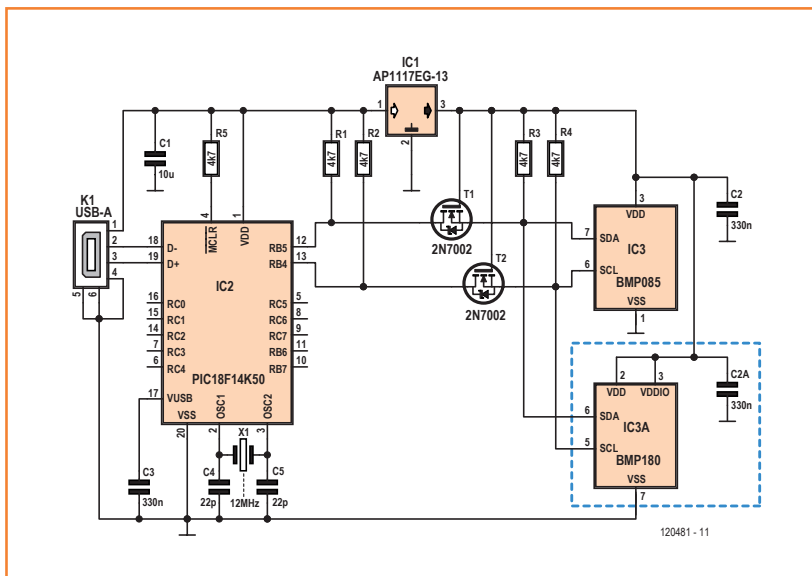


Bild 2.
Der USB-Luftdruck- und Temperaturmesser arbeitet mit einem Sensor-Chip von Bosch-Sensortec und einem Mikrocontroller aus der PIC-Familie.

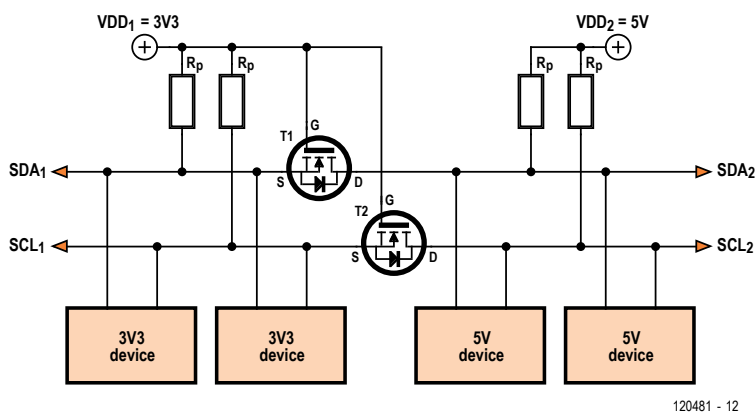


Bild 3.
Die Signalpegelanpassung zwischen 3,3 V und 5 V wird mit zwei FETs und einigen Pullup-Widerständen realisiert.

Stückliste

(Alle R und C: SMD-Bauform 0603)

Widerstände:

R1...R5 = 4k7

Kondensatoren:

C1 = 10 μ /10 V X5R

C2,C3 = 300 n

C4,C5 = 22 p

Halbleiter:

T1,T2 = 2N7002 (SOT232)

IC1 = AP1117E33G-13 (SOT-223)

IC2 = PIC18F14K50-I/SO (programmiert, 120481-41)

IC3 = Luftdrucksensor BMP085 oder BMP180 (Bosch Sensortec)

Außerdem:

X1 = Quarz 12 MHz (HC49US)

USB-A-Buchse, SMD-Ausführung (Lumberg 2410 07)

USB-Stick-Gehäuse (USB1SW, Conrad 531275)

Platine 120481-1

Aufgebautes und getestetes Modul: 120481-91 (siehe [3])

Software bestehend aus Quell- und Hexcode, Windows-Programm und DLL mit Quellcode:

Freier Download von [3]

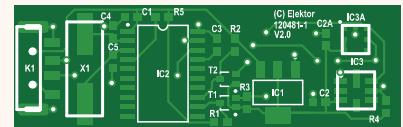


Bild 4.

Das Layout der Platine ist so gestaltet, dass für die Bestückung zwei Sensor-Typen von Bosch-Sensortec zur Wahl stehen.

Diode auf logisch 0. Anschließend leitet der FET, so dass die logischen Signale beider Seiten identisch sind.

In der Schaltung, die Bild 1 zeigt, sind zwei Sensor-Typen angegeben, der BMP085 (IC3) und der BMP180 (IC3A). Der BMP085 wurde vom Hersteller zum Auslauftyp erklärt, bei mehreren Anbietern ist er jedoch noch in größeren Stückzahlen am Lager. Nachfolger des BMP085 ist der BMP180, er ist zwar funktional voll kompatibel, hat jedoch eine andere Bauform mit abweichenden

den Anschlüssen. Auf der Platine kann wahlweise ein Exemplar des einen oder anderen Typs montiert werden. Die im Elektor-Shop erhältliche, aufgebaute und getestete Version ist zurzeit mit dem BMP085 bestückt.

Aufbau

Für den Barostick wurde eine Platine entworfen, die exakt in ein Standard-USB-Stick-Gehäuse passt (siehe **Bild 4**). Wie schon beschrieben, kann die Platine wahlweise entweder mit dem Sensor BP085 oder mit dem Sensor BP180 bestückt werden.

Sämtliche Bauelemente einschließlich USB-Buchse werden auf der Kupferseite montiert. Die Bauelemente haben zwar SMD-Bauform, etwas SMD-Erfahrung vorausgesetzt gelingt die Montage jedoch auch von Hand. Etwas knifflig, aber noch machbar, ist das Löten des Sensors BMP085. Zuerst werden die Beine des Chips und die zugehörigen Platinen-Kontaktflächen verzinnt. Dann wird der Chip in Position gebracht, und seine Anschlüsse werden durch vorsichtiges Berühren mit dem LötKolben an die Kontaktflächen gelötet. Wirklich sichere Verbindungen entstehen, wenn beim Löten an den Kontakten noch geringe Mengen Lötzinn hinzugegeben werden.

Für die Montage des alternativen Sensors BMP180 wird der Einsatz eines Reflow-Lötofen empfohlen, denn bei diesem Chip befinden sich die Anschlüsse auf der Unterseite. Alle übrigen Bauelemente können entweder gleichzeitig im Lötofen oder mit einem konventionellen LötKolben montiert werden. Auch wenn die Montage nach der beschriebenen Methode bisher stets erfolgreich war, kann es nicht schaden, zunächst alle Verbindungen mit

Bild 5.

Im Elektor-Shop ist die aufgebaute und getestete Platine zusammen mit dem passenden Gehäuse erhältlich (siehe [3]).



Schneller vom Entwurf zur fertigen Schaltung



Mit der Schaltungssimulationssoftware NI Multisim können Sie dank präziser Simulation und flexiblem Layout schnell einen Prototyp Ihrer Leiterplatte erstellen. Multisim 12.0 kombiniert eine intuitive Entwicklungsumgebung mit leistungsstarker Analysefunktionalität. Damit können Sie das Schaltungsverhalten optimieren und die Anwendungsleistung früher visualisieren – noch vor dem ersten Prototyp. Zudem verbessert Multisim Ihren Schaltungs-entwurf mit einer Datenbank aus Bauteilen von führenden Halbleiterherstellern sowie einer erweiterbaren Palette an Analysefunktionen, die in NI LabVIEW entwickelt wurden.

- Intuitive Bedienung auch ohne SPICE-Kenntnisse
- Schnell platzieren, routen und entwickeln
- Cosimulation von Schaltungen und Software-Algorithmen mit NI LabVIEW
- Mit weniger Iterationen zum Prototyp

>> ni.com/multisim/d



SENSOR+TEST
Halle 11, Stand 410

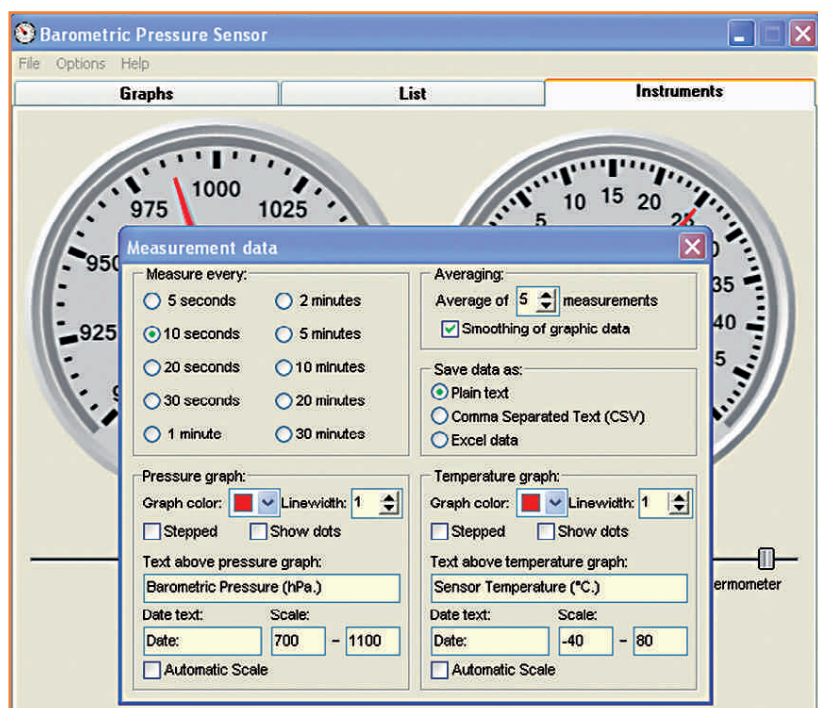


einem Ohmmeter auf einwandfreien Kontakt zu überprüfen.

Der Mikrocontroller muss vor der Montage programmiert werden, sofern das Exemplar nicht bereits programmiert ist. Der Quellcode in Pic-Basic und die zugehörige Hex-Datei sind kostenfrei auf der Webseite dieses Projekts [3] verfügbar. Zum Programmieren ist nicht unbedingt ein Adapter für das SOIC-Gehäuse nötig, es genügt ein DIL-SMD-Adapter mit vergoldeten Kontakten, wie ihn das Foto auf [4] zeigt. Solange die Programmierung läuft, wird der Mikrocontroller starr auf die Adapterplatine gepresst und festgehalten. Auch diese Methode hat in der Praxis ihre Bewährungsprobe bestanden. Trotzdem sollte verifiziert werden, dass der Inhalt des Programmspeichers im Mikrocontroller mit den Daten der Hex-Datei übereinstimmt.

Wenn kein für den PIC18F14K50 passendes Programmiergerät zur Hand ist, liefert der Elektor-Shop ein einsatzfertig programmiertes Exemplar des Controllers. Ein passendes Gehäuse, in das die Platine eingebaut werden kann, ist in der Stückliste angegeben. Im Elektor-Shop ist für den Barostick auch die bereits aufgebaute und getestete Platine zusammen mit dem Gehäuse erhältlich. Über die Bezugsmöglichkeiten gibt die Projektseite [3] Auskunft.

Bild 6.
Das unter Windows laufende PC-Programm stellt die Messdaten grafisch dar, dort lassen sich diverse Parameter einstellen.



Software

Zu diesem Projekt gehört ein Windows-Programm, das die Messergebnisse für Luftdruck und Temperatur auf dem PC-Bildschirm grafisch präsentiert. Die Messdaten können nach Tagen geordnet gespeichert werden, ferner sind die Messintervalle festlegbar. In **Bild 5** ist dargestellt, wie das Programm auf dem Bildschirm aussieht. Dem Anwender bietet das Programm diverse Optionen, um die Sprache, die Farben und Linienstärken sowie weitere Parameter an eigene Präferenzen anzupassen. Die Grafiken der virtuellen runden Messinstrumente können zu vorgegebenen Zeiten über FTP auf eine Webseite geladen werden, beispielsweise um die Repräsentation der Daten in Echtzeit in einen Web-Auftritt einzubinden. Freunde der Wetterbeobachtung haben Gelegenheit, die Messdaten auf einen Server des Wetterkanals „Weather Underground“ [5] hochzuladen.

Wie schon erwähnt, steht auch das Windows-Programm auf der Projektseite [3] zum kostenfreien Download bereit. Wer gern programmiert, dem leistet die Bibliothek PSensor.dll Hilfestellung. Den Einsatz dieser DLL erläutert das in Delphi geschriebene Testprogramm DllDemo.exe. Interessierte Leser können auch auf den Quellcode der Bibliothek PSensor.dll zurückgreifen.

Bevor der Barostick zum ersten Mal in den PC eingesteckt wird, ist das Überprüfen der Stromaufnahme ratsam. Wenn der Strom in der Größenordnung 10 mA liegt, kann der Barostick an den PC angeschlossen werden. Kurz nach dem Einstecken erkennt Windows die neue Hardware. Da sich der Barostick wie ein „HID Device“ verhält, ist das Laden eines spezifischen Treibers nicht notwendig. Windows meldet nun, dass ein „Pressure Sensor“ gefunden wurde. Die nächste Meldung besagt, dass die neue Hardware konfiguriert worden ist, sie steht nun für den Einsatz bereit. Das PC-Programm wird gestartet, dort können anschließend die Einstellungen vorgenommen werden.

Wir wünschen Ihnen langfristig schönes Wetter, was Perioden mit hohen Luftdrücken gleichkommt!

(120481)gd

Weblinks

- [1] www.datasheetarchive.com/barometric-BMP085-datasheet.html
- [2] www.property-protection.nl/Temper
- [3] www.elektor.de/120481
- [4] www.voti.nl/common/pcb-12.jpg
- [5] www.wunderground.com

3. Integrate the following wrt x :

(i) $x^2 \cos 3x$ (ii) $x^3 e^{3x}$

$x^2 x^3 \cos e^{3x} 3x$

not "integrate" literally!

If only RF could be so easy.

Linx
TECHNOLOGIES

Wireless made simple®

RF Modules
Remote Controls
Antennas
RF Connectors
Custom Designs

www.linxtechnologies.com

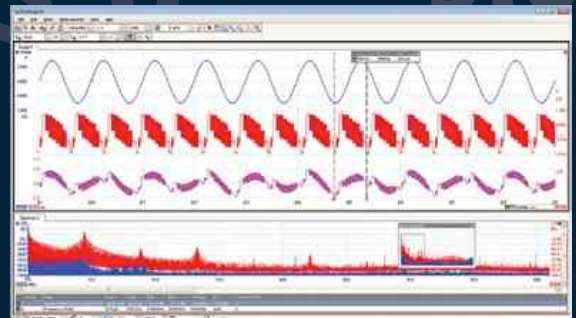
pico®
Technology

DAS WELTWEIT ERSTE USB 3.0 OSZILLOSKOP



PicoScope	3207A	3207B
Bandbreite	250 MHz	250 MHz
Sampling	1 GS/s	1 GS/s
Speicher	256 MS	512 MS
Signalgenerator	Funktionsgenerator	AWG
Preis	€1813	€1451
Stromversorgung	Aus USB anschluss	
Kompatibilität	USB 2.0 & 3.0	

ALLE MODELLE WERDEN MIT TASTKÖPFEN, KOMPLETTER SOFTWARE UND 5 JAHRE GARANTIE GELIEFERT. DIE SOFTWARE ENTHÄLT: MESSUNGEN, SPEKTRUM-ANALYZER, KOMPLETTES SDK, ERWEITERTER TRIGGER, FARB-PERSISTENZ, SERIELLE DEKODIERUNG (CAN, LIN, RS232, I²C, I²S, SPI), MASKEN, MATH-EM, KANÄLE - ALLES ALS STANDARD. KOSTENLOSE UPDATES.



www.usb3scope.com/PS208

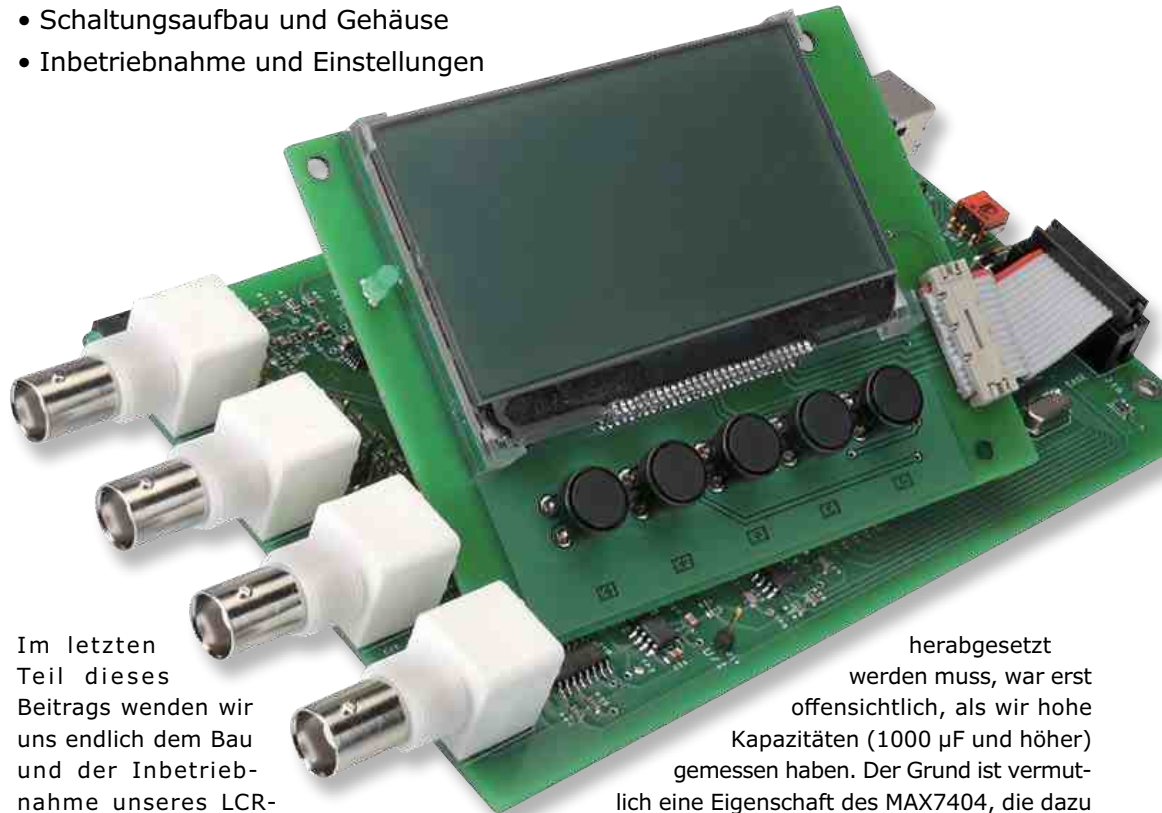
LCR-Meter 2013 (3)

Höchste Präzision zu niedrigen Kosten

In die dritte und letzte Runde geht unser LCR-Meter mit folgenden Themen:

- Kleine Korrekturen am Schaltbild
- Platinenlayout und Stückliste
- Schaltungsaufbau und Gehäuse
- Inbetriebnahme und Einstellungen

Von
Jean-Jacques Aubry
(F)



Im letzten Teil dieses Beitrags wenden wir uns endlich dem Bau und der Inbetriebnahme unseres LCR-Meters zu. Doch bevor der Lötkolben angeheizt wird, kehren wir noch einmal zum Schaltbild im vorangegangenen Teil zurück.

Ungereimtheiten

Bei den Vorbereitungen dieses letzten Teils unseres dreiteiligen Beitrags kam es ans Licht: Zwischen dem im zweiten Teil veröffentlichten Schaltbild und unserer Stückliste hatten sich kleine Ungereimtheiten eingeschlichen. Das Schaltbild des LCR-Meters hatten wir in Elektor, Ausgabe März 2013, veröffentlicht. Darin fehlt die Angabe der Frequenz von Quarz Y1, die Frequenz beträgt 24 MHz. Die parallelen Kopplungskondensatoren C3 und C9 vor dem Puffer U1 müssen die Kapazität 1 μF haben, der Wert 4,7 μF hat sich als zu hoch erwiesen. Dass dieser Wert

herabgesetzt werden muss, war erst offensichtlich, als wir hohe Kapazitäten (1000 μF und höher) gemessen haben. Der Grund ist vermutlich eine Eigenschaft des MAX7404, die dazu führt, dass das Eingangssignal des Analog-Digital-Wandlers mit niedriger Frequenz schwebt. Da hohe Werte von C3 und C9 diese unerwünschte Erscheinung begünstigen, haben wir die Werte herabgesetzt.

Für U2 ist der MAX7404CSA der geeignete Typ, der MAX7404ESA hat den erweiterten Temperaturbereich $-40...+85\text{ }^{\circ}\text{C}$, der jedoch für das LCR-Meter nicht notwendig ist. U10 (Zenerdiode-Symbol rechts neben U9) kann auch ein LM4040 sein. Wir hatten den LM4050 genommen, weil wir ihn griffbereit hatten. Operationsverstärker U6 muss ein OPA365 sein, denn seine Offsetspannung ist wesentlich niedriger als die Offsetspannung des OPA354. Die nächste Korrektur betrifft U19: Dies ist FT232RL, ein Tippfehler ist verantwortlich, dass daraus ein FT235RL wurde.

Wichtige Eigenschaften

Das LCR-Meter arbeitet nach dem Prinzip der automatischen Impedanzmessbrücke. Bei der Entwicklung stand höchste Messgenauigkeit bei unkompliziertem Aufbau an erster Stelle. Gemessen werden Widerstände, Kapazitäten und Induktivitäten passiver Komponenten im Bereich 1 mΩ...1000 MΩ. Die Frequenz des Messsignals kann 100 Hz (oder 120 Hz, abhängig von der Netzfrequenz 50 Hz oder 60 Hz), 1 kHz oder 10 kHz betragen.

Zwei Gerätekonfigurationen stehen zur Wahl:

- Messmodul ohne Display und ohne Bedienelemente, Betrieb nur zusammen mit PC, Verbindung über USB, Bedienung über PC-Programm. Das PC-Programm basiert auf der Qt-Bibliothek von Atmel, es wurde unter Windows XP, Linux Ubuntu 11.04 sowie MacOS X Snow Leopard, Lion und Mountain Lion getestet. Unter neueren Windows-Versionen ist das Programm ebenfalls lauffähig.
- Messmodul ergänzt durch eine Erweiterungsplatine, darauf ein LC-Display und die Bedientaster. Die Funktionen sind vollständig identisch mit der PC-Version. Stromversorgung über die USB-Buchse mit externem Netzteil oder aus externem USB-Port.

Bei den Einstellarbeiten fiel uns auf, dass R17 ein Widerstand mit dem Wert 10 Ω sein muss, die im Schaltbild stehende Angabe „NC“ (*Not Connected*) ist nicht korrekt. Die übrigen Bauelemente mit der Bezeichnung „NC“ müssen nur während der Kalibrierung an ihren Plätzen sein. Hinweise zur Kalibrierprozedur enthält ein nachfolgender Absatz. Wir empfehlen, die genannten Korrekturen in das Schaltbild einzutragen, so dass sie beim Bestücken der Platine nicht vergessen werden. Das korrigierte Schaltbild werden wir auf die Projektseite [3] einstellen, von dort kann es zusammen mit der Software, den Platinenlayouts und den Dokumentationen heruntergeladen werden.

Aufbau

Zum LCR-Meter gehören zwei Platinen, **Bild 1...3** zeigt die Layouts. Die zweite Platine, Erweiterungsplatine genannt, wird **nur für die Stand-alone-Version** benötigt! Bei der PC-Version ist der PC für die Anzeige- und Bedienfunktionen zuständig.

Wenn Sie über die notwendige Arbeitsplatz-Ausstattung verfügen und genügend Erfahrung in der Kunst der SMD-Montage besitzen, können Sie die leeren Platinen in Eigenregie bestücken. Trotzdem möchten wir Sie warnen: Dieses Werk gelingt nur Spezialisten, bitte bedenken Sie vor Baubeginn, worauf Sie sich einlassen.

Wir empfehlen Ihnen, der Stückliste konsequent zu folgen. Damit meinen wir nicht die Reihenfolge der Bauelemente in der Stückliste, sondern die Werte und Toleranzen. Abgesehen von vier Präzisionswiderständen haben alle Widerstände die Toleranz 1 %. Widerstand R50 hat den Wert 0 Ω,

er verbindet die analoge und die digitale Masse an einem einzigen, definierten Schaltungspunkt. Eine Lötbrücke ist für R50 ausreichend. Die Kondensatoren müssen unbedingt den Angaben in der Stückliste entsprechen, kritisch sind insbesondere die NPO-Typen.

Der nächste Hinweis betrifft den Quarz. Auf der Platine befindet sich ein Lötspunkt mit der Bezeichnung „XTAL CASE“. Dieser Lötspunkt muss über ein kurzes Stück Draht mit dem Metallgehäuse des Quarz verbunden werden, nur dann ist die Abschirmung des Quarzes wirksam. Doch Vorsicht beim Löten am Gehäuse, der Quarz ist empfindlich gegen Überhitzung, die Lötzeit darf nur kurz sein! Ein weiterer, nicht weniger wichtiger Hinweis zur Bestückung im Do-it-yourself-Verfahren: Der Mikrocontroller wird vom Hersteller absolut „nackt“ ausgeliefert. Vor der Montage muss der Bootloader in den Mikrocontroller geflasht werden. Das geschieht mit einem separaten Modul, das die Bezeichnung „USB Debug Adapter“ hat und bei Silicon Laboratories erhältlich ist.

Die folgenden Anmerkungen beziehen sich auf das Display- und Bedienmodul der Stand-alone-Version: Mit „HE10“ ist ein Flachkabel-Steckverbinder gemeint, der unmittelbar auf die Platine gelötet wird (**Bild 4a** und **4b**). Dieser Steckverbinder muss flach sein, damit die Platine in das vorgesehene Gehäuse passt. Das Display wird an einer Kante über eine Reihe kleiner Lötstützpunkte festgelötet, parallel dazu liegen zwei Lötspunktreihen normaler Größe. Diese Verbindungen erschienen uns nicht genügend stabil, falls das LCR-Meter rauen mechanischen Beanspruchungen ausgesetzt ist. Deshalb haben wir das Dis-

play oberhalb der Lötunkte zusätzlich mit einem Zweikomponentenkleber fixiert.

Für den Fall, dass Unsicherheiten bei der Wahl der korrekten Bauelemente auftauchen, haben wir auf der Projektseite [3] eine zweite Version der Stückliste (BOM, *Bill Of Materials*) eingestellt. Jene Version enthält weitere, möglicherweise wichtige Details.

Die Reihenfolge, in der die Bauelemente montiert werden, hängt von den Abmessungen ab. Den Anfang machen die kleinsten Exemplare, womit nicht nur die Widerstände und Kondensatoren gemeint sind. Auch die ICs mit den schmalsten Abständen zwischen den Anschlüssen (*Lead Spacing*) gehören dazu. Die auf der Platinenrückseite zu montierenden Komponenten dürfen natürlich nicht fehlen. Die mit „NC“ markierten Kompo-

nenten sind nur für die Kalibrierung bestimmt, sie werden erst später montiert.

Die vorstehenden Hinweise gelten für Lötprofis und andere Do-it-yourself-Enthusiasten. Wenn Sie nicht zu dieser Gruppe gehören oder Ihre Zeit zu knapp bemessen ist, können Sie die aufgebauten und funktionsgetesteten Module im Elektor-Shop erwerben. Die Module kommen dann ins Haus, sie müssen nur noch im Gehäuse befestigt werden. Sofort nach der Kalibrierung ist das Gerät betriebsbereit. Das LCR-Meter ist ein empfindliches elektronisches Messgerät, das vor Störeinflüssen, insbesondere aus dem Stromnetz abgeschirmt werden muss. Ein Metallgehäuse, zum Beispiel der Typ 1455L1601 von Hammond, ist eine gute Wahl, um die Module gemeinsam unterzubringen. Ein Bohrplan für dieses Gehäuse gehört zur Dokumentation auf der Projektseite [3].

Stückliste Hauptplatine

Widerstände:

(SMD 0805, 1 %, wenn nicht anders angegeben)

R1,R16,R17,R28 = 10 Ω
 R2,R34 = 820 k
 R3,R5,R11,R13 = 8k2
 R4,R10,R47,R55,R56,R71,R72,R78,R81,R82,R94,R95 = 10 k
 R6,R58,R59 = 1k8
 R7,R100 = 5 k Trimpoti (Vishay-Sfernice TS53YJ502MR10)
 R8,R60,R62 = 16 k
 R9,R23,R25,R35,R38,R39,R41,R52,R68,R69,R70,R87,R88,R96 = 56 Ω
 R12,R14 = 5k6
 R15,R97 = NC (Not Connected)
 R18 = 10 k 0,05 %, 10 ppm (Panasonic ERA6ARW103V)
 R19 = 1 Ω
 R20 = 1 k 0,05 %, 10 ppm (Panasonic ERA6ARW102V)
 R21 = 100 Ω 0,05 %, 5 ppm (Vishay-Dale TNPU0805100RAZEN00)
 R22 = 100 k 0,05 %, 10 ppm (Panasonic ERA6ARW104V)
 R24,R26,R27,R29,R33,R36,R37,R40,R44,R57,R64 = 100 Ω
 R30,R61,R76,R77,R80,R101 = 20 k
 R31 = 750 Ω
 R32,R42,R49,R66,R93 = 100 k
 R43,R84,R85,R86,R89 = 2Ω2
 R45,R73 = 39 k
 R46,R90,R91 = 680 Ω
 R48,R51,R74 = 4k7
 R50 = 0 Ω
 R53,R65 = 470 Ω
 R54 = 1 k
 R63,R98 = 1k6
 R67 = 62 Ω
 R75 = 5 k Trimpoti (Bourns 3266W-1-502LF)
 R79 = 4k3
 R81 = 7k5
 R83 = 30 k
 R92 = 430 Ω
 R99 = 2 k

Kondensatoren:

(SMD 0805, wenn nicht anders angegeben)

C1,C2,C4,C10,C11,C12,C20,C26,C27,C28,C29,C31,C33,C34,C35,C36,

C37,C40,C41,C43,C44,C45,C46,C47,C48,C49,C50,C51,C53,C54,C61,
 C63,C65,C78,C80,C86,C87,C88 = 100 n/25 V, 10 %, X7R
 C3,C9,C62,C64,C90 = 1 μ/25 V, 10 %, X7R
 C5,C13,C18 = 15 n/50 V, 5 %, NP 0, SMD 1206
 C6,C7,C14,C15,C16,C56 = 47 n/50 V, 5 %, NP 0, SMD 1206
 C8,C73,C89 = 1 n/50 V, 5 %, NP 0
 C17,C21,C23,C55,C60 = NC (Not Connected)
 C19,C25 = 150 p/50 V, 5 %, NP 0
 C22,C52,C58,C59 = 1,5 n/50 V, 5 %, NP 0
 C24,C32,C69,C72,C74,C75,C81,C85 = 2μ2/16 V, 10 %, X7R
 C30,C57 = 4n7/50 V, 5 %, NP 0
 C38,C39,C83,C84 = 33 p, 5 %, NP 0
 C42,C70 = 33 μ/6,3 V, Tantal, Bauform A (Vishay-Sprague 293D336X96R3A2TE3)
 C66,C71,C82 = 10 n/50 V, 10 %, X7R
 C67,C77,C79 = 4μ7/10 V, 10 %, X5R
 C68,C76 = 470 μ/6,3 V, Tantal, Bauform D (Kemet T495D477K006ATE100)

Induktivitäten:

L1 = Doppelinduktivität 20 μH, SMD (Bourns PM3602-20-RC)
 L2,L3 = Ferritperle, SMD 0805 (Murata BLM21PG221SN1D)

Halbleiter:

(SMD, wenn nicht anders angegeben)

D1,D2,D3,D4,D5 = BAV199 (SOT23)
 D6 = LED 3 mm grün, bedrahtet, Montage horizontal (Dialight 551-0207F)
 D7 = MBR0520 (SOD123)
 D8,D9 = LED rot, SMD 0805 (z. B. Kingbright KP-2012SURC)
 D10 = BAT54A (SOT23)
 U1 = OPA725AIDBV (TI, SOT23-5)
 U2 = MAX7404CSA (Maxim, SOIC-8)
 U3 = 74HCT4052D (SOIC-16)
 U4,U20 = TLC2274AID (TI, SOIC-14)
 U5 = INA128U (TI, SOIC-8)
 U6,U11,U14 = OPA365DBV (TI, SOT23-5)
 U7 = PGA103U (TI, SOIC-8)
 U8 = 74HCT4053D (SOIC-16)
 U9 = C8051F061-GQ (Silicon Laboratories, TQFP-64)
 U10 = LM4040D25IDBZ (TI, SOT23)
 U12 = DAC8811CDGK (TI, MSOP-8)

Vorbereitungen

Bohren Sie an den Seiten runde Durchbrüche für die BNC-Buchsen und für die LED D6. Die USB-Buchse und der Drehschalter SW1 haben ihren Platz auf der anderen Seite, hier müssen die Durchbrüche rechteckig sein. Wenn keine Stanzwerkzeuge verfügbar sind, genügen auch ein Satz feiner Feilen, ein Metalllineal, ein gutes Augenmaß sowie natürlich etwas Ausdauer und Geduld. Das Gehäuse muss mit dem Massepunkt der Hauptplatine verbunden werden. Dazu ist eine Bohrung im Gehäuseboden nötig, die der Bohrung der Hauptplatine nahe J16 gegenüber liegt. Eine M3-Gewindeschraube zusammen mit einer 5,5 mm langen Abstandsrolle, mehreren Distanzscheiben und einer Mutter stellen die elektrische und mechanische Verbindung her. Das Display- und Bedienmodul (DKM) wird im Gehäusedeckel montiert. Auf

der Oberseite wird der Deckel mit den Funktionen der Bedienelemente beschriftet. Die Schrift kann graviert werden, oder sie wird im Siebdruckverfahren aufgetragen (siehe **Bild 5**). Das Display- und Bedienmodul ist stabil befestigt, wenn es von vier 12 mm langen M3-Gewindeschrauben mit 7 mm langen Abstandsrollen und einigen zusätzlichen Distanzscheiben gehalten wird.

Zum Schluss dieses Absatzes noch zwei Hinweise zu den Jumpers: Im Regelfall startet das PC-Programm über die USB-Verbindung die Firmware des Mikrocontrollers. Das ist jedoch nur möglich, wenn das LCR-Meter fehlerfrei arbeitet. Ist diese Voraussetzung nicht erfüllt, beispielsweise weil ein Firmware-Update einen Bug enthält, muss der Bootloader einen unbedingten Reboot durchführen. Zum unbedingten Reboot wird der Bootloader mit Jumper J17 veranlasst.

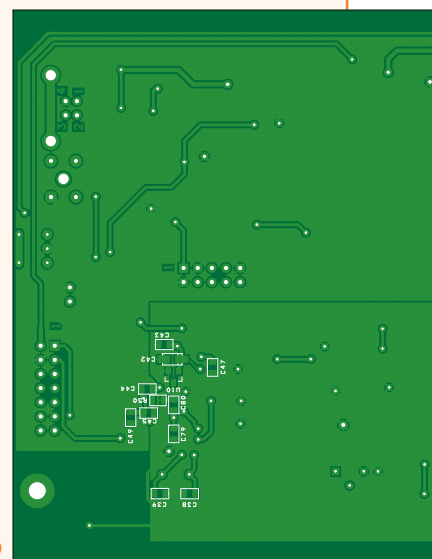
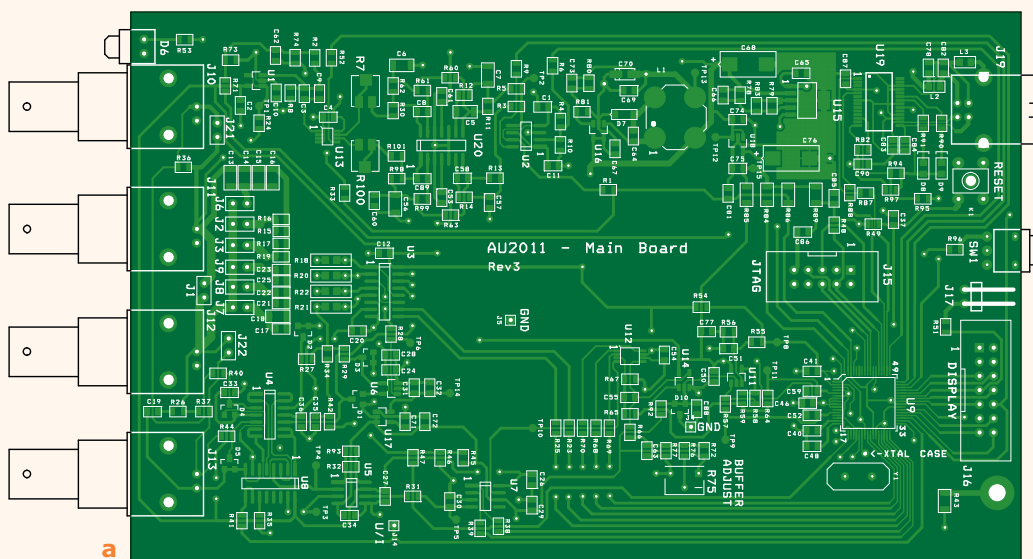


Bild 1. Diese Seite (a) der doppelseitigen Messmodul-Platine wird zwar Bestückungsseite genannt...
...doch auch auf der anderen Seite (ohne Aufdruck!) müssen Komponenten montiert werden (b).

U13 = SN74LVC2G53DCT (TI, SM-8)
U15 = REG102GA-A (TI, SOT223-5)
U16 = LT1611CS5 (Linear Technology, SOT23-5)
U17 = TPS72325DBV (TI, SOT23-5)
U18 = TLV70030DDC (TI, SOT23-5)
U19 = FT232RL (FTDI, SSOP-28)

Außerdem:

Y1 = Quarz 24 MHz (z. B. Euroquartz 24.000 MHz
HC49/4H/30/50/40+85/18pF/ATF)
SW1 = Drehschalter (RS Components 734-6934)
J1,J2,J3,J6,J7,J8,J17,J21,J22 = Stiftkontaktleiste 2-polig, Raster
2,54 mm

J4,J5,J14 = Stiftkontakt 1-polig abgewinkelt (siehe Text)
J10,J11,J12,J13 = BNC-Buchse horizontal, isoliert (z. B. TE Connectivity
1-1337543-0)
J15 = Stiftkontaktleiste 10-polig HE10 (z. B. Multicomp MC9A12-1034)
J16 = Stiftkontaktleiste 14-polig HE10 (z. B. Multicomp MC9A12-1434)
J17 = Stiftkontaktleiste 2-polig abgewinkelt, Raster 2,54 mm
J19 = USB-Buchse Typ B, horizontal (z. B. TE Connectivity 292304-1)
K1 = Drucktaster (Omron B3F-10xx Series)
Platine 110758-1

oder aufgebaute und getestete Platine 110758-91
Gehäuse: Hammond, 1455L1601 mit Beschriftung
Schrauben und Muttern (Gewinde M3), Abstandsrollen, Scheiben

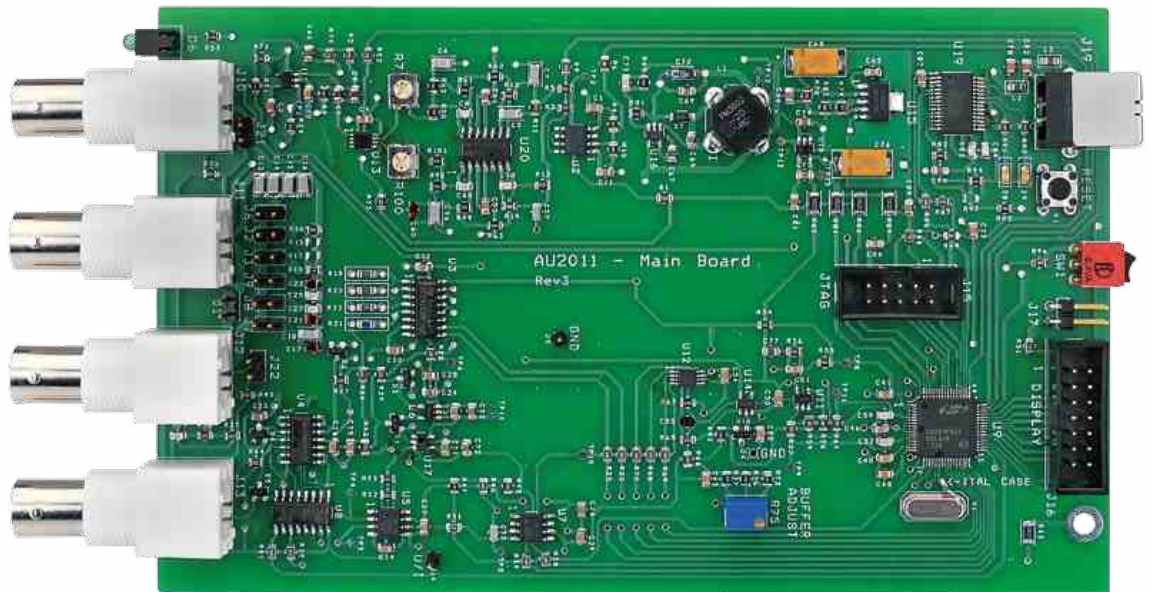


Bild 2.
Dies ist der Prototyp des
LCR-Meters.

Die Dokumentation, die Bestandteil des Downloads auf der Projektseite [3] ist, enthält eine Beschreibung der Startprozedur.

Wir empfehlen, die Kontaktstifte der Testpunkte J4, J5 und J14 rechtwinklig abzubiegen, so dass sie für die Messklemmen leicht zugänglich sind. Hinweise zu den übrigen Jumpers enthält die Online-Dokumentation auf der Projektseite.

Halbautomatische Kalibrierung

Hochpräzise Messgeräte wie unser LCR-Meter legen nahe, dass zum Kalibrieren und Einstellen außer einem hohen gerätetechnischen Aufwand auch Spezialkenntnisse gefordert sind. Beides trifft hier nicht zu. Außer einem Multimeter zur Kontrolle der Betriebsspannungen werden keine Messgeräte benötigt, es genügen ein kleiner Schraubendreher und etwas Geduld. Die Software

Stückliste Display- und Bedienmodul

Widerstände:

(SMD 0805, 1 %)
R1,R2,R3,R4,R10,R11 = 4k7
R5,R6,R7,R9 = 56 Ω
R8 = 1 k

Kondensatoren:

(SMD 0805)
C1,C2,C3,C4,C5,C6,C7,C8,C9,C10 = 1 μ /25 V, 10 %, X7R

Halbleiter:

D1,D2,D3,D4 = BAT54A (SOT23)
D5 = LED 3 mm grün, bedrahtet (z. B. Kingbright L-424GDT)
Q1,Q2 = MOSFET N-Kanal (SOT23, z. B. Fairchild FDF303N)

Außerdem:

U1 = Grafik-Display: Displaytech 64128M-FC-BW-3
J1 = Stiftkontaktleiste 14-polig, **Low-Profile** HE10, **Höhe \leq 5,4 mm** (Harting 09 18 114 9622)
K1,K2,K3,K4,K5 = Drucktaster (Multicomp TS0B22)
Flachkabel-Steckverbinder für 14-adriges Kabel (z. B. Multicomp MC6FD014-30P1)
14-adriges Flachkabel, Länge 1 m, 10 cm werden benötigt (3M 3365-14)

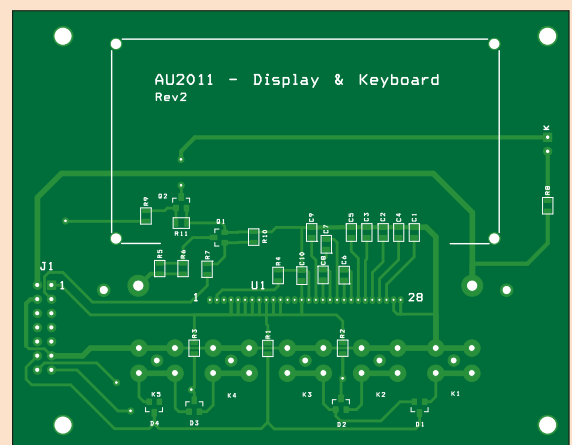


Bild 3. Platine für das optionale Display- und Bedienmodul.

Platine 110758-2
oder aufgebaute und getestete Platine 110758-92

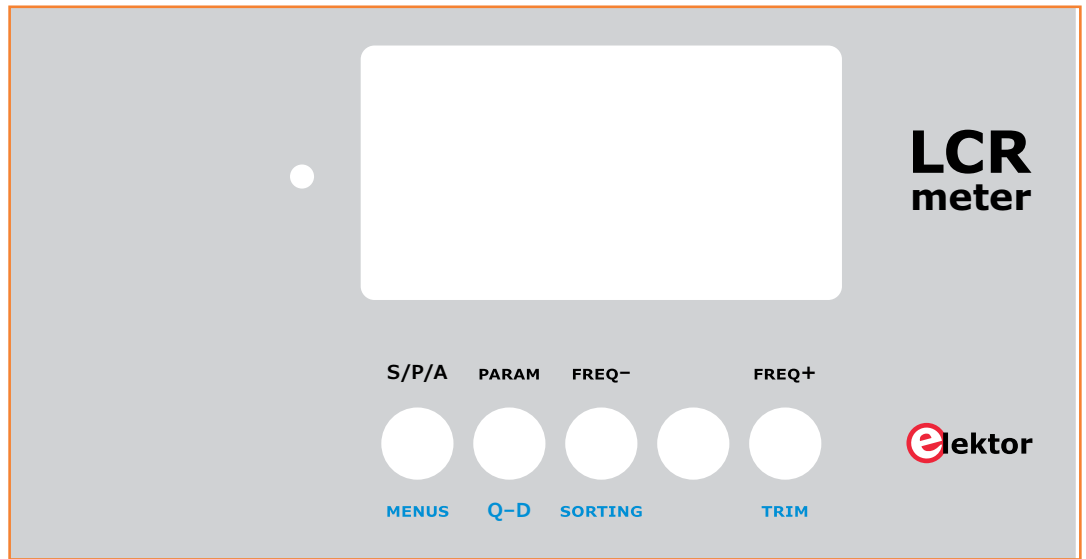


Bild 6.
Beschriftung der Frontplatte,
für Gravur oder Siebdruck.

LEDs D8 und D9 auf. Dies ist ein Zeichen dafür, dass der PC mit dem FT232R (U19) auf der Messmodul-Platine Daten austauscht.

Mit dem „USB Debug Adapter“ aus **Bild 8**, angeschlossen an J15, und dem Programm *Flash-Util* wird der Bootloader in den Mikrocontroller

geflasht. Die Online-Dokumentation erklärt diese Prozedur sowie die nachfolgenden Schritte.

Danach starten Sie auf dem PC das Programm *AU2011*. Das Programm stellt die Verbindung mit dem Bootloader her und öffnet einen seriellen Port. Ist auch dies gelungen, muss nur noch über



Bild 7a.
Die Kelvin-Messklemmen mit
BNC-Anschlüssen kosten im
Handel ungefähr 10 €.



Bild 7b.
Der Messkopf von Hameg passt unmittelbar auf die LCR-Meter-Anschlüsse.

Utilities/Load Firmware (Menü von *AU2011*) die Firmware in den Mikrocontroller geladen werden. Starten Sie nun das LCR-Meter neu und führen Sie die halbautomatische Einstellung der Offsets durch. Der Ablauf dieser Prozedur ist im Dokument *Getting Started* unter [3] beschrieben. Klicken Sie im PC-Programm *AU2011* auf *Start*, um die erste Messung durchzuführen. Wenn Sie jetzt das Symbol der offenen Messklemmen sehen (**Bild 9**), dann haben wir Anlass, Ihnen zu gratulieren: Sie haben das LCR-Meter 2013 erfolgreich aufgebaut!

(130093)gd

- Gehäusebohrpläne für PC- und Stand-alone-Version
- Frontplatte (Entwurf, nur für Stand-alone-Version)
- Manuals, Getting Started und User Manual



Bild 8.
In den leeren Mikrocontroller muss der Bootloader geflasht werden, mit Hilfe dieses Moduls.

Weblinks und Literatur

- [1] LCR-Meter 2013, Teil 1, www.elektor.de/110758
- [2] LCR-Meter 2013, Teil 2, www.elektor.de/130022
- [3] LCR-Meter 2013, Teil 3, www.elektor.de/130093

Download: Software, Platinenlayout und Dokumentation

- Software (Bootloader, Firmware und PC-Programm)
- Platinenlayouts 1 und 2
- Bestückungspläne Oberseiten Platinen 1 und 2
- Bestückungsplan Unterseite Platine 1
- Erweiterte Stückliste (BOM)
- XL-Version des Schaltbilds

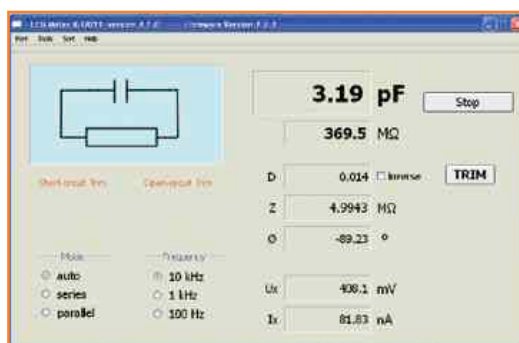
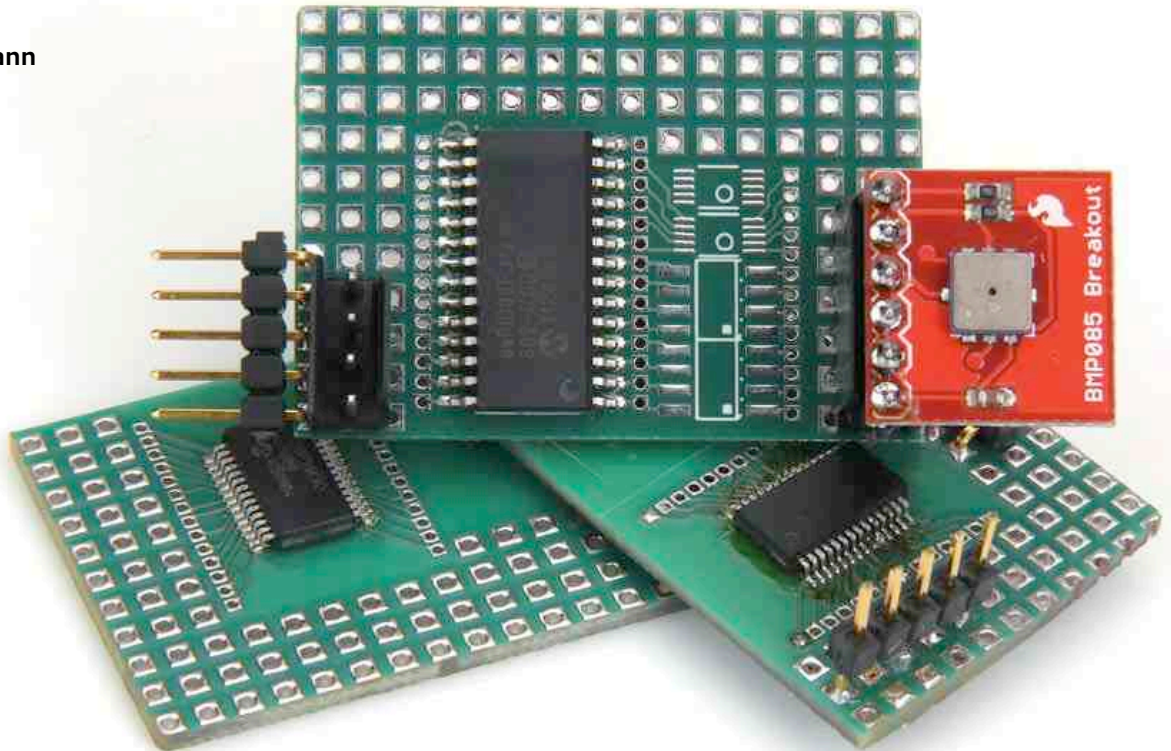


Bild 9.
Das Fenster mit dem Symbol des offenen Messkreises ist das Zeichen dafür, dass das LCR-Meter ordnungsgemäß arbeitet.

PICo PROto

Wie testet man schnell einen neuen Sensor? Und wie probiert man eine neue Idee aus, ohne jeweils ein komplexes und teures Entwicklungs-Board konfigurieren zu müssen? Diese Frage stelle ich mir oft, und getreu der Regel, dass man gerade auf die einfachsten Dinge am häufigsten zurückgreift, habe ich PICo PROto entwickelt, ein minimalistisches Prototyping-Tool...

Von
Michel Kuenemann
(F)



Technische Eigenschaften

- passend für SO28- und SSOP28-Gehäuse
- minimale Kosten und Platzbedarf
- maximale Flexibilität
- kürzeste Aufbauzeit

Neue Technologien sind faszinierend. Vor allem der Einsatz von neuartigen Bauteilen, hauptsächlich im MEMS-Bereich (*microelectro-mechanical systems*) interessiert mich. Diese werden im Allgemeinen mit 3,3 V betrieben und liefern ihre

Daten über Analogausgänge oder einen I²C-Bus, seltener via SPI.

Viele der modernen Sensoren werden in winzigen Gehäusen gefertigt, welche nicht per Hand gelötet werden können. Glücklicherweise haben die Hersteller dieser Wunderwerke ein Herz für Entwickler, die sie einsetzen sollen, und bieten hierfür oft kleine Entwicklungsplatinen mit aufgelötetem Sensor an. Diese so genannten *Breakout-Boards*

besitzen einen Verbinder im 2,54-mm-Raster und sind 1:1 mit den Anschlüssen des eigentlichen Bausteins verbunden.

Ich setze laufend verschiedene Mikrocontroller aus der PIC18-Familie von *Microchip* im 28-Pin-Gehäuse ein. Der Wechsel zwischen den verschiedenen Typen ist einfach, da *Microchip* die hervorragende Idee hatte, die Anschlüsse dieser Bausteine zu standardisieren. Außerdem werden die Bausteine im traditionellen DIP-Gehäuse angeboten, so dass sich auf Experimentierplatinen völlig ohne Löten Versuchsaufbauten herstellen lassen. Allerdings haben die Steckbretter drei große Nachteile: erstens lassen Größe und Gewicht den Einbau etwa in Flugmodelle nicht zu. Zweitens ist die Qualität der Verbindungen eher mau und drittens kann man die Schaltung selten lange so belassen, da das wertvolle Breadboard

Geeignet für alle PIC16/PIC18 in 28-poligen Gehäusen

dringend für den nächsten Einsatz benötigt wird. Sie haben Recht, wenn Sie hier einwenden, dass *Microchip* viele bezahlbare Evaluation-Boards anbietet, die nur auf meine Ideen gewartet haben, aber für den Großteil meiner Versuche sind diese Boards in punkto Anschlüsse, Platzbedarf und Preis deutlich überdimensioniert.

Ich habe daher eine minimalistische doppel-seitige Platine entworfen; für PIC16 und PIC18 im SO28-Gehäuse auf der einen Seite und im SSOP28-Gehäuse auf der anderen. Auch ist Platz für einige SMD-Bausteine. Außen herum ist ein Feld mit quadratischen Löt pads im 2,54-mm-Raster für bedrahtete Bauteile und den unverzichtbaren (Programmier-)Anschluss, der den Prototyp mit der Außenwelt verbindet. Durch die

geringe Größe von 28 x 42 mm kann der Prototyp sowohl in einem Flugzeugmodell oder einem kleinen Roboter als auch in größeren Aufbauten verwendet werden. Der Zwang zum Platzsparen hat auch sein Gutes, da der Entwickler sich auf den eigentlichen Zweck konzentrieren muss und nicht in Versuchung kommt, die Schaltung mit Überflüssigem zu überfrachten.

Das Schaltbild

Keine Überraschungen, es ist die Einfachheit selbst. Sogar die Bezeichnung Schaltbild (**Bild 1**) ist etwas hoch gegriffen, da keinerlei Bauteile enthalten sind. Somit kann man direkt zur Praxis übergehen. Nebst den SO28- beziehungsweise SSOP28-Pads für den Controller (bitte immer nur

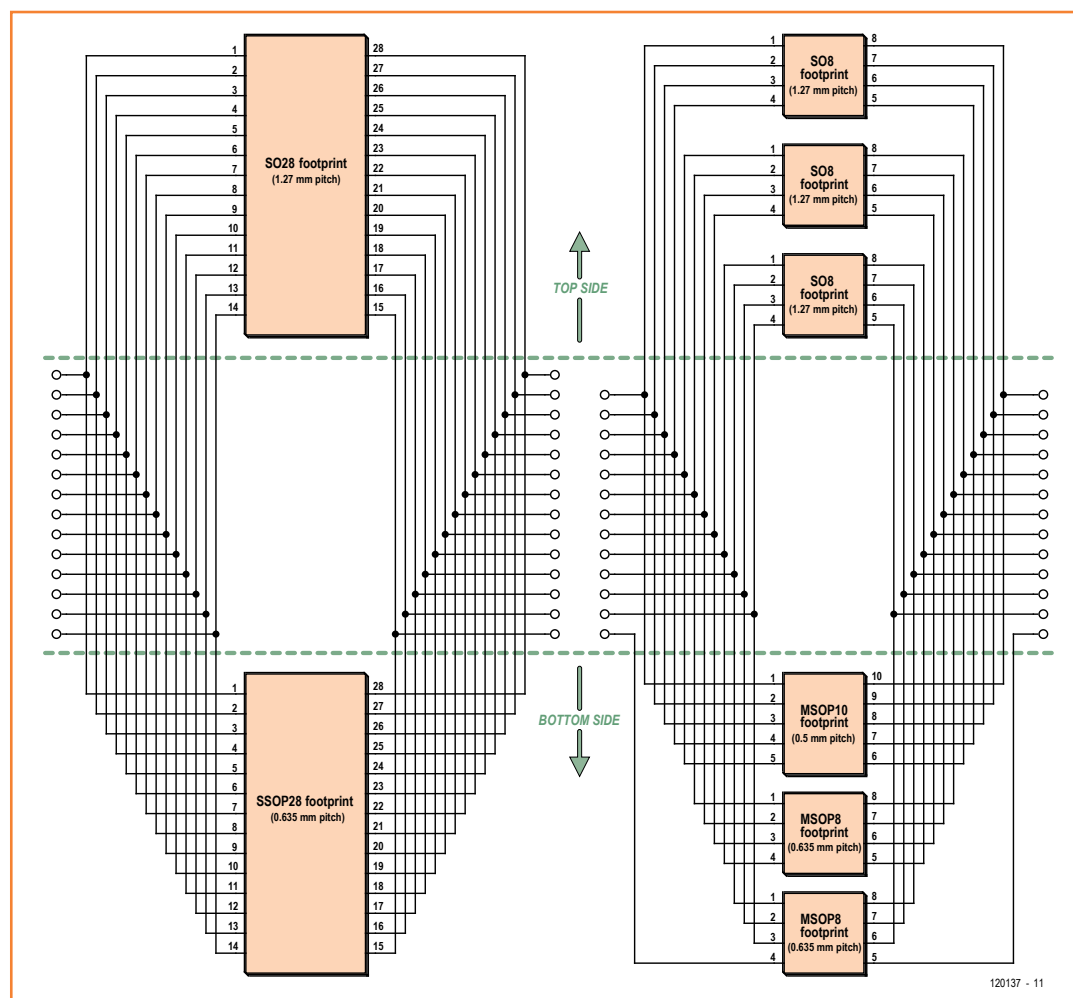


Bild 1.
Dies ist kein normales Schaltbild, da es keine Bauteile aufweist, sondern nur Löt pads. PICO PROTO ist kompatibel mit allen 8-Bit-PICs im 28-poligen Gehäuse.

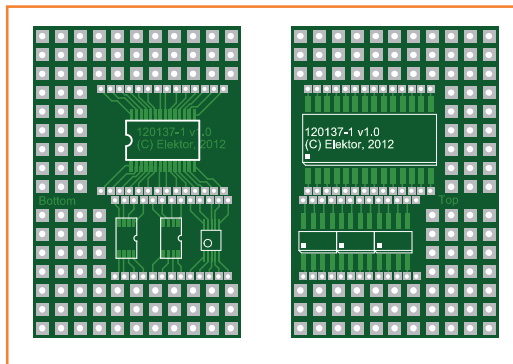


Bild 2.
Die doppelseitige und durchkontaktierte Leiterplatte auf 150 % vergrößert.

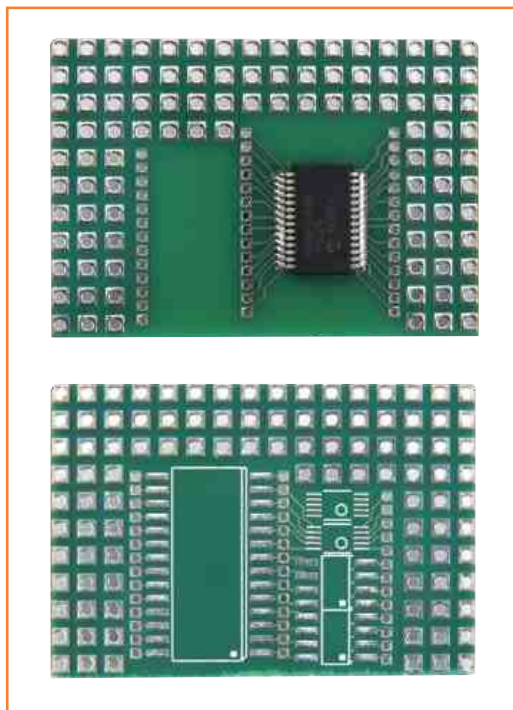


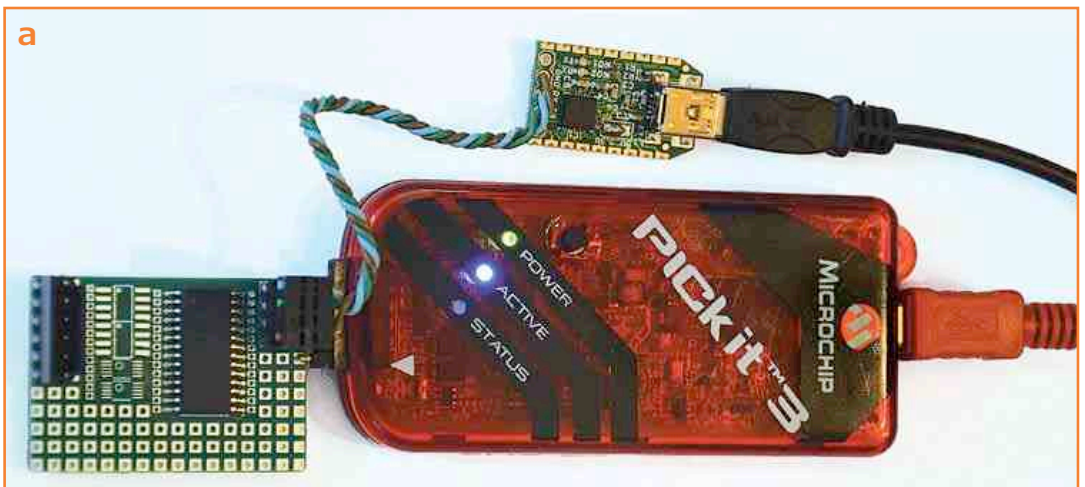
Bild 3.
Ein PICo PROto mit unbestückter Vorderseite und einem Bauteil im SO28-Gehäuse auf der Rückseite.

einen montieren!) wurden Lötflächen für SO8- und MSOP8-Bausteine vorgesehen.

Dies zeigt die Vergrößerung der doppelseitigen Platine in **Bild 2** genauer. **Bild 3** zeigt Vorder- und Rückseite zweier **PICo PROto**. Das Layout weicht noch ein wenig von der endgültigen Version ab; links ohne Bauteile, rechts die Rückseite mit einem TSSOP28-Baustein. Und bitte keine hässlichen Kommentare zur Qualität der Lötarbeiten!

Einsatz von PICo PROto

Zuerst löten Sie den Controller Ihrer Wahl auf - je nach Bauform - Vorder- oder Rückseite. Löten Sie danach eine 5-polige Stiftleiste im 2,54 mm-Raster ein, gegebenenfalls gewinkelt. Diese erlaubt den Anschluss des notwendigen PICKit- oder ICD-Programmers, um Ihre Programme zu flashen und zu debuggen. Verbinden Sie Stiftleiste und Controller wie in **Bild 4** dargestellt. Prüfen Sie jetzt, ob Ihre Entwicklungsumgebung MPLAB den Baustein erkennt. Verbinden Sie dazu den Programmer mit der Platine und Ihrem Entwicklungsrechner. Starten Sie MPLAB, wählen Sie im Menu *Programmer* Ihr Tool und prüfen Sie, ob die Verbindung aufgebaut wird. Angenommen, Sie verwenden PICKit3, dann wird MPLAB jetzt folgenden Fehler anzeigen: *You must connect to a target device to use PICKit*. Dies liegt daran, dass der μC noch nicht mit Spannung versorgt wird. Gehen Sie zu *Programmer* \rightarrow *settings* \rightarrow *Power*, prüfen Sie, ob die für Ihre Anwendung passende Spannung eingestellt ist und klicken das Kästchen *Power target circuit from PICKit3* an. Falls MPLAB eine Fehlermeldung wie *Target Device ID (00005b60) does not match expected DeviceID (00005960)* ausgibt, dann ist



das ein gutes Zeichen, falls *Target Device ID* von Null verschieden ist. Wählen Sie unter *Configure* → *Select device* exakt den eingesetzten Controller-Typ aus und alles sollte in Ordnung sein. Damit ist der handwerkliche Teil beendet!

Falls Sie mit Ihrem Controller über den USB kommunizieren wollen, nehmen Sie doch einfach den BOB-FT232 [2] zu Hilfe. Nach Einlöten und Anschließen von Verbinder K2 kann die Verbindung hergestellt werden; BOB-FT232 übernimmt auch die Stromversorgung Ihrer Platine (im **Bild 4** unten ist die Verbindung PICo PROTo mit BOB zu sehen). In diesem Fall müssen Sie vor dem Anschluss von BOB in MPLAB das Kästchen *Power target circuit from PICKit3* deaktivieren. Auch am BOB muss vorher mittels Jumper JP1 die für Ihre Schaltung passende Versorgungsspannung ausgewählt werden.

Falls Ihnen das Blinken einer LED als Anzeige für den korrekten Programmablauf im Betrieb ausreicht, dann sollten Sie direkt eine rote LED wie im Schaltplan dargestellt anschließen.

Ein gestandener Elektroniker sollte den PICo PROTo in weniger als einer Stunde aufgebaut haben. Der Freiraum auf der Platine steht nun für eigene Versuche zur Verfügung.

Abschließend noch zwei Beispiele (**Bild 5**) für den Einsatz von **PICo PROTo**:

Test eines barometrischen Sensors mit angeschlossener BOB.

Test eines Verfahrens zur Sprachsynthese mit einem PIC18F27J13 auf der Rückseite und einem Klasse-D-Verstärker mit einem SSM2301 auf der Vorderseite.

(120137)

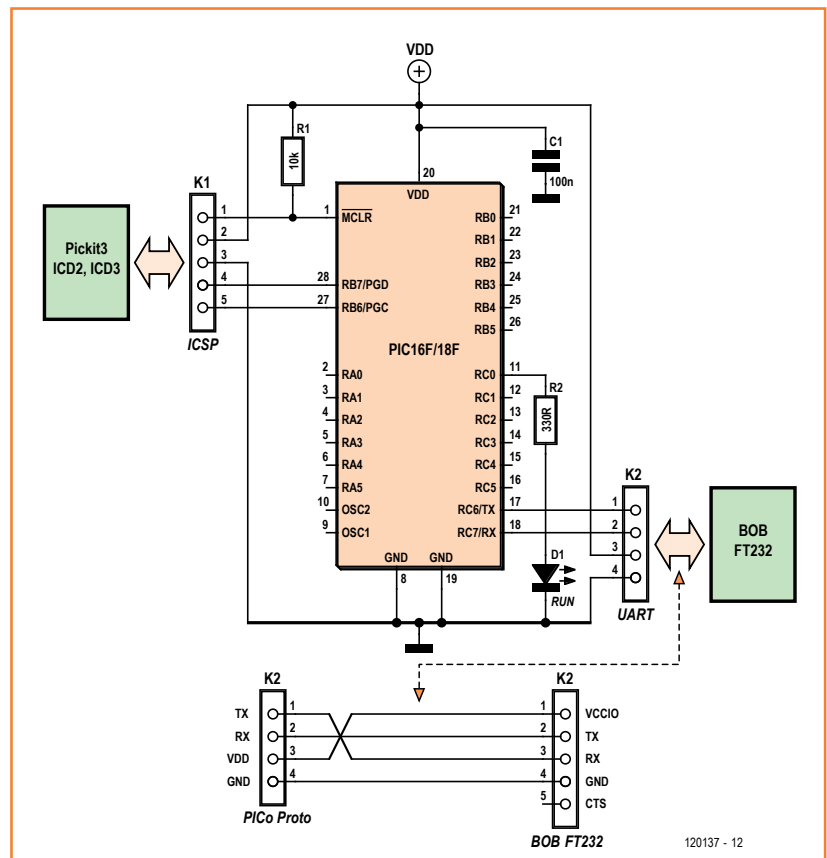


Bild 4.
Anschlussschema eines PICo PROTo.

Weblinks

- [1] www.elektor.de/120137
- [2] USB-Seriell-Adapter BOB-FT232R,
Elektor September 2011,
www.elektor.de/110553

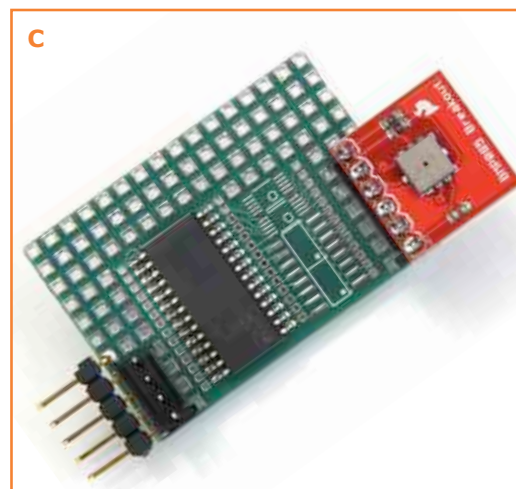
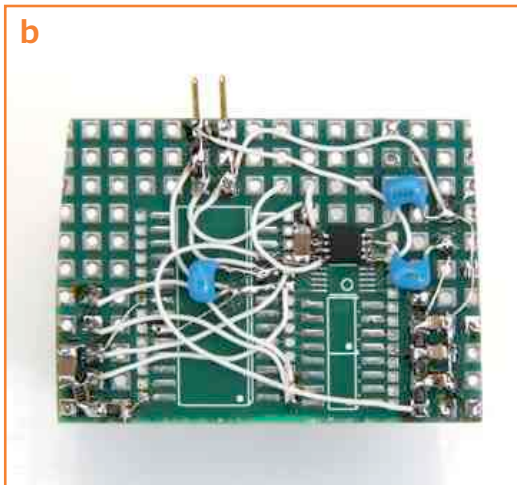


Bild 5
a - Testaufstellung mit
Luftdrucksensor (und BOB).

b - Rückseite mit
PIC18F27J13 im SSOP28-
Gehäuse.

c - Auf der Vorderseite sitzt
ein Klasse-D-Verstärker
SSM2301.

Tools für Vektor-Motorsteuerungen

MotorMind & SigmaBoard

Von
Roland Gehrmann,
Consumer- und Industrie-ICs,
Toshiba Electronics Europe

In der letzten Elektor-Ausgabe wurde das Konzept einer Motorsteuerung vorgestellt, welche auf der Grundlage von Toshiba's TMPM37x-Familie spezieller ARM-Cortex-M3-Controller besondere, Firmware-basierte Funktionen bietet. Nun wird gezeigt, wie Software dabei hilft, BLDC-Motoren effizient, präzise und leise bei einstellbaren Geschwindigkeiten zu betreiben.

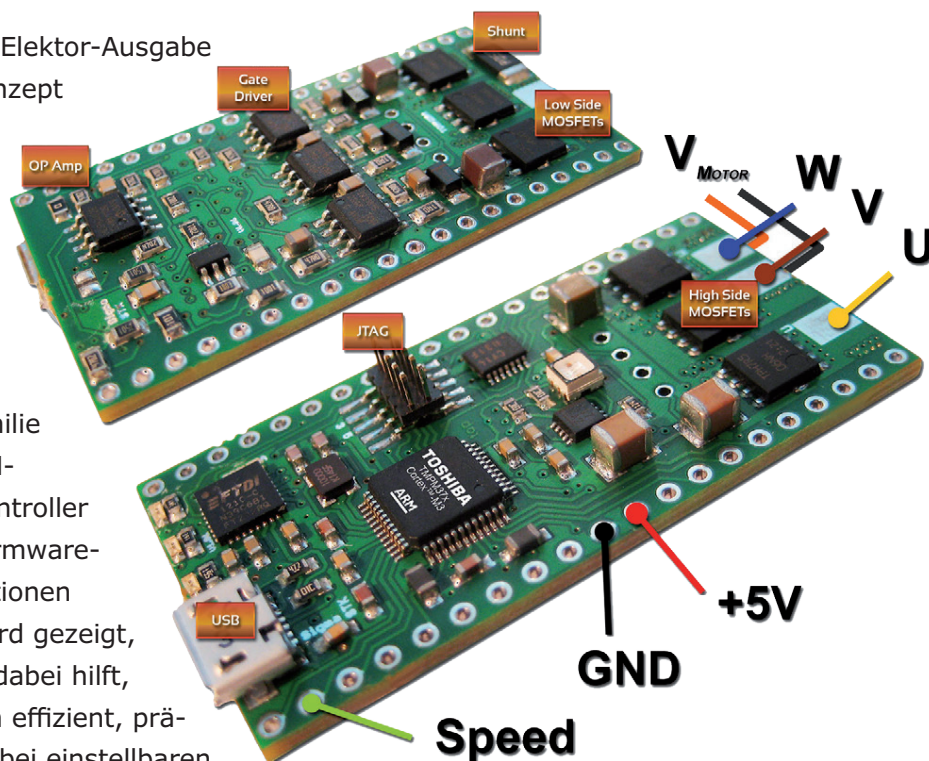


Bild 1. Ober- und Unterseite des SigmaBoards.

Neben PMD-Funktionen (**P**rogrammable **M**otor **D**rive) bieten diese Mikrocontroller eine VE (**V**ector **E**ngine), wodurch der Umgang mit dem komplexen Gebiet der FOC (**F**ield **O**riented **C**ontrol) erleichtert wird. Die Mikrocontroller-Hardware ist nur ein (wichtiger) Aspekt für eine gut funktionierende Steuerung. Selbst mit speziell an die Aufgabe angepasster Hardware bleibt eine intelligente Motorsteuerung eine Herausforderung. Die Verfügbarkeit guter Support-Tools spielt eine große Rolle bei einer erfolgreichen System-Implementierung. Es muss also nicht nur das System an sich passen, sondern auch das Support-Ökosystem stimmen. Für das Prototyping sind Punkte wie die Verfügbarkeit von Referenz-Designs, Star-

ter-Kits und auch von Third-Party-Tools durchaus relevant. Besonders dem letzten Punkt wird Rechnung getragen, wenn man Mikrocontroller mit nichtproprietären Architekturen wählt. Da die Toshiba-Controller auf einem ARM-Cortex-M3-Kern basieren, setzt man damit auf einen Defacto-Industriestandard, für den viele Firmen wie Atollic, IAR oder Keil Unterstützung anbieten.

Entwicklungs-Hardware

Man ist nicht nur auf Third-Party-Tools angewiesen, da die Hersteller von Mikrocontrollern zunehmend in den Support investieren und ihre eigene Entwicklungs-Software und -Hardware anbieten. Ein Beispiel ist das neue, preiswerte

SigmaBoard (**Bild 1**) von Toshiba.

Dieses komplette kleine Entwicklungs-Board misst nur 2,5 x 5 cm und dient nicht nur als Starter-Kit, sondern fungiert gleichzeitig als Referenz-Design. Man kann damit sehr gut FOC- bzw. Vektor-Steuern für bürstenlose Gleichstrommotoren bis zu 36 V / 2 A realisieren. Auf der „digitalen“ Oberseite sitzt ein TMPM373-Mikrocontroller von Toshiba, ein USB/seriell-Wandler mit einem Durchsatz von 115 kbit/s, eine USB-Buchse zum Anschluss an einen PC, eine RGB-LED zur Anzeige der Motorphase, die High-Side-MOSFETs sowie ein U-, V- und W-Anschluss für den Motor. Auf der unteren „analogen“ Seite befinden sich die Gate-Treiber, die Strom-Sensorik mit einem 50-mΩ-Shunt-Widerstand und zugehörigem Verstärker, ein Überstrom-Komparator und die Low-Side-MOSFETs.

Beim TMPM373 handelt es sich um einen 48-poligen SMD-Chip mit stromsparendem, 32-bittigen Controller, der bei einer Betriebsspannung von 5 V mit einer Taktfrequenz von bis zu 80 MHz betrieben werden kann. Der Controller enthält die schon angesprochenen VE- und PMD-Funktionen. Seine Peripherie-Funktionen sind speziell auf die Anwendung in Motorsteuerungen optimiert. Der auf dem SigmaBoard bestückte Controller ist schon mit der Motorsteuerungs-Firmware von Toshiba versehen, was die Entwicklung spezieller Software überflüssig macht. Der Flash-Speicher des Chips unterstützt Rapid Programming. Zusätzlich vereinfacht eine OFD-Einheit (**O**scillation **F**requency **D**etection) die Kompatibilität zu den Sicherheitsstandards nach IEC 60730.

Für maximale Flexibilität gibt es auch noch Anschlüsse zur Ansteuerung einer zusätzlichen Leistungs-Platine auf dem SigmaBoard, so dass man mit der gleichen Firmware und derselben Logik auch Motoren größerer Leistung ansteuern oder andere Anwendungen erschließen kann.

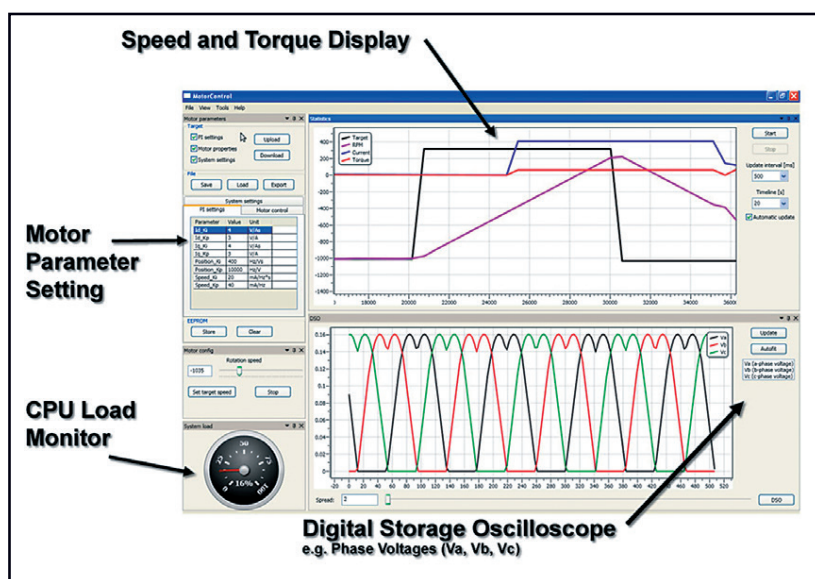
Software-Support

Das SigmaBoard wurde für den Einsatz mit Toshiba's MotorMind-Software konzipiert. Diese freie Software läuft auf einem PC oder einem Android-System und stellt eine intuitive und daher einfach zu bedienende GUI zur Verfügung. Bei einem Android-Smartphone oder Tablet erfolgt die bidirektionale Kommunikation über eine Bluetooth-Verbindung mit dem Board. Das ist nicht nur hoch flexibel, sondern impliziert gleichzeitig eine galvanische Trennung. Dies ist vor allem beim Ausprobieren von Motoren für höhere Span-

nungen sinnvoll, hilft aber auch beim Testen von Antrieben für E-Bikes im realen Einsatz.

Mit MotorMind kann man alle relevanten Motor-Parameter konfigurieren. Man kann z.B. die Drehzahl ändern und dabei die Reaktion auf dem grafischen Display in Form von Ziel-Drehzahl, aktuelle Drehzahl und Drehmoment in Echtzeit verfolgen. Die Software bietet sogar bestimmte DSO-Funktionen (**D**igital **S**torage **O**scilloscope) mit erweiterten Trigger-Möglichkeiten.

Mit MotorMind ist ein angepasstes Setup eines Motors in einer Stunde machbar. Einmal konfiguriert kann das SigmaBoard dann als fertige Motorsteuerung ohne PC fungieren. Für einen kompletten



Antrieb reicht daher die Kombination von Motor, Stromversorgung und SigmaBoard. Über ein externes analoges Signal kann man dann die Drehzahl einstellen. Es gibt noch einen PWM-Speed-Eingang, Fehler-Erkennung und -Anzeige und außerdem können auch Tachosignale angeschlossen werden, was aber nicht unbedingt erforderlich ist.

MotorMind-Fähigkeiten

Mit MotorMind lassen sich Motor-Parameter wie Pol-Paare, Strom, Drehrichtung, Beschleunigung und Encoder-Details eingeben und auf das SigmaBoard laden. Es gibt Einstellungen für PI-Regler (**P**roportional-**I**ntegral) und System-Parameter wie Totzeit, PWM-Frequenz und Shutdown/Restart-Verhalten – alles schön grafisch mit GUI. Es können auch Einstellungen vom SigmaBoard gelesen werden.

Bild 2.
Anzeige von Werten
der Motorsteuerung in
MotorMind.

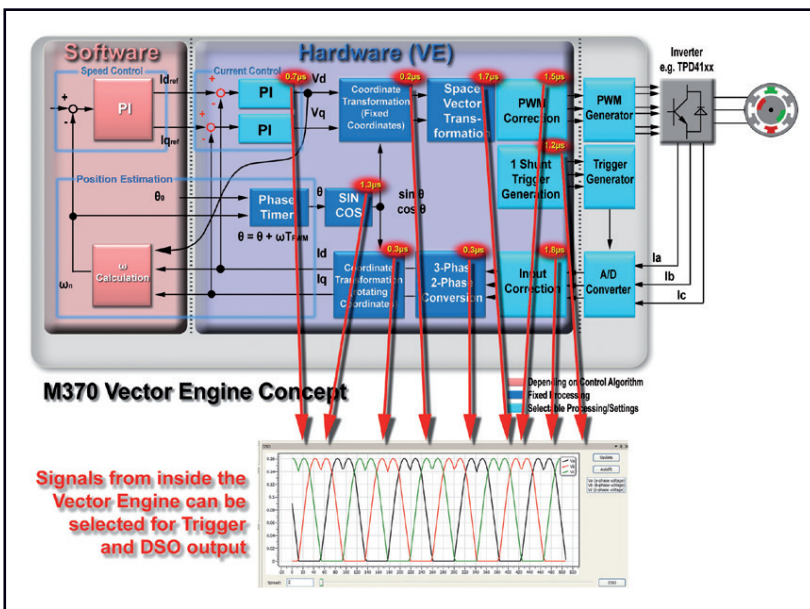


Bild 3.
Die Vector Engine des Mikrocontrollers.

Die Software hat diverse Fenster, darunter sogar eine Anzeige, die über die Auslastung des Controllers informiert. Dann gibt es noch ein Fenster zur Motor-Konfiguration, das neben einem Start/Stop-Knopf einen Schieberegler zur Einstellung der Drehzahl enthält. Außerdem gibt es noch zwei größere Fenster: eines mit einer aktuellen Anzeige von Drehzahl, Drehmoment und Strom relativ zur Zieldrehzahl sowie das DSO-Display. Die DSO-Funktion erfasst Signale mit einer Auflösung bis zu 50 μ s bei einer PWM-Frequenz von 20 kHz. Bis zu acht VE- und Firmware-Parameter aus einer Liste von 32 möglichen Signalen können gleichzeitig dargestellt werden. Darunter sind die Spannungen und Ströme der Pole, Ströme für d- und q-Achsen, Proportionalkoeffizienten und andere interne Signale der integrierten VE (siehe **Bild 2**). Man kann wie bei einem „richtigen“ DSO die Trigger-Quellen und Trigger-Bedingungen wie fallende oder ansteigende Flanken, mittlerer oder linksseitiger Trigger und die Trigger-Schwelle einstellen (siehe **Bild 3**). Zusätzlich ist es noch möglich, jeden n-ten Event zu loggen. Mit $n \leq 256$ können damit auch längere Abschnitte aufgezeichnet werden. Die Tick-Dauer und die gesamte Aufzeichnungszeit werden unten am Fenster eingeblendet.

Fazit

Generell unterstützt MotorMind einen Entwickler besonders dadurch, dass man beobachten kann, wie Controller-Auslastung, Motorverhalten und

Werte bei Änderungen einzelner Parameter in Echtzeit reagieren. Es lassen sich so gut Betriebsarten wie Beschleunigen und Bremsen beurteilen. Parameter können leicht in eine Header-Datei exportiert werden, die dann nachfolgend in die Firmware eingebunden werden kann. Die Funktionen zum Sichern und Einlesen von Parametern erlauben die Verwendung der Software auch mit unterschiedlichen Motor-Anwendungen.

Last not least: Der Wettbewerb [2] von Toshiba und Elektor zu Motorsteuerungen läuft noch. Machen Sie mit!

(120725)

Weblinks

- [1] Support:
<http://forum.toshiba-components.com/forum.php>
- [2] Toshiba-Wettbewerb:
www.elektor-labs.com/M370-contest
- [3] Demo-Video zu MotorMind:
www.youtube.com/watch?v=iqhsvtEMMEK

Design-Wettbewerb mit Toshiba M370 BLDC-Motor-Controller mit M370 - attraktive Preise zu gewinnen

Seit dem 26. Februar 2013 läuft ein hochinteressanter Schaltungswettbewerb von Toshiba in Kooperation mit Elektor. Prämiert werden innovative Steuerungen für BLDC-Motoren auf der Basis des SigmaBoards. Wir sind sehr gespannt auf neue Ideen!

Es gibt interessante Sachpreise zu gewinnen:

1. Preis: Toshiba Multimedia-Notebook der Spitzenklasse mit 15"-Display
2. Preis: Toshiba Tablet AT-300
3. Preis: 60 GB SSD PC Upgrade Kit – Interface SATA 6.0 Gbit/s, Size 2.5-inch (6,4 cm)
4. Preis: 8 GB Flash-Air-SD-Card
- 5.-10. Preis: 64 GB USB-Stick

Der Wettbewerb wird bis zum 31. Juli 2013 dauern. Mehr Infos zum Wettbewerb gibt es unter: www.elektor-labs.com/M370-contest

ES WURDE SOCLUTIONS GEBOREN



soclutions™
UNIQUE SOLUTIONS ON A SINGLE CHIP


- SOCLUTIONS EIN HOHES INTEGRATIONSNIVEAU LÖSUNG
- SICHERHEIT IN IHRE KOMMUNIKATION UND SPEICHER
- WIR HELFEN IHNEN UM IHRE LÖSUNGEN SICHERZUSTELLEN

Wir kommen gleich ...

besuchen sie uns in: www.soclutions.com

UNSER WICHTIGSTES WERKZEUG

LEITERPLATTEN WEITER GEDACHT.



SCHNELLE VIELFALT IST BESSER!
Online kalkulieren. Online bestellen.

LEITON
RECHNEN SIE MIT BESTEM SERVICE

Wir denken weiter. Als lösungsorientiertes Unternehmen möchten wir unsere anspruchsvollen Kunden immer wieder mit neuen Ideen und sinnvollen Innovationen begeistern. Als ein führender Online-Anbieter von Leiterplatten bieten wir erstklassige Leistungen in den Bereichen **Service, Fairness, Technologie, Auswahl, Komfort, Geschwindigkeit und Zuverlässigkeit**. Unsere Webseite bietet eine enorme Auswahl an Lösungen in der Online-Kalkulation sowie weiterführendes Wissen, Informationen und Entwicklerwerkzeuge. Und sollte einmal etwas sehr Ausgefallenes gefordert sein, finden wir gerne **Ihre persönliche Lösung**. Sie können bei LeitOn immer mit bestem Service rechnen.

www.leiton.de Info-Hotline +49 (0)30 701 73 49 0

Workshop-DVD **Android** Programmierung und Hardware-Steuerung

Konnten Sie aus zeitlichen Gründen nicht an unserem ElektorLive-Seminartag am 20.10.2012 in Dortmund teilnehmen und haben somit auch die beiden populären Android-Seminare verpasst!? Kein Problem! Mit dieser neuen DVD-Video sind Sie jetzt trotzdem live dabei!

Entwickler und Buchautor Stefan Schwark stellt in seinem Seminar die Grundlagen der Programmierung eigener Android-Applikationen dar. Anhand von kurzen Programmen erklärt er den Umgang mit der Entwicklungsumgebung „Eclipse“ und dem Android SDK. Der Schwerpunkt liegt hierbei auf Android-Apps, die einfache Funktionen ausführen und so auch für den Anfänger nachvollziehbar sind. Das Spektrum reicht von einfachen Rechenprogrammen über Applikationen zur Abfrage der Hardware des Android-Gerätes bis zu Programmen für die Kommunikation über das Internet.

Im zweiten Seminar zeigt Bernhard Wörndl-Aichriedler (Entwickler des „AndroPod“-Moduls) in einem Crash-Kurs, wie Sie Ihr Android-Gerät als Schaltzentrale in eigenen Projekten benutzen können.



ISBN 978-3-89576-272-7

€ 24,90 • CHF 30,90



Weitere Infos & Bestellung unter
www.elektor.de/android-dvd

Sorry about my English

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

Haben Sie sich schon einmal verzweifelt durch Online-Diskussionsgruppen zum Thema Elektronik gekämpft, auf der Suche nach dem goldenen Tipp, der ihre nicht funktionierende Schaltung zum Laufen bringen könnte? Wahrscheinlich sind Sie dann nach dreiviertel der siebten Seite auf einen Satz wie „sorry about my english“ oder ähnliches gestoßen. Ich habe so etwas auch schon öfter auf der Elektor.Labs-Seite gesehen.



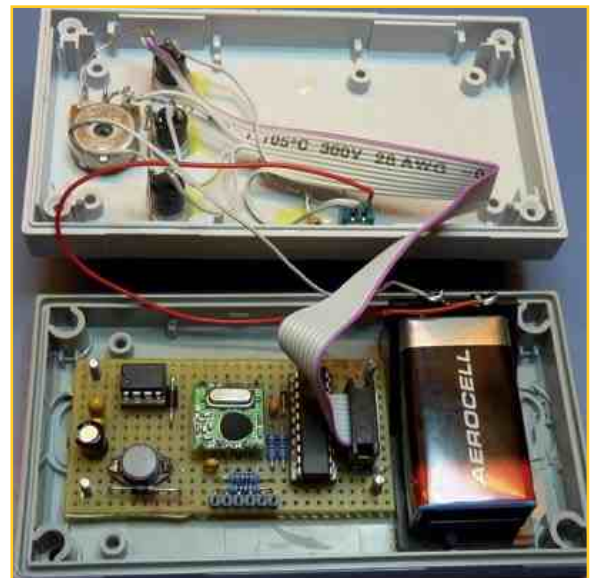
Dear Poster, let me tell you this, as the moderator of the .labs site I don't give a [🎵] [🎵] about your English, you just saved my [🎵] day, maybe my [🎵] job (sorry for my English ;-). Also: Posten Sie weiterhin interessante Sachen und machen Sie sich keinen Kopf über Grammatik oder Interpunktion. Ich bin sicher, dass die meisten Menschen Sie verstehen, wenn Sie über spannende Dinge schreiben...

EDITOR'S CHOICE

Mutiger Elektronik-Ingenieur verhindert Bahnkrieg

Weil zwei konkurrierende Fahrdienstleiter nur widerwillig die Kontrolle des jeweils anderen Zuges übernehmen und dies zu groben Unhöflichkeiten zwischen den Parteien führt, beschloss Elektroniker waltro, dass damit Schluss sein müsse. Statt die andauernden Feindseligkeiten zu ignorieren, fand er, dass es an der Zeit sei, dagegen etwas zu tun. Eines Nachts, als beide verfeindeten Parteien schliefen, brach unser tapferer Ingenieur in die Züge ein und ersetzte ihre alten Kontrollsysteme durch eine clevere Eigenproduktion. Am nächsten Morgen war der Frieden dank der Multi-Channel-Fähigkeiten der neuen Regelung hergestellt. Well done, waltro!

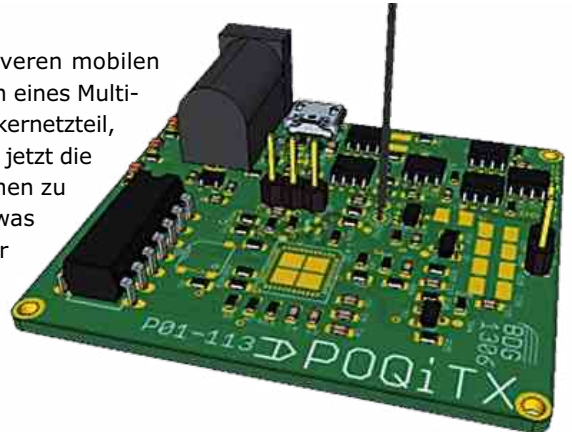
www.elektor-labs.com/node/3020



Perfektes Projekt vom schlechten Titel ermordet!

OP BifrostDevGrp hatte glücklich die Entwicklung einer kleinen und cleveren mobilen geregelten batteriebetriebenen 3,3 V / 5 V-Stromversorgung einschließlich eines Multi-Source-Laders abgeschlossen, deren 950 mAh-Li-Ion-Zelle mit einem Steckernetzteil, von einem USB-Anschluss und sogar drahtlos geladen werden konnte. Und jetzt die Katastrophe: Der OP beschloss, seinem Projekt einen gar zu lustigen Namen zu geben und es auf der Elektor.Labs-Webseite zu posten. Er ahnte nicht, was er da angerichtet hatte. Wenn doch wenigstens die ersten Zeilen seiner Beschreibung fesselnder gewesen wären... Aber so geriet das Projekt in Vergessenheit. Zum Glück halten wir Ausschau nach solchen Projekten und konnten es gerade noch retten!

www.elektor-labs.com/node/2969



Der beste Low-Power-Röhrenverstärker aller Zeiten

Leidenschaft (und manchmal auch Geld) ist der Schlüssel zu großen Dingen. OP Ken hat eine Passion für High-end-Audio und startete ein Projekt, um den optimalen Low-Power-Röhrenverstärker zu entwerfen. Der OP hat eine Menge Vorarbeit geleistet und eine Auswahl von Teilen und Materialien getroffen, aber noch nicht alle Details geregelt. Dazu braucht er nun Ihre Hilfe. Spüren Sie auch den Audiophilen in sich? Möchten Sie auch ein Teil des Besten aller Zeiten sein? Seien Sie ehrlich, wer würde das nicht? So teilen Sie Ihre Meinung zu Trafos, Kondensatoren und anderen, den Klang bestimmenden Bauteilen mit!

www.elektor-labs.com/node/2949

Müllkalender zu kompliziert

Ich habe bisher nie einen Müllkalender meiner Stadt gebraucht, ich wusste fast instinktiv, welche Tonne an welchem Tag vor die Tür gestellt werden musste. Jetzt hat sich das Schema geändert und ich komme nicht mehr klar. Was ich brauche, ist eine Gedächtnisstütze! MarkDonner hat eine Idee für eine solche Vorrichtung auf .Labs gepostet. Und dann schloss sich Amigaman an und berichtet von einem TTL-Grab aus den 80ern, das die gleiche Funktion zur Verfügung stellte. Ich entdeckte, dass ich nicht allein bin. Es gibt noch andere Leute mit der gleichen Erkrankung und jetzt haben wir auch unsere eigene Support-Gruppe. Sind Sie auch auf der Suche nach Hilfe? Dann begleiten Sie uns und arbeiten mit am elektronischen Müllkalender!

www.elektor-labs.com/node/2946



(130090-I)

www.elektor-labs.com

Anmerkung: OP steht für Original Poster, der Person, die ein Online-Projekt oder eine Diskussion beginnt. OPs, die die Chance wahren wollen, dass ihr Projekt in Elektor veröffentlicht wird, sollten immer darauf achten, dass ihre E-Mail-Adresse für den Zugang zu Elektor.Labs korrekt ist. Nur so können wir auch Kontakt mit Ihnen aufnehmen.

Fallstricke in der Platinenproduktion

Von **Thijs Beckers**
(Elektor-Redaktion)

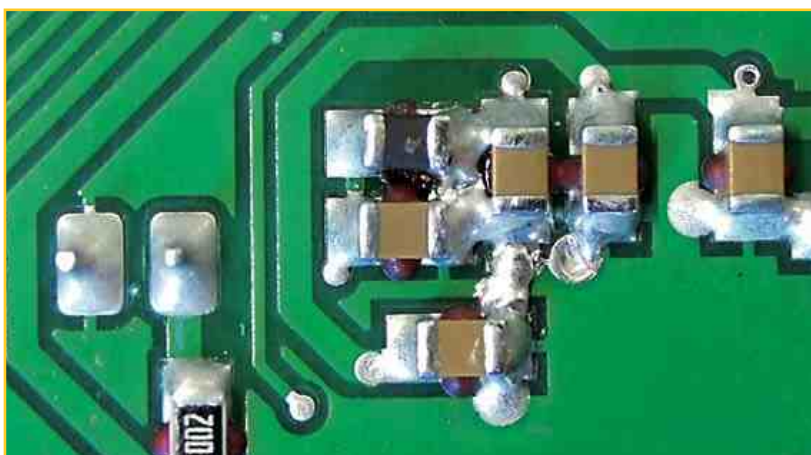
Reflow- oder Wellenlöt ist eine Technik, die in Produktionslinien komplette Leiterplatten sehr schnell verzinnt. Dabei wird die Platine über eine „Welle“ von flüssigem Lot bewegt, wodurch alle Pads und Bauteile auf einer Seite der Leiterplatte sehr schnell verzinnt werden. Das Internet zeigt in tausenden von Bildern und kurzen Videos, wie das in der Praxis funktioniert. Dieser Prozess wird in der Industrie seit vielen Jahren verwendet und sollte seine Praxistauglichkeit ausreichend bewiesen haben. Aber es gibt einige Dinge, die beim Reflow-Löten einfach nicht vermieden werden können. Werfen Sie einen genauen Blick auf das erste Bild. Das Lot am linken Ende des SMD-Kondensators weist eine viel steilere Rampe als das am anderen Ende auf. Und das nicht nur, weil sich an einem Ende eine Durchkontaktierung und folglich kein Lötstopplack befindet.

Wenn sich die Leiterplatte durch die Lötmedium-Welle bewegt, gelangt in der Bewegungsrichtung etwas weniger Lot auf die vordere und etwas mehr auf die Rückseite des Bauteils. Genau wie ein Kometenschweif in Richtung der Sonne zeigt der Lot-

„Schwanz“ in die Bewegungsrichtung der Lot-Welle. Dieser Effekt kann auch am SMD-Widerstand ganz links beobachtet werden. Offenbar bewegte sich die Lotwelle von links nach rechts. Unglücklicherweise führt dies zu einigen Beschränkungen beim Entwurf des Platinenlayouts. Schauen Sie einmal auf die zweite Aufnahme der Platine. Die Löttrichtung ist die gleiche wie bei der ersten. Haben Sie schon das Problem erkannt?

Auf der linken Seite der Induktivität (obere schwarze Komponente) und des Kondensators (beige Komponente darunter) sieht man die miteinander durch das Platinenlayout verbundenen Pads. Aber auf der rechten Seite gibt es einen offensichtlichen Kurzschluss zwischen den Pads, wodurch die Bauteile nutzlos werden und die ganze Schaltung nicht mehr funktionieren kann (in diesem Fall durch einen massiven Kurzschluss der Spannungsversorgung nach Masse). Wie konnte das passieren? Sicher, es gibt eine Lötstopplacke zwischen den Pads, die nun unter einem Klumpen von Lot begraben ist. Aber dies hat nicht den Kurzschluss verhindert. Denn hier zeigen sich nämlich die Grenzen des Reflow-Lötens. Alles wäre gut gewesen, wenn der Kondensator nicht direkt neben der LC-Entkopplung platziert wäre, die wie eine Wand für die Lötwellen wirkt und eine „Falle“ für das Lötzinn zwischen diesen drei Bauteilen darstellt. Kein Lötstopplack würde dies verhindern! Die einzige Lösung besteht darin, die Platine so zu routen, dass die Bauteile ausreichend weit voneinander entfernt sind. Oder Sie wechseln zu einem alternativen Lötverfahren (Lötkolben?). Natürlich wäre es am besten, wenn der Platinendesigner von vornherein wüsste, mit welchem Verfahren die Platine gelötet werden wird, so dass er diese Art von Fallstricken vermeiden könnte.

Und wenn Sie sich fragen, was diese rötlichen Ausbuchtungen zu bedeuten haben, die unter den Bauteilen hervorquellen: Klebstoff, der die Komponenten während des Lötvorgangs in Position hält. Die Platinenoberfläche, die Sie auf den Bildern sehen, ist eigentlich die Unterseite. Während des Reflow-Lötens stehen die SMDs immer auf dem Kopf. Wenn man sie nicht auf die Platine kleben würde, so verschwänden sie in der Lötwellen. Doch keine Sorge, die Mitarbeiter an der Platinenbestückungsanlage kümmern sich darum ;).



(130027)

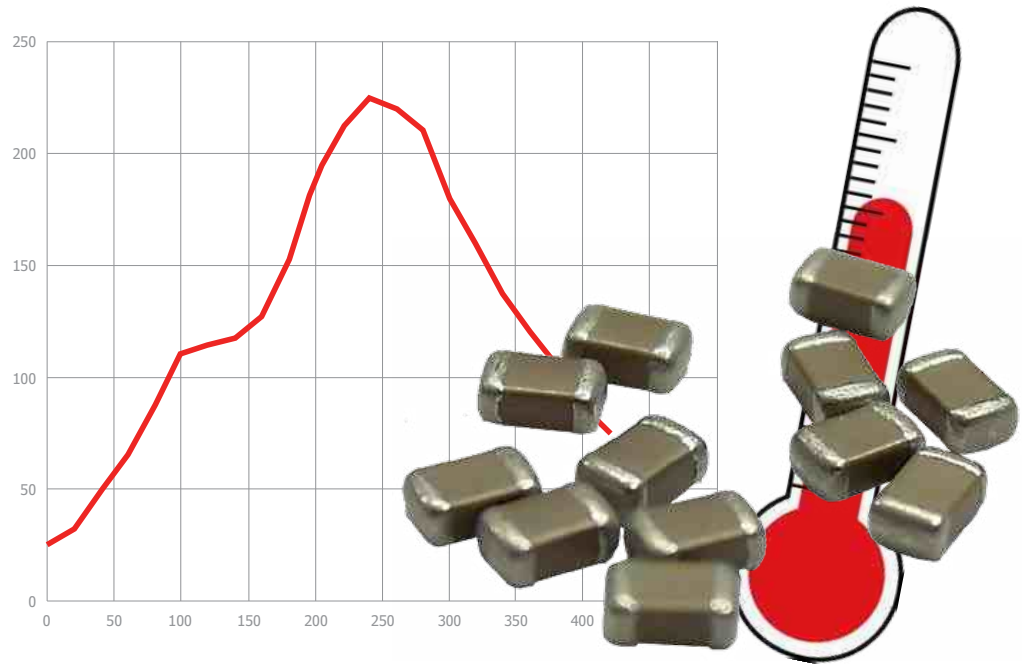
Von den Machern von Elektor!



Oder frei Haus unter www.elektor.de/roehren9 bestellen!

Keramikkondensatoren voll daneben!

Von **Thijs Beckers**
(Elektor-Redaktion)



Immer ist was los im Elektor-Labor! Bei der Montage des Prototyps unseres brandneuen *500 ppm LCR-Meters* lernte mein Kollege Jan schmerzlich eine oft übersehene Eigenschaft von Keramikkondensatoren kennen. Es passierte, als er keramische Kondensatoren mit einem Wert von 150 pF benötigte. Taufrisch kamen sie aus dem Verpackungsmaterial auf den Labortisch und wurden mit unserer bewährten SMD-Rework-Station auf die Platine gelötet. Aber später, während des Kalibriervorgangs, traten Probleme auf und Jan konnte das LCR-Meter nicht korrekt abgleichen. Jan ist es gewohnt, dass Bauteile nicht das tun, was man von ihnen erwartet, und überprüfte den Keramikkondensator, der für die Kalibrierung eingesetzt wurde. Und er maß einen Wert von 157 pF statt der versprochenen 150 pF. Dies erwies sich bei der Kalibrierung als zu hoch. Dann hat Jan ein paar weitere Kondensatoren aus der Lieferung vermessen: Der Nennwert wurde weit überschritten, alle gemessenen Kondensatoren wiesen Werte zwischen 168 pF und 183 pF statt 150 pF $\pm 5\%$ auf.

Dies veranlasste uns zu einem kleinen Test. Wir haben eine Reihe von Kondensatoren dem Standard-Verfahren - Reflow-Löten im SMD-Ofen - ausgesetzt und die Werte vor und nach dem Erhitzen verglichen. Es stellte sich heraus, dass ihre

Werte vor der großen Hitze durchweg zwischen 170 pF und 180 pF lagen, aber danach waren sie deutlich gesunken, fast genau auf den richtigen Sollwert. Zum Beispiel wurden aus 178 pF 156 pF, aus 173 pF 152 pF.

Bis zur Deadline für diesen Artikel ist es uns nicht gelungen, das Phänomen zu erklären, aber wir vermuten, dass das Dielektrikum dabei eine entscheidende Rolle spielt. Wenn Sie eine Idee haben, was der genaue Grund sein könnte, schreiben Sie bitte an editor@elektor.com (Betreff: ceramicz). Als Belohnung für Ihre Bemühungen und eine sinnvolle Antwort halten wir ein *mbed LPC1114 Kit* [1] von NXP bereit und veröffentlichen Ihre Klarstellung in einer späteren Ausgabe. Viele unserer Leser sind ja erfahrene Ingenieure und wissen, dass bestimmte (Erhitzungs-)Protokolle in den Produktionsprozess eingebunden sind, aber wenn Sie ein Newbie sind und/oder keinen Zugriff auf professionell abgeglichenes Baumaterial haben: Achten Sie auf temperaturkritische Komponenten und schenken Sie ihnen die erforderliche Aufmerksamkeit!

(130118)

[1] <http://mbed.org/handbook/mbed-NXP-LPC1114>

ECD 7 Bauteilbibliothek mit über 75.000 Komponenten



NEU

Diese CD-ROM bietet Ihnen acht Datenbanken für ICs, Germanium- und Silizium-Transistoren, FETs, Thyristoren, Triacs, Dioden und Optokoppler. Weitere elf Anwendungen zur Berechnung von Vorwiderständen bei LEDs, Spannungsteiler, Ohmsches Gesetz sowie Farbcodeschlüssel für Widerstände und Induktivitäten etc. runden das Programmpaket ab.

Jede Datenbank zeigt für (fast) jedes Bauelement eine Gehäuseskizze, die Anschlussbelegung, die technischen Daten (soweit bekannt) und verfügt über eine Suchroutine nach Bauteil-Parameter.

Alle genannten Datenbank-Anwendungen sind interaktiv, d. h. Sie können Bauteile hinzufügen, ändern oder ergänzen.

ISBN 978-90-5381-298-3 · € 29,50 · CHF 36,60



Weitere Infos &
Bestellung unter
www.elektor.de/ecd7



FRONTPLATTEN & GEHÄUSE

Kostengünstige Einzelstücke
und Kleinserien

Individuelle Frontplatten können mit dem Frontplatten Designer mühelos gestaltet werden.

Der Frontplatten Designer wird kostenlos im Internet oder auf CD zur Verfügung gestellt.

- Automatische Preisberechnung
- Lieferung innerhalb von 5-8 Tagen
- 24-Stunden-Service bei Bedarf



Preisbeispiel: 34,90 €
zzgl. Ust./Versand

Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 8058695-0
Fax +49 (0)30 8058695-33 · Web info@schaeffer-ag.de · www.schaeffer-ag.de

ARM-Mikrocontroller 2 30 Projekte in C für Fortgeschrittene

Die Projekte in diesem Buch sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Das heißt nicht, dass diese Projekte einfach sind. Sie sind aber einfach zu verstehen. Es wird beispielsweise die USB-Verbindung zur Kommunikation benutzt, eine Methode, die im mbed-Board so einfach integriert ist, dass sie sich auch für ein Einsteiger-Buch eignet.

Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung. Es bedeutet, dass man keinerlei Software auf seinem PC installieren muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC. Sie können von jedem beliebigen Ort der Welt auf Ihr Projekt zugreifen und daran weiterarbeiten. Wenn Sie fertig sind, genügen ein paar einfache Mausklicks, um Ihr Programm auf das mbed-System zu übertragen. Natürlich können Sie die Projekte auch auf Ihren eigenen PC laden und dort speichern.

Die Quelltexte zu den Beispielprogrammen stehen gratis unter www.elektor.de/arm-buch2 zum Download bereit. Das zum Buch gehörige Hardware-Starterkit kann unter www.elektor.de/arm-kit2 geordert werden.

NEU



243 Seiten (kart.) • Format 17 x 23,5 cm

ISBN 978-3-89576-271-0

€ 39,80 • CHF 49,40

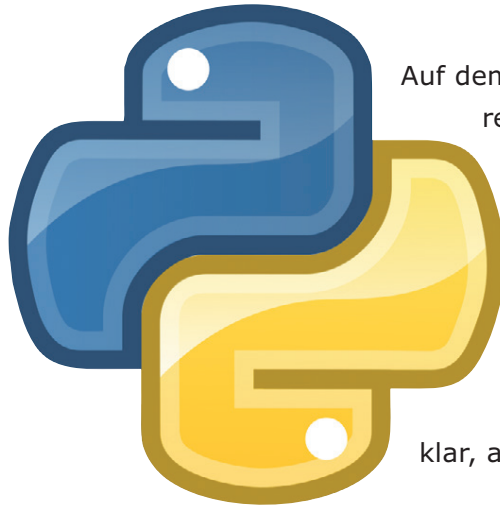


Weitere Infos & Bestellung unter
www.elektor.de/arm-buch2

Von BASIC nach Python (1)

Ein Erfahrungsbericht

Von
Jean-Claude Feltes
(LU)



Auf dem ersten PC des Autors aus den 80er Jahren war zufälligerweise QuickBASIC installiert. Dies führte zu einer intensiven Beschäftigung mit der Programmierung. Später kam das allseits bekannte Visual Basic ins Spiel. Mit der Hinwendung zu Linux musste dann ein guter Ersatz für BASIC her. Und das wurde ein richtiger Umstieg, denn Python ist schön knapp und klar, aber auch anders...

Vielleicht fragen Sie sich, was die Besprechung einer Programmiersprache in Elektor zu suchen hat. Nun, Python eignet sich z.B. gut zur hardwarenahen Programmierung kleiner Rechnersysteme wie z.B. Raspberry Pi. Python und Elektronik gehen da eine durchaus enge Verbindung ein. Ursprünglich brachte mich ein Kollege auf Python. Diese Programmiersprache und ihr Konzept gefielen mir sofort, obwohl es recht große Unterschiede zu Visual Basic gibt. Ein Python-Quelltext beeindruckt durch Knappheit und Klarheit – er kommt ganz ohne Klammern und Strichpunkte aus, die zumindest mich bei Pascal, C und Java schon immer gestört haben. Doch es gibt noch weitere Unterschiede.

BASIC versus Python

Ein wichtiger Punkt ist: Python ist eine reine Interpreter-Sprache, es gibt keinen Compiler dafür. Dies ist sowohl Vorteil als auch Nachteil. Ein Python-Programm ist zunächst einmal langsamer als ein kompiliertes Programm, doch die leistungsstarken Bibliotheken bewirken, dass man davon nicht viel spürt. Weiter erfordern Python-Programme, dass auf dem Zielsystem der Interpreter plus eventuell benötigte Module installiert sein müssen.

Von Vorteil ist, dass ein Programm schnell geändert werden kann, wenn man z.B. eine andere

Schnittstelle ansprechen will. Für kleinere Hacks kann man leicht ganz auf aufwendige Benutzerschnittstellen verzichten und Variablenwerte direkt im Code setzen. Bei interpretiertem Code kann man sogar zur Laufzeit Funktionen definieren, um beispielsweise einen Funktionsplotter zu programmieren.

Python ist sehr stark objektorientiert. Wenn es um einfache Programme geht, dann ist das weniger wichtig. In der Literatur und den Beispielprogrammen wird davon aber gerne und ausgiebig Gebrauch gemacht. Man würde Vieles nicht verstehen, wenn einem das objektorientierte Konzept fremd bleibt.

Gewöhnungsbedürftig (aber praktisch) ist der Minimalismus: Code-Blöcke werden nicht durch Klammern oder „begin“ bzw. „end“ gruppiert, sondern durch schlichte Einrückung des Quelltextes. Wie der Vergleich im **Kasten C, BASIC und Python** zeigt, führt dies zu sehr übersichtlichem und kurzem Code.

Weitere Unterschiede: Im Gegensatz zu BASIC wird zwischen Groß- und Kleinschreibung unterschieden. Schön für Elektroniker ist, dass Python mit komplexen Variablen umgehen kann. Es gibt passende Interpreter für Windows, Linux und OS X.

C, BASIC und Python

C	Quick/Visual Basic	Python
<pre>#include <math.h> #include <stdio.h> int main(int argc, char *argv[]) { printf ("Hello World\n"); int i; int x; for (i=0; i<11;i++) { if(i%2==0) { x = pow(i,2); printf ("%d ^2 = %d \n",i,x);} else { x = pow(i,3); printf ("%d ^3 = %d \n",i,x);} } return 0; }</pre>	<pre>Print "Hello world!" For x = 1 To 10 If x Mod 2 = 0 Then Print x; "^2 = "; x^2 Else Print x; "^3 = "; x^3 End If Next x</pre> <p>Bei VB muss Print durch Debug.Print ersetzt werden, und der Code muss z.B. in einem Sub Form_Load() stehen.</p>	<pre>print "Hello world!" for x in range(0,11): if x % 2 == 0: print x, "^2 = ", x**2 else: print x, "^3 = ", x**3</pre>

Installation

Wenn man sich den für das eigene OS passenden Interpreter installiert, dann nimmt man dabei am besten auch einige sinnvolle Zusatzmodule mit. Es stellt sich die Frage: Python 2.x oder 3? Für Version 3 fehlen leider noch wichtige Bibliotheken. Hinzu kommt, dass es einen Backport von Verbesserungen in Version 2.7 gab, weswegen man auch ganz gut damit leben kann. In **Tabelle 1**

sind interessante Module nebst Download-URLs aufgelistet.

Die Windows-Varianten verfügen über bequeme Installer. Auf anderen Systemen geht man wie folgt vor:

Download (Archiv) in ein temporäres Verzeichnis entpacken
in die Kommandozeile eingeben: *python setup.py install*

Tabelle 1: Module und Download-Links

Python 2.7 Interpreter	www.python.org/download/ Auf Linux-Systemen ist Python schon installiert.
Numpy Wissenschaftliche Zahlenverarbeitung etc.	http://pypi.python.org/pypi/numpy
Matplotlib Diagramme	http://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.1.0/
PySerial Zugriff auf die serielle Schnittstelle	http://sourceforge.net/projects/pyserial/files/
PyParallel Zugriff auf die parallele Schnittstelle	http://sourceforge.net/projects/pyserial/files/
PyUSB USB-Modul	http://sourceforge.net/projects/pyusb/
WxPython Erstellung von GUIs	www.wxpython.org/download.php
Geany Editor mit Syntax-Highlighting	www.geany.org/

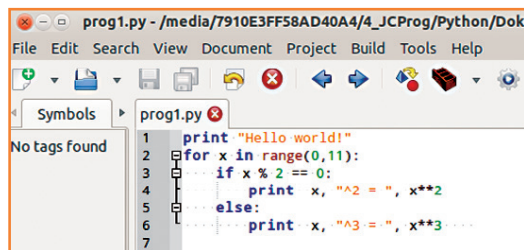


Bild 1.
Der freie Editor Geany.

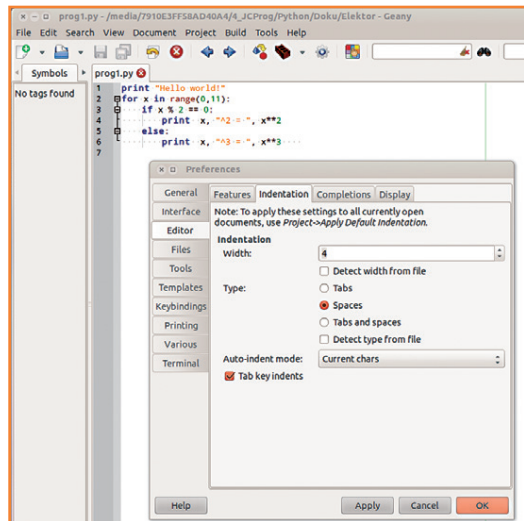


Bild 2.
Die „pythonisch“ korrekte
Einrückung um 4
Leerzeichen.

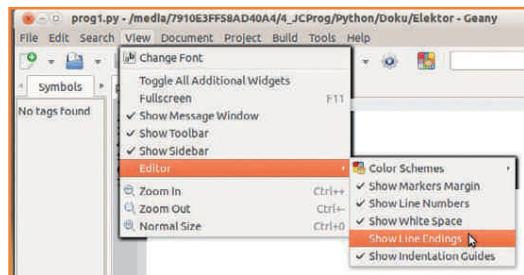


Bild 3.
Weitere Einstellungen in
Geany.

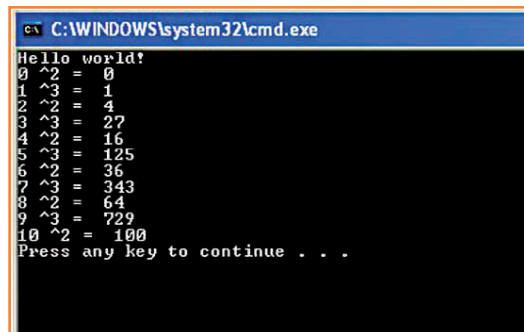


Bild 4.
Resultat des Testprogramms
unter Windows.

Daraufhin kopiert das Python-Script die erforderlichen Dateien in ein dem Interpreter bekanntes Unterverzeichnis. Dieses ist vom Betriebssystem abhängig („/usr/lib/python2.7“ für Ubuntu oder „\Python27\Lib“ für Windows XP).

IDE oder Nicht-IDE?

Zu Anfang habe ich eine IDE wie die von VB sehr vermisst. Doch konnte ich mich ganz gut an das Fehlen eines direkten Äquivalents gewöhnen. Für einfache Programme genügt ein guter Editor. Ein Beispiel ist der von mir bevorzugte „Geany“, der als Freeware für Windows und Linux verfügbar ist. Er bietet automatisches Syntax-Highlighting, wenn man die Datei mit der Extension „.py“ sichert. Schön ist auch sein Code-Folding und dass man ein Script direkt aus dem Editor heraus starten kann.

Selbst bei OS X muss man nicht lange suchen, denn die Apple-eigene Programmierumgebung Xcode kapituliert auch nicht vor Python. Man muss lediglich das 1,65 GB große Paket kostenlos aus dem Apple-App-Store laden. Der Xcode-Editor stellt .py-Dateien auf Doppelklick hin korrekt dar.

Ein erstes Programm

Programme können mit beliebigen Editoren geschrieben werden. **Bild 1** zeigt, wie so etwas in Geany aussieht. In Python ist Einrückung des Quelltextes sehr wichtig: Es können entweder Tabulatoren oder Leerzeichen verwendet werden - Mischen ist nicht erlaubt. „Pythonisch korrekt“ sind vier Leerzeichen. **Bild 2** zeigt, wie man das bei Geany einstellt. Für das praktische Programmieren empfehlen sich noch die Anzeige von Markierungsrand, Zeilennummern und Einrückhinweisen wie in **Bild 3**.

Wenn man das Minimal-Testprogramm von Bild 1 startet, dann sieht das Resultat unter Windows so aus wie in **Bild 4**. Falls man das Programm nicht direkt aus dem Editor starten will oder kann, ist das auch über die Sequenz „python test.py“ per Kommandozeile möglich. Bei GUI-Scripts funktioniert eventuell auch ein Doppelklick auf die .py-Datei, doch bei Textprogrammen wie test.py schließt sich nach der Ausgabe sofort wieder das Fenster, so dass man nichts erkennen kann. Wenn das Programm unter Linux benutzt werden soll, gehört es zum guten Ton, vor dem eigentlichen Quellcode noch die beiden Zeilen:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

einzufügen. Die erste Zeile informiert das Betriebssystem über den zu verwendenden Interpreter und die zweite über den Zeichensatz. BASIC-Programmierer müssen sich außerdem an ein paar Python-typische Eigenheiten gewöhnen, die im **Kasten Python-Spezialitäten** zusammengefasst sind.

Die Python-Shell

Startet man Python von der Kommandozeile (DOS-Box bei Windows bzw. Terminal bei Linux oder OS X), dann bekommt man ein Prompt „>>>“. Nun kann man Befehle interaktiv eingeben und auf diese Weise diejenigen ausprobieren, die man noch nicht kennt:

```
>>> s = "hello"
>>> s.upper()
'HELLO'
```

In der Shell kann man auch Bibliotheksmodule importieren:

```
import time
```

Hilfe zu diesen Modulen erhält man per „help(<Modul>)“, bzw. eine Liste der Objekte und Methoden mit „dir(<Modul>)“.

Anfänger-Fehler

Wer an andere Programmiersprachen gewöhnt ist, dem passieren leicht typische Fehler. Ein schnell fabrizierter Fehler betrifft die Einrückung, die ja bei Python besonderes Gewicht hat. Wenn der Interpreter „Unexpected indent“ meldet, klemmt es genau daran. Eventuell ist dann ein Leerzeichen zu viel bzw. zu wenig vorhanden, oder aber Tabs und Leerzeichen wurden gemischt. Nochmals: Python-Standard sind vier Leerzeichen. Man muss genau auf Integer- und Float-Zahlen und -Variablen achten, denn $3/5 = 0$ und nur $3.0/5.0 = 0.6$!

Unter Python-Formalia fällt, dass für jedes Unterprogramm bzw. jede Methode eine Klammer zum Abschluss benötigt wird. Zum Schließen der seriellen Schnittstelle wäre also „s.close“ falsch und nur „s.close()“ wird akzeptiert.

Externe Hardware

Das Ansprechen externer Hardware ist für Elektroniker ein entscheidender Aspekt einer Programmiersprache. Für Python gibt es fertige Module zum Ansteuern von PC-Schnittstellen: *pyUSB*,

Listing 1: ScanSerial.py

```
import serial
def scan_serial():
    """ Scans for available serial ports """
    portnames = []
    # Windows
    for i in range(256):
        try:
            name = "COM"+str(i)
            s = serial.Serial(name)
            s.close()
            portnames.append(name)
        except:
            pass
    # Linux
    for i in range(256):
        try:
            name = "/dev/ttyS"+str(i)
            s = serial.Serial(name)
            s.close()
            portnames.append(name)
        except:
            pass
    # Linux USB
    for i in range(256):
        try:
            name = "/dev/ttyUSB"+str(i)
            s = serial.Serial(name)
            s.close()
            portnames.append(name)
        except:
            pass
    return portnames
#-----
# main
portnames = scan_serial()
for p in portnames:
    print p
```

Listing 2: ReadSerial.py

```
"""Read and print serial data from COM1 (9600baud)"""
import serial
# init serial port COM1 / ttyS0
sCOM1 = serial.Serial(0)
sCOM1.setBaudrate(9600)
if sCOM1.isOpen()==False:
    sCOM1.open()
# read lines of data until user presses <Ctrl-C>
while(1):
    line = sCOM1.readline()
    print line
sCOM1.close()
```

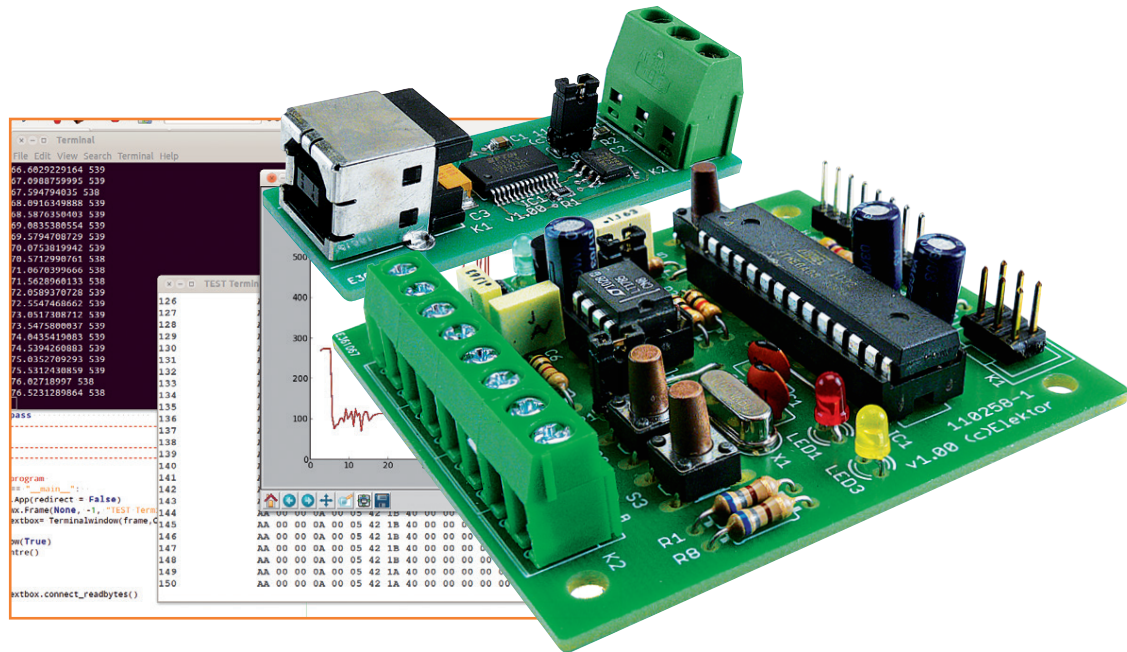


Bild 5.
Python eignet sich auch sehr gut, um ElektorBus-Nachrichten mit Messwerten zu verarbeiten.

pySerial, pyParallel und pyI2C.

Als einfaches Beispiel zunächst die serielle Schnittstelle: Zur Kommunikation wird dem Port eine Instanz des Serial-Objekts im Modul „serial.py“ zugeordnet. Serielle Ports können über ihre Nummer oder ihren Namen („COM1“ bei Windows oder „/dev/ttyS0“ bei Linux) angesprochen werden.

Die einfachste Methode nach vorhandenen Ports zu suchen besteht darin, sie versuchsweise zu instanziiieren (siehe **Listing 1**). Nur wenn das gelingt, existiert der jeweilige Port auch. So wer-

den auch virtuelle USB-Ports erfasst.

Listing 2 zeigt, dass es recht einfach ist, Daten über die serielle Schnittstelle zu empfangen, die z.B. von einem Mikrocontroller stammen können. Nach der Zuordnung des Objekts sCOM1 zu Port COM1 wird die Baudrate angegeben (Parity- und Stop-Bits haben hier Default-Werte) und der Port geöffnet. Beim Experimentieren passiert es gerne, dass sich das Programm mit geöffnetem Port verabschiedet. Beim nächsten Programmstart lässt sich dann der Port nicht öffnen. Besser öffnet man einen Port also nur dann, wenn er nicht schon geöffnet war.

Python-Spezialitäten

Variablen werden implizit durch Zuweisung deklariert: **x = 5.0**

Es gibt keine For-Next-Schleife wie bei anderen Programmiersprachen. Allerdings kann eine Iteration über Teilobjekte eines Objekts durchgeführt werden, z.B. über die Zeichen eines Strings oder die Zeilen einer Datei. Als Ersatz für das klassische For-Next kann man „for i in range“ verwenden.

```
for i in range(0,5):  
    print i
```

Dies ist eine Iteration über alle Bestandteile des Objekts „range(0,5)“, d.h. über die Liste [0,1,2,3,4]. Achtung: Der Endwert (hier 5) ist nicht mehr Teil der Liste!

Arrays (Listen) können unterschiedliche Objekte enthalten:

```
x = [0, 3.14, „Ham“ , „Eggs“]
```

Funktionen können mehrere Werte als „Tupel“ zurückgeben:

```
(x, y, z) = myfunction(v)
```

Es gibt keine SUBs - nur Funktionen, wie in C. Diese werden mit „def...()“ deklariert und können bei Bedarf mehrere Werte oder Objekte zurückgeben.

Das Programm läuft etwas unschön in einer Endlosschleife, bis es mit <Ctrl-C> abgebrochen wird. Eleganter wäre die Abfrage einer Taste zur Beendigung des Programms. Dies ist zwar möglich, aber nicht ganz so einfach, da sich hier die Betriebssysteme unterscheiden. Die Funktion „raw_input()“ sollte nicht benutzt werden, da sie stur auf die Eingabe von <Enter> wartet. Wenn man ein Programm mit GUI gestalten will, gibt es z.B. mit wxPython einfache Möglichkeiten auf Tastendrücke und andere Events zu reagieren. Mit der kleinen Programmiererweiterung von **Listing 3** kann man schon Daten loggen. Zwecks Übersichtlichkeit wurden Unterprogramme benutzt. Diese werden am Anfang mit „def...()“ definiert und erst am Ende kommt das Hauptprogramm. Zunächst wird in „show_log()“ nachgesehen (try / except) ob schon eine LOG-Datei existiert, in diesem Fall wird ihr Inhalt gelesen und ausgegeben. Schön ist, dass der ganze Text mit „file.read“ gelesen werden kann. Dann wird

Linux Debugging (2-tägiges Seminar)

Das Ziel dieses Kurses ist es, einen grundlegenden Überblick über Debugging / Tracing / Profiling Möglichkeiten unter GNU/Linux zu geben. Hands-on-Übungen geben Ihnen die notwendige praktische Erfahrung, um das richtige Tool für Ihre Debugging-Anforderungen auswählen zu können.

Hands-on-Sessions werden auf Ubuntu durchgeführt. Der gesamte Workshop ist konzipiert, um Sie schnell „up-to-speed“ zu bringen. Erfinden Sie das Rad nicht neu, sondern lernen Sie von einem erfahrenen Trainer und nehmen Sie dieses Wissen mit nach Hause, um diese neu erworbene Fähigkeit bei eigenen Entwicklungen effektiv zu nutzen.

Referent: Robert Berger

- Teilnahmegebühr: 995,00 € (inkl. MwSt.)

Echtzeitbetriebssysteme in Theorie und Praxis (3-tägiges Seminar)

Das modular aufgebautes Seminar ist je nach Interesse und Wissenstand konzipiert, so dass Sie beliebig einsteigen können.

Tag 1 Was ist Echtzeit? Was ist FreeRTOS? Installation von Werkzeugen, tasks, scheduling, queues,...

Tag 2 Interrupts, Nebenläufigkeit (concurrency), gegenseitiger Ausschluss (mutual exclusion), Speicherverwaltung,...

Tag 3 allgemeine Methoden zur Vermeidung von Fehlern, Wie kann man die Fehler, die man nicht vermeiden konnte, finden (mit Schwerpunkt Echtzeitbetriebssysteme)?

Ziel des Kurses ist es, Ihnen die grundlegenden Konzepte eines Echtzeitbetriebssystems und dessen Handhabung zu vermitteln. Was sind zum Beispiel Vor- und Nachteile? Wie kann man diverse Programme auf einem PC erstellen/kompilieren und auf einem eingebetteten System ausführen? Eine Kombination aus Theorie und praktischen Übungen wird es Ihnen ermöglichen, das neu erworbene Wissen bei Eigenentwicklungen einzusetzen.

Referent: Robert Berger

- Teilnahmegebühr: für alle 3 Seminar-Tage: 1.799,00 € (inkl. MwSt.)
für 2 Seminar-Tage: 1.250,00 € (inkl. MwSt.)
für 1 Seminar-Tag: 650,00 € (inkl. MwSt.)



Arduino - Programmierung und Projektentwicklung (Tagesseminar)

Nach einer kurzen Einführung und der Inbetriebnahme des Arduino-Boards erfolgt eine systematische Einführung in verschiedene Themengebiete. Dabei wird neben den erforderlichen theoretischen Grundlagen stets größter Wert auf eine praxisorientierte Ausrichtung gelegt. So werden wichtige Techniken wie AD-Wandlung, Timer oder Interrupts anhand von Praxisprojekten ausführlich erläutert. Den Abschluss des Seminars bildet eine Einführung in die eigenständige Entwicklung von Projekten und Systemen. Der Seminarteilnehmer wird damit in die Lage versetzt, auch komplexe eigene Ideen in praxistaugliche Geräte umzusetzen.

Referent: Dr. Günter Spanner

- Teilnahmegebühr: 349,00 € (inkl. MwSt.)

Listing 3: ReadSerial2.py

```
"""Read and print serial data from COM1 (9600baud)"""
import serial

def show_log():
    """ show result of last logging"""
    try:
        file = open("test.log", "r")
        text = file.read()
        file.close()
        print "CONTENTS OF LAST LOG FILE:"
        print text
        print "END OF LOG FILE"
    except:
        print "NO LOG FILE FOUND"
        pass

def open_port_log(port):
    """ open serial port and LOG file"""
    # init serial port COM1
    sCOM = serial.Serial(port-1)
    sCOM.setBaudrate(9600)
    if sCOM.isOpen()==False:
        sCOM.open()
    file = open("test.log", "w")
    return sCOM, file

def receive(sCOM, file ):
    """ read lines of data until user presses <Ctrl-C>"""
    while(1):
        line = sCOM.readline()
        print line
        file.write(line)

port=1
show_log()
ok = raw_input("NEW LOG (y/n) ?")
if ok== "y":
    s, f=open_port_log(port)
    receive(s, f)

# file and port closing done by interpreter at <Ctrl-C>
```

der serielle Port geöffnet. Jede empfangene Zeile wird angezeigt und in die Datei geschrieben.

Fazit & Ausblick

Wenn Sie nun Geschmack an Python gefunden haben, dann sollten Sie jetzt in der Lage sein, sich den Interpreter und einige Bibliotheken zu installieren und auf Anhieb einige einfache Programme zu schreiben. Alle Listings finden Sie auch unter [1]. Im zweiten Teil dieser Serie wird es dann um für Elektroniker interessante Feinheiten wie Diagramme und Fourier-Synthese sowie die Ausstattung mit einer schönen Benutzerschnittstelle gehen. Und im dritten Teil wird's praktisch. Wir arbeiten mit einem kleinen Mikrocontrollerboard, das über den ElektorBus Messwerte an den PC sendet (siehe **Bild 5**).

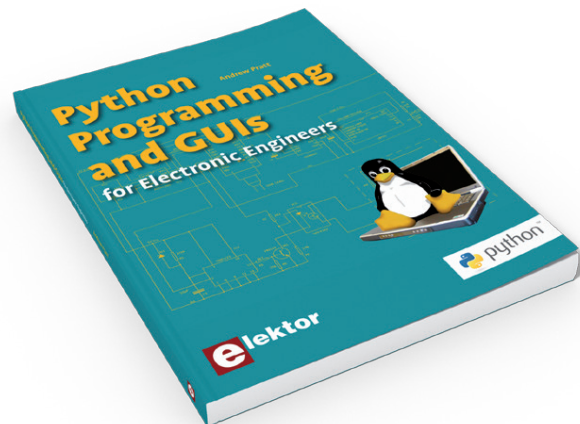
(110483)

Weblinks & Literatur

- [1] Listings etc.:
www.elektor.de/110483
- [2] Homepage des Autors:
<http://staff.itam.lu/feljc/home.html>
- [3] Doku zu Python:
<https://pypi.python.org/pypi/RPi.GPIO>
- [4] Python-Tutorials:
www.awaretek.com/tutorials.html
- [5] Referenz:
Michael Weigend: Python gepackt
- [6] Einführung:
J.M. Hughes: Real World Instrumentation with Python
- [7] Python für Elektroniker:
Andrew Pratt: Python Programming and GUIs for Electronic Engineers
www.elektor.de/python-programming

Über den Autor

Jean-Claude Feltes unterrichtet als Diplom-Ingenieur für Elektronik am Lycée Technique des Arts et Métiers in Luxemburg. Diese Berufsschule für Technik und Kunst führt zum Gesellen, Techniker oder BTS-Abschluss. Er beschäftigt sich auch in seiner Freizeit viel mit Elektronik und Programmierung (siehe [2]).



Alle Elektor-Artikel der „70er-Jahre“ auf DVD!

NEU



**Ein Muss
für jeden
Elektor-Leser!**

ISBN 978-3-89576-263-5
€ 69,00 • CHF 85,60

elektor

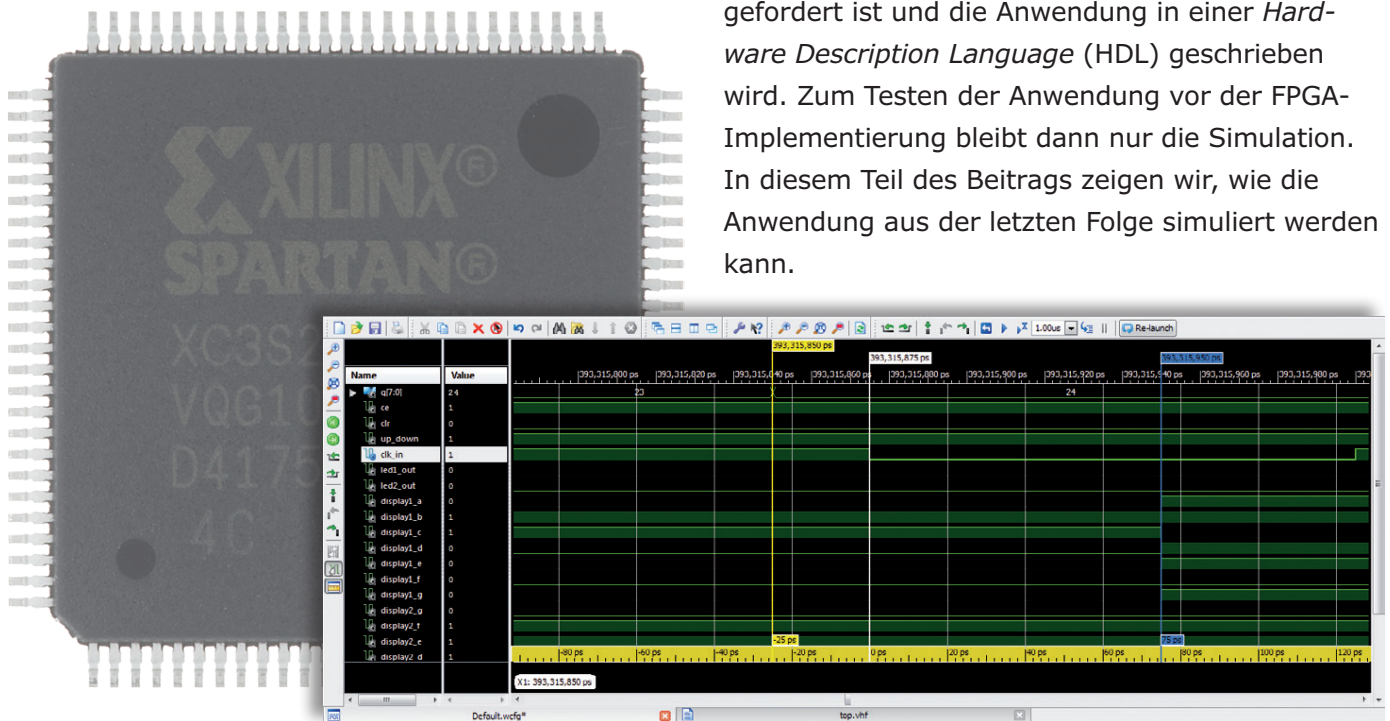
Jetzt unter
www.elektor.de/70-79
bestellen!

Bauen Sie Ihren Chip! (4)

250.000 Gatter laufen Probe

Von **Clemens Valens**
(Elektor .LABS)

Auf dem Weg zum einsatzfähigen FPGA ist die Simulation eine wichtige Zwischenstation. Solange die Anwendung aus nur wenigen Gattern besteht, bedarf es normalerweise keines Probelaufs. Doch das ändert sich, wenn die volle Leistung gefordert ist und die Anwendung in einer *Hardware Description Language* (HDL) geschrieben wird. Zum Testen der Anwendung vor der FPGA-Implementierung bleibt dann nur die Simulation. In diesem Teil des Beitrags zeigen wir, wie die Anwendung aus der letzten Folge simuliert werden kann.



Wenn nach Implementieren einer FPGA-Anwendung Komplikationen auftreten, muss dies nicht an mangelnden Fähigkeiten des Entwicklers liegen. Dafür können auch die Chip-Architektur oder ein Bug in der Entwicklungs-Software verantwortlich sein. Insbesondere FPGAs mit sehr hohen Taktgeschwindigkeiten neigen bisweilen zu „unerklärlichem“ Verhalten. Die Gefahr steigt, wenn die Signalwege und Signallaufzeiten innerhalb des Chips ein gewisses Maß übersteigen. Der FPGA-Entwickler trägt dem Rechnung und simuliert das Produkt entwicklungsbegleitend, beginnend in einem frühen Stadium. Verhält sich der FPGA wie erwartet? Produziert der FPGA Nebeneffekte, die kritisch sein könnten?

Simulationen werden in unterschiedlichen Stadien der Produktentwicklung durchgeführt. Den Anfang macht die funktionale Simulation, sie soll

den Beweis für die generelle Tauglichkeit des Konzepts erbringen. Allein genügt das natürlich nicht, vor der Produktreife sind noch unzählige Hürden zu nehmen. Jeder Schritt bedeutet, dass weitere Fehler eingeschleust werden können, die eine nachfolgende Simulation ans Licht bringen kann. Wenn nach dem letzten Feinschliff die letzte Simulation erfolgreich bestanden wurde, kann der Entwickler davon ausgehen, dass ihm sein Werk gelungen ist. Natürlich ist dieses viel versprechende Omen noch keine Garantie, auch die ausgefeilteste Simulation bleibt eine Nachahmung der Realität.

Simulator öffnen

Wir haben uns zum Ziel gesetzt, den zweistelligen BCD-Aufwärts-Abwärts-Zähler aus der letzten Folge zu simulieren. Zuerst erstellen wir eine

Kopie jenes Projekts, das wir *part3* genannt hatten (siehe Teil 3 der Artikelreihe, [3]). Unser neues Projekt, um das es jetzt geht, hat folgerichtig den Namen *part4*. Öffnen Sie das neue Projekt in ISE, bevor Sie an dieser Stelle weiter lesen. Den ISE-Simulator finden Sie oben im Tab *Design*, dort wo *View* angegeben ist. In der letzten Folge war die *Implementation View* (Ansicht) aktiviert, jetzt muss die *Simulation View* gestartet werden. Wenn Sie die Funktion aktivieren, indem Sie auf *Simulation* klicken, ändert sich der Inhalt des Fensters *Processes*. Wahrscheinlich haben sie auch bemerkt, dass oben im Fenster *Hierarchy* ein Element hinzugekommen ist, das die Bezeichnung *Behavioral* trägt. Da die Ausgangsbasis ein nicht kompiliertes Projekt war, enthält das Fenster *Processes* nur eine einzige Zeile: *Design Utilities*. In der Hierarchie wählen Sie *top*, so dass im Prozessfenster der Prozess *ISim Simulator* erscheint. Klappen Sie diese Zeile mit einem Klick auf das Pluszeichen aus. Wählen Sie den Prozess *Simulate Behavioral Model* und klicken Sie links auf die Schaltfläche unter der grünen Pfeilspitze. Alternativ können Sie auch mit der rechten Maustaste auf die Zeile klicken und im Kontextmenü *Run* wählen, oder auch auf die Zeile mit der linken Maustaste doppelklicken. ISE macht sich nun ans Werk, und nach kurzer Zeit erscheint (hoffentlich) im Fenster *Console* die Meldung, dass der Prozess „completed successfully“ wurde. Nach erfolgreichem Verlauf öffnet sich neben dem ISE-Hauptfenster ein zusätzliches ISim-Fenster, das

die Signale ähnlich einem Logic-Analyser darstellt (**Bild 1**). Auf der Konsole unten im ISim-Fenster werden Sie darauf hingewiesen, dass Sie im Modus *Lite* arbeiten, weil Sie (noch) keine Volllizenz erworben haben. Außerdem steht dort, dass die Auflösung der Simulation 1 ps (Picosekunde) beträgt. Rechts oben sehen Sie die Information, dass die gesamte Simulation ein Zeitintervall von 1 µs (Mikrosekunde) umfasst.

Schrittweises Simulieren

Klicken Sie auf die Schaltfläche *Restart* (blau mit eckigem Pfeil, *Restart* ist auch über das Menü *Simulation* erreichbar). Klicken Sie in der Signalliste mit der rechten Maustaste auf das Signal *ce*. Dieses Signal ist doppelt vorhanden, der Mausklick ist beliebig. Sie können auch auf die grafische Darstellung dieses Signals klicken. Achten Sie darauf, dass gleichzeitig nur ein Signal ausgewählt ist. In dem nun erscheinenden Menü wählen Sie *Force Constant*, denn es handelt sich um ein quasi konstantes Signal, verglichen beispielsweise mit einem Taktsignal. ISim öffnet daraufhin das Dialogfenster *Force Selected Signal*, in das mehrere Parameter eingetragen werden können. Tragen Sie in das Feld *Force to Value* die Zahl „1“ und in das Feld *Cancel after Time Offset* die Zeit „1 ms“ ein. Die übrigen Parameter können unverändert bleiben. Klicken Sie auf *OK*. Das Signal *ce* wird 1, sobald die Simulation startet, es behält diesen Zustand für 1 ms bei. Wiederholen Sie die Prozedur für das Signal *up_down*

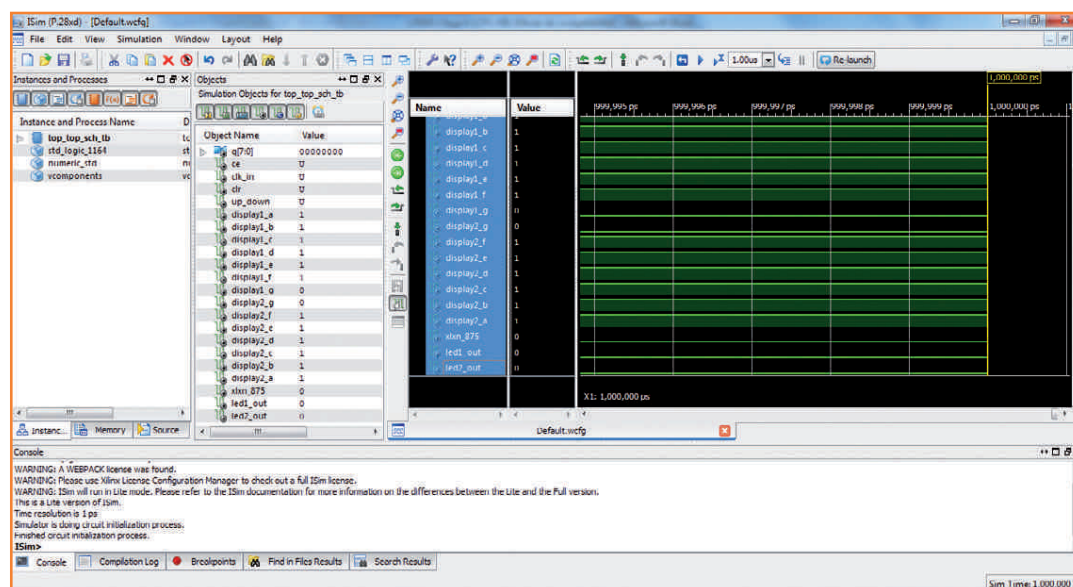
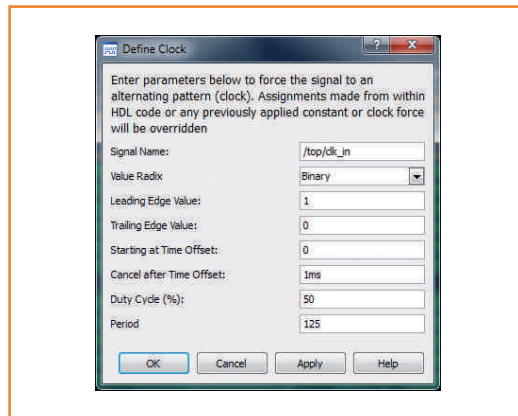


Bild 1.
ISim glänzt hier mit seiner Funktionalität. Noch ruht das Geschehen, denn die Definitionen der auslösenden Events fehlen.

Bild 2.
Zahlenwerte ohne
Einheiten sind im Fenster
Define Clock Zeiten in
Picosekunden.



und anschließend mit dem *Force to Value*-Wert 0 für das Signal `clr`.

Für das Signal `clk_in` führen Sie eine identische Prozedur durch, doch statt *Force Constant* wählen Sie hier *Force Clock*. Das Dialogfenster *Define Clock*, das jetzt erscheint, gleicht dem Fenster *Force Selected Signal*, doch es gibt Unterschiede. Tragen Sie die Werte aus **Bild 2** ein und klicken Sie auf *OK*. Diese Werte haben zur Folge, dass das Taktsignal mit einer Vorderflanke zur Zeit 0 startet, während 1 ms aktiv bleibt, den Duty-Cycle 50 % hat und die Periode „125“ beträgt. Gemeint sind 125 ps, denn die vorgegebene Zeiteinheit ist die Picosekunde. Diese Zeiteinheit gilt, solange keine andere Einheit angegeben wird. Die Taktfrequenz auf dem FPGA-Board beträgt 8 MHz, die Periode ist hier 125 ns. Deshalb tragen Sie nicht „125“, sondern „125 ns“ ein.

Die Eingangssignale sind nun definiert, Sie können die Simulation starten. Nach einem Klick auf die Schaltfläche *Run All* (oder über das Menü *Simulation*) läuft die Simulation so lange, bis neue Events nicht mehr auftreten oder bis ein Breakpoint die Simulation anhält. Da kein Breakpoint definiert ist, endet die Simulation erst, wenn die auslösenden Signale wegfallen, also nach 1 ms. Ein Taktsignal erzeugt natürlich fortlaufend neue Events. Mit einem Klick auf die Schaltfläche *Break* (Pausensymbol) können Sie die Simulation jederzeit unterbrechen. Bitte beachten Sie: Nach einem Abbruch öffnet ISim eine Datei mit einem VHDL-Quelltext (meistens `top.vhd`), die *waveform*-Ansicht befindet sich dann dahinter, was am Tab *wcfg* erkennbar ist. Klicken Sie auf den Tab, um ihn in den Vordergrund zurückzuholen.

Sie können die Simulation auch schrittweise ausführen, indem Sie die Schaltfläche *Run* neben der Schaltfläche *Run All* anklicken. Vorher müssen

Sie noch die Simulationsdauer in das daneben befindliche Feld eingeben, beispielsweise „10 ms“. Jeder Klick auf *Run* löst dann ein Simulationsintervall von 10 ms aus.

Wenn die Simulation stoppt, weil sie beendet ist oder abgebrochen wurde, können Sie die Signale betrachten (**Bild 3**). Sie können die Signale größer oder kleiner zoomen und durch die Signale scrollen. Die Reihenfolge und Gruppierung der Signale ist durch Ziehen mit der Maus änderbar. Falls dies ausnahmsweise nicht möglich sein sollte, kann ein Blick auf die Eingangssignale weiter helfen. Falls sie nicht grün, sondern orange dargestellt sind, kann dies daran liegen, dass die Signaldefinitionen fehlen.

Nachdem der passende Zoomfaktor eingestellt ist, können Sie das Taktsignal oder andere veränderliche Signale beobachten. Signale auf Bussen, also Multi-Bit-Signale wie `q[7:0]` erscheinen mit ihren aktuellen Werten in binärem Format. Das Darstellungsformat können Sie über die Option *Radix* in dem Popup-Menü ändern, in dem auch *Force Constant* steht. Links neben der grafischen Signaldarstellung werden die Signalwerte an der Position des gelben Cursors angezeigt.

Auf dem Prüfstand

Das beschriebene schrittweise Simulieren einer FPGA-Anwendung ist zwar anschaulich, aber im Verlauf der Entwicklungsarbeiten wenig praktisch. Lästig ist auf die Dauer, dass die Signale jedes Mal über Dialogfenster neu definiert werden müssen. Falls Sie flinke Finger auf der PC-Tastatur haben, können Sie stattdessen textorientierte Kommandos in die Konsole eingeben. Allerdings müssen Sie sich zuvor die Kommandos und ihre Bedeutungen zu eigen machen. Das Dokument *plugin_ism.pdf*, das auf der Website von Xilinx verfügbar ist, gibt darüber Auskunft. Eine effizientere Vorgehensweise, die auch Xilinx empfiehlt, ist das Automatisieren der Simulation mit einer so genannten *test bench* (sinngemäß: Prüfstand). Eine solche *test bench* (oder auch *test fixture*) ist eine Datei, die zum ISE-Projekt hinzugefügt werden muss, während die *Simulation View* aktiv ist. Das geschieht wie gewohnt über *New Source*, aufrufbar durch einen rechten Mausklick in das *Hierarchy*-Fenster. In dem *New Source Wizard*, der jetzt erscheint, wählen Sie *VHDL Test Bench*. Vergeben Sie einen Namen, zum Beispiel *test_bench*, vergewissern Sie sich, dass *Add to project* angehakt ist, und klicken Sie auf *Next*. Jetzt müssen Sie die Test-Bench einer Source-Datei zuord-

Test-Bench in Verilog

Ähnlich den zahlreichen Computer-Programmiersprachen existieren mehrere Sprachen für FPGA-Implementierungen. Im Text haben wir uns mit VHDL beschäftigt, denn ISE scheint hauptsächlich für diese Sprache konzipiert zu sein. Damit Verilog, eine andere wichtige Sprache nicht zu kurz kommt, bauen wir die Test-Bench hier noch einmal in Verilog.

Das Taktsignal wird in Verilog mit dem Kommando `always` generiert (Achtung: Anders als in VHDL werden in Verilog Groß- und Kleinschreibung unterschieden!). Dank des Invert-Operators „!“ umfasst dies nur eine Zeile. Die Frequenz wird mit dem Delay-Symbol „#“ bestimmt, hier sind es 62,5 ns. Zeiteinheit ist in Verilog die Nanosekunde, die Auflösung beträgt trotzdem 1 ps.

```
always
  #62.5 CLK_IN = !CLK_IN;
```

Der logische Zustand des Signals `clk_in` wird in Abständen von 62,5 ns invertiert. In Anlehnung an VHDL ist auch folgende Schreibweise möglich:

```
always
begin
  CLK_IN = 1;
  #62.5;
```

```
CLK_IN = 0;
#62.5;
end
```

Darin muss `clk_in` nicht initialisiert werden. Die Initialisierung der auslösenden Events wird in einem so genannten `initial`-Block vorgenommen, er wird ein Mal zu Beginn der Simulation ausgeführt. Die Struktur ist der VHDL-Struktur ähnlich, auch hier steht am Anfang eine Pause von 100 ns:

```
initial
begin
  #100;
  CLK_IN <= 0;
  CE <= 1;
  CLR <= 0;
  UP_DOWN <= 1;
end
```

Anstelle von „<=“ kann auch „=“ geschrieben werden. Vollständigkeitshalber ist `clk_in` initialisiert, was zwar nicht notwendig ist, aber auch nicht schadet.

Die Blöcke `always` und `initial` ersetzen gemeinsam den Block `ifdef auto_init ... endif`, unten in der von ISE generierten Verilog-Test-Bench.

Ist alles klar? Dann bitte simulieren!

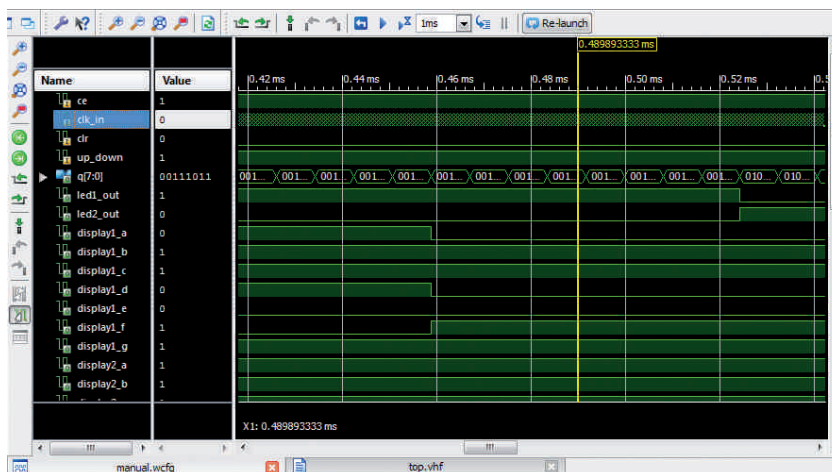


Bild 3.
Das Signalfenster nach einer Simulation. In diesem weiten Zoom wird das Signal `clk_in` lediglich als Schraffur dargestellt. Der Wert des 8-bit-Bussignals kann in unterschiedlichen Formaten angezeigt werden.

nen. Da die gesamte Implementierung getestet wird, binden Sie die Datei an *top*. Klicken Sie auf *Next*, so dass die übliche Zusammenfassung erscheint. Aus der Zusammenfassung geht hervor, dass der Dateiname die Erweiterung *.vhd* erhalten hat. Dieses Indiz deutet auf eine VHDL-Datei hin. Klicken Sie auf *Finish*, um den Vorgang abzuschließen.

Wichtig ist hier, dass Sie *test_bench* nicht in der *Implementation View*, sondern in der *Simulation View* hinzufügen müssen. Eine eventuell falsche Zuordnung können Sie korrigieren, indem sie in den Dateieigenschaften (*Source Properties*) die *View Association* auf *Simulation* setzen.

Das Hinzufügen der Test-Bench hat weiter reichende Auswirkungen als vielleicht erwartet. Im Hierarchie-Fenster erscheint zwischen dem Chip und *top* eine weitere Zeile, die den zunächst kryptischen Text *top_top_sch_tb* enthält. Der Eintrag *top* ist zu *UUT - top* geworden, wobei *UUT* ein Kürzel für *Unit Under Test* ist. Aus der Hierar-

korrekten Zeiten. Zum Testen unseres Aufwärts-Abwärts-Zählers müssen wir in VHDL oder Verilog die Vorgänge reproduzieren, die wir oben schrittweise durchgeführt haben. Außer einem Taktsignal benötigen wir einige quasi-konstante Signale. Mit den konstanten Signalen wollen wir beginnen.

Der Rahmen unserer VHDL-Test-Bench hat folgende Gestalt, die reservierten VHDL-Kommandos sind in Großbuchstaben geschrieben (was nicht zwingend notwendig ist!):

```
tb : PROCESS
BEGIN
    WAIT;
END PROCESS;
```

Dies ist gewissermaßen die Test-Bench-Hauptroutine, oder besser, der Prozess. Unser Prozess mit dem Namen *tb* ist lediglich eine Warteschleife. Ein Prozess sieht ähnlich einer Subroutine in einem

Prozesse werden parallel ausgeführt: Ein ideales Multitasking

chie geht folglich klar hervor, was und durch wen getestet wird. Die Datei *test_bench.vhd* wurde inzwischen im Editor geöffnet, wie Sie sehen, ist der Inhalt unverändert.

Wir wollen noch eine zweite Test-Bench hinzufügen, diesmal jedoch nicht in VHDL, sondern in Verilog. Das geschieht in gleicher Weise wie beschrieben, doch diesmal wählen wir *Verilog Test Fixture* als Source-Typ. Nach Hinzufügen der zweiten Test-Bench ist im Hierarchie-Fenster offensichtlich eine zweite Hierarchie dazugekommen. Daraus lässt sich schließen, dass im gleichen Projekt mehrere Test-Benchs einsetzbar sind (das ist uns allerdings bisher noch nicht gelungen!). ISE ist vermutlich so konzipiert, dass die VHDL-Test-Bench den Vorzug hat. Der Name der Verilog-Datei trägt die Erweiterung *.v*, und das Icon der Verilog-Datei unterscheidet sich vom Icon der VHDL-Datei. Da Verilog- und VHDL-Dateien unterschiedliche Namensweiterungen haben, kann der gleiche Dateiname an beide Test-Benchs vergeben werden. Ob dies wirklich sinnvoll ist, sei dahingestellt.

Und nun: VHDL

Die Test-Bench enthält lediglich ein Gerüst, was noch fehlt, ist das Aktivieren der Signale zu den

Programm aus, doch im Gegensatz zu Subroutinen werden Prozesse nicht sequentiell, sondern parallel, also gleichzeitig ausgeführt. Außerdem werden Prozesse fortlaufend wiederholt, solange sie nicht unterbrochen werden, wie zum Beispiel hier durch die unendliche Warteschleife *WAIT*. In diesen Prozess fügen wir die Signale *ce*, *clr* und *up_down* ein und ordnen ihnen logische Zustände zu:

```
tb : PROCESS
BEGIN
    WAIT FOR 100 NS;
    ce <= ,1';
    clr <= ,0';
    up_down <= ,1';
    WAIT;
END PROCESS;
```

Bevor der Prozess in die Warteschleife mündet, werden die Signale *ce* und *up_down* auf 1 gesetzt, *clr* wird 0. Diese Kommandos werden nicht nacheinander, sondern gleichzeitig ausgeführt, die Reihenfolge im Prozess ist beliebig. Das Kommando *WAIT FOR 100 NS* hält den Prozess für 100 ns an, um das Ende des Reset-Signals abzuwarten. In der Hilfe zu ISim wird diese Maß-

nahme empfohlen.

Das Erzeugen des Taktsignals ist etwas komplexer, es ist ein eigener Prozess nötig, der gleichzeitig mit dem Prozess tb ausgeführt wird:

```
clock : PROCESS
BEGIN
  clk_in <= ,1';
  WAIT FOR 62500 PS;
  clk_in <= ,0';
  WAIT FOR 62500 PS;
END PROCESS;
```

Dieser Prozess endet nicht mit der unendlichen Warteschleife WAIT, er kann nicht unterbrochen werden und wird deshalb ständig wiederholt. In diesem Prozess wird zuerst clk_in auf 1 gesetzt, gefolgt von einer 62,5 ns langen Pause, clk_in wird auf 0 gesetzt, es folgt eine weitere 62,5 ns lange Pause, und schließlich beginnt der Prozess von vorn. Das Ergebnis ist ein Signal mit der Frequenz 8 MHz und dem Duty-Cycle 50 %. Da der Duty-Cycle 50 % beträgt, kann die Pause als Konstante geschrieben werden, wir hängen sie an die Liste SIGNAL an:

```
CONSTANT clock_half_period : TIME :=
62500 PS;
```

Die Konstante clock_half_period ist vom Typ TIME, wir können sie zusammen mit dem Kommando WAIT verwenden. Jetzt sieht der Prozess wie folgt aus:

```
clock : PROCESS
BEGIN
  clk_in <= ,1';
  WAIT FOR clock_half_period;
  clk_in <= ,0';
```

```
WAIT FOR clock_half_period;
END PROCESS;
```

Die Test-Bench ist nun vollständig, die Simulation kann gestartet werden. Wählen Sie in der Hierarchie die Test-Bench, damit der ISim-Prozess sichtbar ist. Wenn Sie die Zeile ausklappen, stehen zwei Optionen zur Wahl: *Behavioral Check Syntax* und *Simulate Behavioral Model*. Führen Sie die erste Option aus, um sicher zu gehen, dass Ihnen keine Tippfehler unterlaufen sind. Fahren Sie fort, wenn hier ein grüner Haken erscheint.

Öffnen Sie mit einem rechten Mausklick auf *Simulate Behavioral Model* die Simulationseigenschaften und haken Sie die Option *Run for Specified Time* an. Damit haben Sie festgelegt, dass die

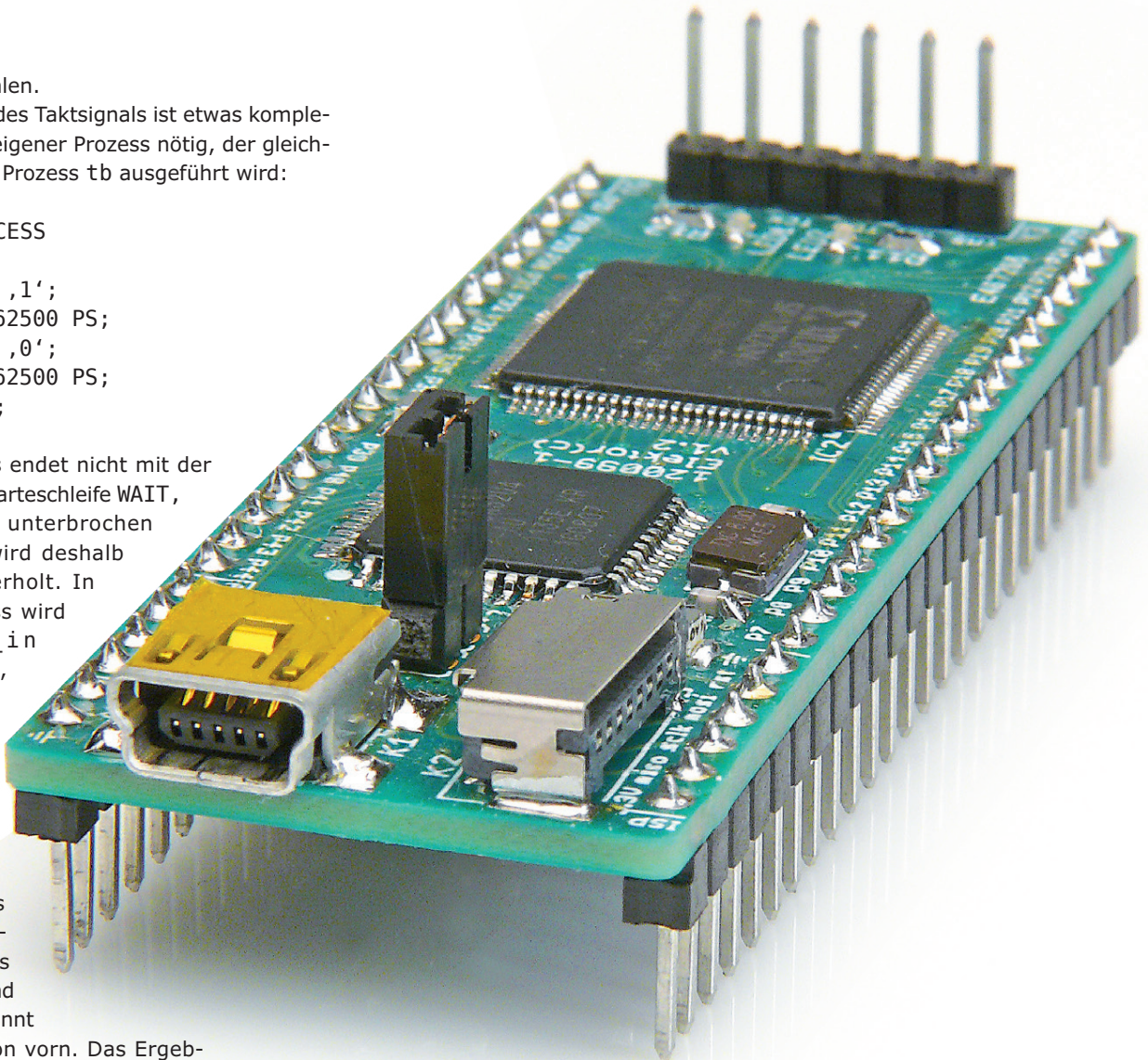
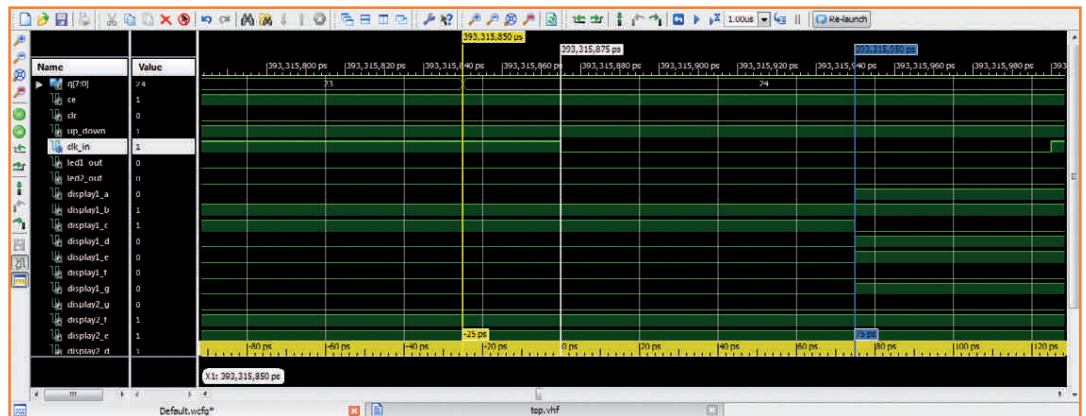


Bild 4.
Zeitintervalle sind mit Markern messbar. In dieser Simulation hat die Taktperiode die Länge 250 ps. Nach dem Anklicken ist der weiße Marker der Bezugspunkt. ISim berechnet die zeitlichen Abstände zu den anderen Markern. Die Zeiteinheit lässt sich nach Rechtsklick auf die Zeitachse einstellen.



Simulation nach dem Start bis zum Ablauf der vorgegebenen Zeit läuft, oder so lange, bis Sie auf *Break* klicken. Schließen Sie das Eigenschaftendialogfenster und wählen Sie *Simulate Behavioral Model*. Das ISim-Fenster öffnet sich, doch es geschieht zunächst nichts. Wenn Sie auf die Schaltfläche *Run All* oder auch auf *Run (for-the-time-specified-on-the-toolbar)* klicken, beginnt die Simulation. Lassen Sie die Simulation so lange laufen, bis in der Zeitleiste die gewünschte Dauer erscheint. Klicken Sie auf *Break*, um die Simulation zu stoppen. Jetzt können Sie wie bei der schrittweisen Simulation die Signale betrachten. Falls die Zeitachse auf einen sehr kleinen Maßstab eingestellt ist, scheinen die Signale statisch zu sein. Mit einem Klick auf die Schaltfläche *Zoom to Full View* holen Sie die Simulation vollständig ins Bild, so dass auch langsame Signale sichtbar sind. Anschließend können Sie die Signalteile her-

vorholen, die für Sie von Interesse sind (**Bild 4**). Die Reihenfolge der Signale in der grafischen Darstellung entspricht der Reihenfolge in der Liste **SIGNAL**. Organisieren Sie diese Liste so, dass die Darstellung optimal ist (achten Sie auf die Kommas!). Vergessen Sie nicht, das ISim-Fenster zu schließen, immer wenn Sie eine modifizierte oder neue Test-Bench verwenden. ISE gibt eine Fehlermeldung aus, falls das ISim-Fenster geöffnet bleibt.

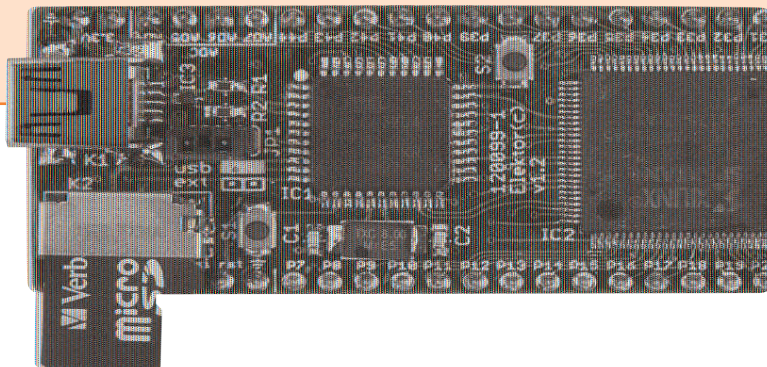
Das war's!

Wir sind ans Ende unserer Einführung in die Simulation der FPGA-Implementierungen gekommen. Sie sollten nun in der Lage sein, einfache Simulationen selbstständig durchzuführen. Weil viele wichtige Details nicht einmal gestreift werden konnten, empfehlen wir Ihnen, die Fachliteratur ebenso wie einschlägige Webseiten zu Rate zu ziehen. Eine Implementierung zu simulieren, ist zunächst vergleichsweise einfach. Die effiziente, aussagekräftige Simulation setzt jedoch solide Kenntnisse über programmierbare Logikbausteine, Hardware-Programmiersprachen und zugehörige Entwicklungs-Tools voraus.

(130065)gd

Das FPGA-Experimentierboard, vollständig aufgebaut und getestet, ist über Elektor für 59,95 € plus Versandkosten erhältlich.

Auf www.elektor.de/120099 steht Näheres!



Weblinks

- [1] Teil 1: www.elektor.de/120099
- [2] Teil 2: www.elektor.de/120630
- [3] Teil 3: www.elektor.de/120743
- [4] Teil 4: www.elektor.de/130065



powered by Eurocircuits

Platinen – Prototypen – Multilayer – Kleinserien

- Höchste Präzision und Industrie-Qualität zum günstigen Preis
- Kein Mindestbestellwert
- Keine Film- oder Einrichtungskosten
- Keine versteckten Kosten
- Online-Preisrechner
- Versand bereits ab 2 Werktagen möglich
- Fünf individuelle, leistungsstarke Service-Optionen stehen zur Auswahl

➔ PCB proto

Ideal für Privatleute, die schnell und günstig maximal 2 Leiterplatten nach vordefinierten Spezifikationen benötigen.

➔ STANDARD pool

Diese Option ist für Firmen konzipiert, die ihre Kleinserie nach den am häufigsten verwendeten Spezifikationen produzieren lassen wollen.

➔ TECH pool

Wenn Ihre Entwicklung sehr anspruchsvolle Spezifikationen erfordert, ist 100-µm-Technologie die beste Wahl.

➔ IMS pool

Bei dieser Option werden Aluminiumkern-Leiterplatten verwendet, um eine hohe Wärmeabfuhr zu gewährleisten.

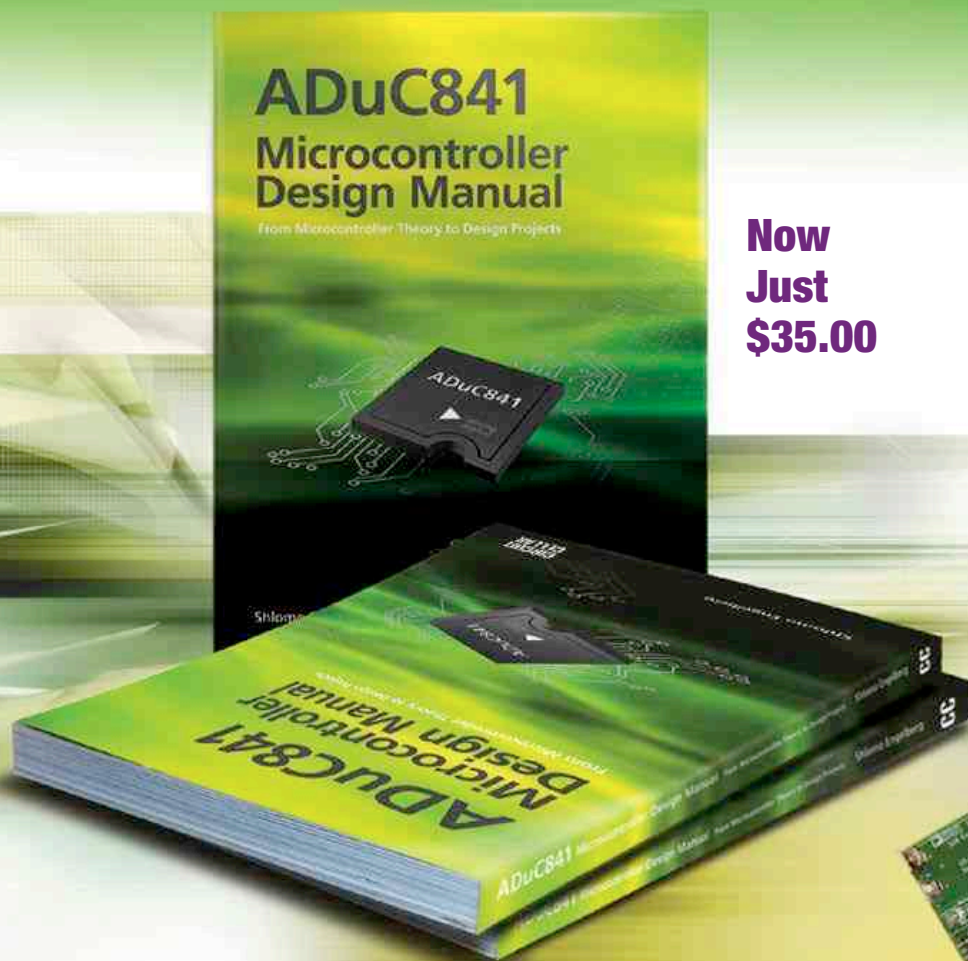
➔ On demand

Wählen Sie selbst aus, nach welchen Spezifikationen und mit welchen Materialien Ihre Platinen angefertigt werden sollen!

Wählen Sie den für Ihre Ansprüche passenden Service und bestellen Sie jetzt Ihre Platinen unter www.elektorpcbservice.de!

ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects

If you've ever wanted to design and program with the ADuC841 microcontroller, or other microcontrollers in the 8051 family, this is the book for you. With introductory and advanced labs, you'll soon master the many ways to use a microcontroller. Perfect for academics!



**Now
Just
\$35.00**

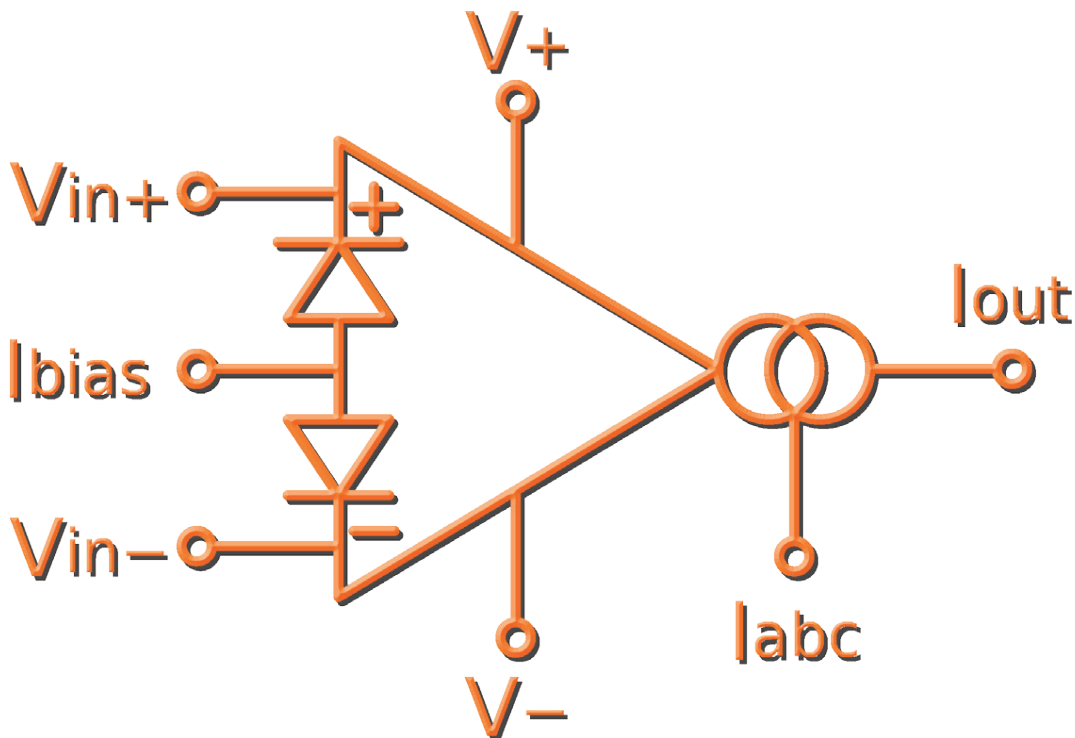
Buy it today!

www.cc-webshop.com

OTA-basierter Dreieck-Generator

Mehr tun mit einem Transkonduktanz-Verstärker

Viele Anwendungen von OTAs wurden schon beschrieben, eine Basisschaltung zur Erzeugung eines nicht-sinusoidalen Signals dürfte aber schwer zu finden sein. Dieser Beitrag beschreibt einen kombinierten Dreieck/Rechteckgenerator, der auf zwei OTAs beruht.



Ein *Operational Transconductance Amplifier* (OTA), im Deutschen auch als Transkonduktanz-Operationsverstärker bekannt, ist ein wunderbares elektronisches Bauteil. Es verwandelt eine (Differenz-)Spannung an seinen Eingängen in einen proportionalen Strom an seinem Ausgang. Seine Parameter können leicht durch Einstellen des Biasstroms I_{abc} geändert werden, ohne dass andere Bauteile angepasst werden müssten. In der Literatur [1], [2] und [3] und in anderen Büchern und Artikeln wurden Unmengen von OTA-Anwendungen beschrieben, wir fügen jetzt dieser Sammlung mit einer Generatorschaltung ein weiteres Kapitel hinzu.

Wie es funktioniert

Die Schaltung des Generators in **Bild 1** zeigt zwei OTAs. OTA1 mit seiner Kapazität C bildet einen Integrator. Die Kapazität C wird von OTA1 mit einem Konstantstrom I_{02} geladen. V_C ist linear abhängig von der Zeit t . OTA2 mit den Widerständen $R1$ und $R2$ arbeitet als Schmitt-Trigger. Der Ausgangsstrom I_{02} ist für die Triggerspannung V_{R2} über dem Widerstand $R2$ verantwortlich. Wenn die Spannung V_C den Schwellwert erreicht, kippt der Ausgang von OTA2 in den entgegengesetzten Zustand, wodurch der Ausgangsstrom seine Richtung zu $-I_{02}$ und die Ausgangsspannung V_2 ihre Polarität ändert. Durch die Gegenkopp-

Von **Libor Gajdošík**
(Tschechien)

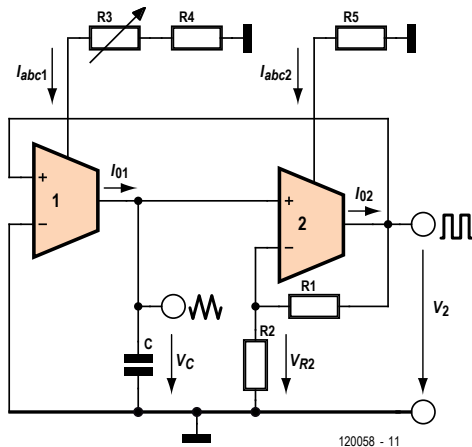
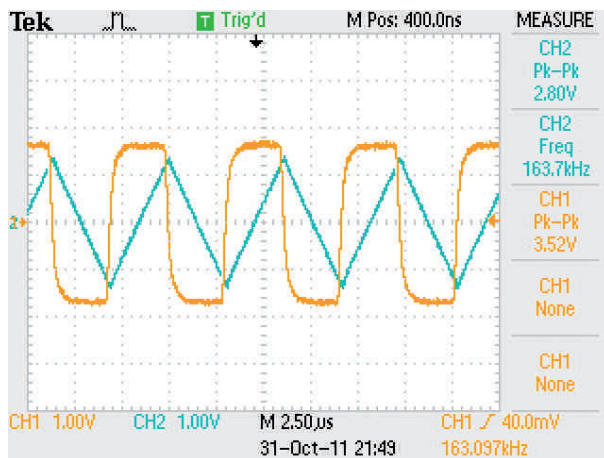


Bild 1.
Basisselemente zum Aufbau
des Generators.

lung wird die Ausgangsspannung U_2 dem Eingang des Integrators zugeführt. Dessen Ausgang ändert seinen Zustand zum entgegengesetzten $-I_{01}$ (durch die Spannung V_2) und die Kapazität sinkt, bis der neue Wert V_{R2} erreicht ist. Dieser Vorgang wiederholt sich ad infinitum. Die Funktionsweise des Generators basiert auf der Tatsache, dass unter den gegebenen Betriebsbedingungen der Ausgangsstrom des OTAs positive und negative Sättigungswerte erreicht, wenn die Eingangsspannung den linearen Spannungsbe-

Bild 2.
Rechteckige und dreieckige
Wellenformen, vom OTA-
Generator produziert.



reich des OTAs verlässt. Dieser Bereich ist in der Regel klein und liegt bei 20...50 mV. Ist die Triggerspannung V_{R2} viel höher als die Eingangsspannung für den linearen Bereich, kann diese Eingangsspannung ignoriert werden. Der OTA-Eingang besteht aus einem Differenzverstärker, so dass beide Eingangsströme als gering und damit als gleich angesehen werden können. Innerhalb einer Periode T wird die Kapazität von $t=0$ und $V_C=0$ bis zum Zeitpunkt $t=t_1$ geladen, wobei V_C den Maximalwert V_{Cm} erreicht. In einer Gleichung:

$$V_{Cm} = \frac{1}{C} \int_0^{t_1} I_{01} \times dt = \frac{I_{01}}{C} t_1 \quad (1)$$

Dieser Wert V_{Cm} entspricht der Triggerspannung V_{R2} , so dass man schreiben kann:

$$\frac{I_{01}}{C} t_1 = V_2 \frac{R_2}{R_1 + R_2} \quad (2)$$

Da sich V_2 aufgrund des Stroms I_{02} über den Widerständen R_1 und R_2 abbildet, ist auch dies richtig:

$$V_2 = I_{02} (R_1 + R_2) \quad (3)$$

(3) wird in (2) eingesetzt und die Zeit t_1 kalkuliert:

$$t_1 = \frac{CR_2 I_{02}}{I_{01}} \quad (4)$$

Es wird davon ausgegangen, dass negativer und positiver Ausgangsstrom des OTAs gleich sind. Da die Dreieck-Wellenform eine teilweise lineare Funktion und sowohl horizontal als auch vertikal symmetrisch ist, ist die Gesamtdauer der Schwingungen T gleich $4t_1$. Für die Frequenz f ist die folgende Formel erfüllt:

$$f = \frac{1}{T} = \frac{I_{01}}{4CR_2 I_{02}} \approx \frac{I_{abc1}}{4CR_2 I_{abc2}} \quad (5)$$

Für Sättigung und Bias-Ströme haben wir:

$$I_{01} \approx I_{abc1} \quad \text{und} \quad I_{02} \approx I_{abc2} \quad (6)$$

Da der Spannungsteiler am Ausgang nicht belastet ist, gilt die folgende Beziehung für die Amplitude V_{Cm} des Dreiecksignals:

$$V_{Cm} = V_2 \frac{R_2}{R_1 + R_2} = I_{02} R_2 \approx I_{abc2} R_2 \quad (7)$$

Die vereinfachte Herleitung der Formeln ist zwar nicht 100%-ig richtig, aber sie kann für eine ungefähre Abschätzung der Frequenz und der Spannung verwendet werden.

In der Praxis

Die Messungen wurden mit dem OTA LM13700, $R_1=820\ \Omega$, $R_2=2\ \text{k}\Omega$ und $C=1\ \text{nF}$ durchgeführt. Die Bias-Ströme wurden mit den Widerständen zwischen Bias-Eingang und Masse eingestellt. Für I_{abc1} wurden $R_3=1\ \text{M}\Omega$ und $R_4=10\ \text{k}\Omega$ eingesetzt, für I_{abc2} wurde $R_5=15\ \text{k}\Omega$ durchgehend verwendet, um einen konstanten Wert von V_2 und V_{Cm} für alle Messungen zu erhalten. Die Versorgungsspannung betrug $\pm 10\ \text{V}$.

Der Ausgangsstrom I_{O1} lag zwischen $14\ \mu\text{A}$ und $0,87\ \text{mA}$, die Frequenz zwischen $3,6\ \text{kHz}$ und $160\ \text{kHz}$. Als Ausgangsstrom I_{O2} wurde $0,57\ \text{mA}$ gemessen. **Bild 2** zeigt die gemessenen Wellenformen.

(120058)

Quellen

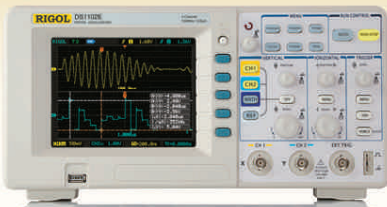
- [1] T. Parveen, Operational transconductance amplifier and analog integrated circuits, New Delhi, India, I.K. International Publishing House, 2009.
- [2] R.L. Geiger und E. Sanchez-Sinencio, „Active filter design using operational transconductance amplifiers: A tutorial“, IEEE Circuits Devices Magazine, vol.1, pp.20 - 32, März 1985.
- [3] W.R. Grise. (Oktober 1998). Operational Transconductance Amplifiers (OTA) for synthesis of voltage-controlled active-filter tuned oscillators. TECHNOLOGY INTERFACE: The Electronic Journal for Engineering Technology. Online verfügbar: <http://technologyinterface.nmsu.edu/fall98/electronics/grise/griseota.html>

Anzeige

BATRONIX

ELEKTOR SONDERPREIS AKTION

EXKLUSIV FÜR DIE LESER DER ELEKTOR UND NUR BIS ZUM 10.06.2013!



DS1102E Oszilloskop
100 Mhz, 1 GSa/s, 1 Mpts,
Bestseller, FFT, USB, LAN,
automatische Messungen,
3 Jahre Garantie

nur 367,- *



DG4000 Generatoren
2 Kanäle, 60 - 160 MHz,
500 MSa/s, voll arbiträr,
16 kpts, Bestseller, 3 Jahre
Garantie

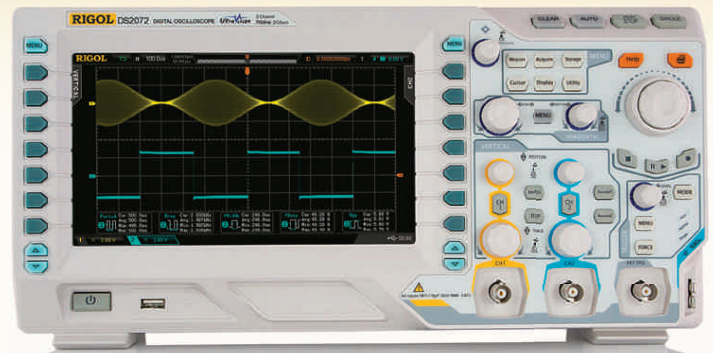
nur 731,- *



**DSA815 Spektrum
Analyser**
9 kHz bis 1.5 GHz,
intuitive Bedienung,
unschlagbares Preis-
Leistungs-Verhältnis

nur 1190,- *

IHR EINSTIEG IN DIE MESSTECHNIK PROFILIGA:



Rigol DS2000 Oszilloskope

- 70 bis 200 MHz, 2 GSa/s, 14 Mpts bis 56 Mpts
- Neue, extrem rauscharme Eingangsstufen
- Automatische Messungen mit Statistik
- Umfangreiche mathematische Funktionen
- Professionelle serielle Trigger
- 3 Jahre Garantie

nur 799,- *

Jetzt die Batronix Sonderpreisaktion für
Elektor-Leser nutzen und modernste
Labortechnik zum Sparpreis bestellen:
www.batronix.com/go/23



* Alle Preise sind bereits inkl. MwSt. und kostenlosem Versand in alle EU Länder. Diese Sonderpreise gelten exklusiv für Elektor Leser und bis zum 10.06.2013.

Batronix Elektronik | Lise-Meitner-Str. 1-7 | 24223 Schwentinental | Tel.: +49 (0)4307/8243-20 | Fax: +49 (0)4307/8243-39 | www.batronix.com | service@batronix.com

Flexible Bedienung

Programmierung von Menüs für Text-Displays

Auch auf einem kleinen Display lassen sich viele Parameter und Einstelloptionen eines Projekts anzeigen, wenn man geschickt eine Menüführung programmiert. Voraussetzung für eine bequeme Bedienung sind Taster, die sinnvoll neben dem Display angeordnet sind. Das Linux-Extension-Board bietet genau dies. Nun benötigt man nur noch passenden Code.

Von
Benedikt Sauter [1]

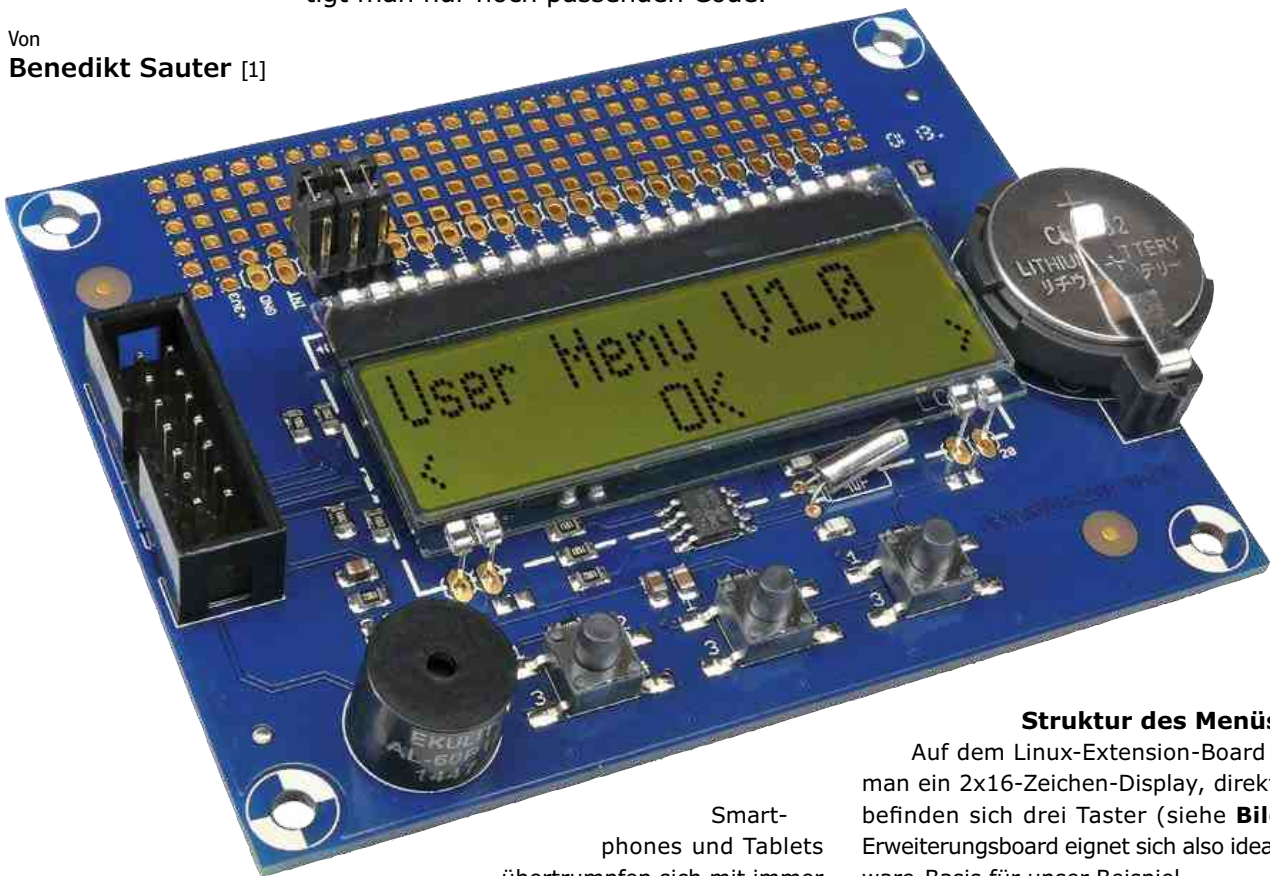


Bild 1.
Das Linux-Extension-Board bietet ein Text-Display und drei sinnvoll angeordnete Taster.

Smartphones und Tablets übertrumpfen sich mit immer feiner aufgelösten Displays, doch an vielen anderen Geräten findet man immer noch einfache Text-Anzeigen. Bei etlichen Druckern, Faxgeräten, NAS-Servern oder Mobiltelefonen kann man hier mit Tastern durch einfache Menüs navigieren, Einstellungen vornehmen beziehungsweise interne Statusinformationen abrufen. In diesem Artikel wird an einem Beispiel gezeigt, wie man eine solche Lösung für das eigene Projekt umsetzen kann.

Struktur des Menüs

Auf dem Linux-Extension-Board [2] findet man ein 2x16-Zeichen-Display, direkt darunter befinden sich drei Taster (siehe **Bild 1**). Das Erweiterungsboard eignet sich also ideal als Hardware-Basis für unser Beispiel.

Zuerst müssen wir eine Liste mit den Menüpunkten erstellen, die über diese Bedieneinheit einstellbar beziehungsweise einsehbar sein sollen. Ein einfaches Menü könnte wie folgt aussehen:

- Konfiguration
- Minimum-Temperatur
- Maximum-Temperatur
- Täglicher Startzeitpunkt

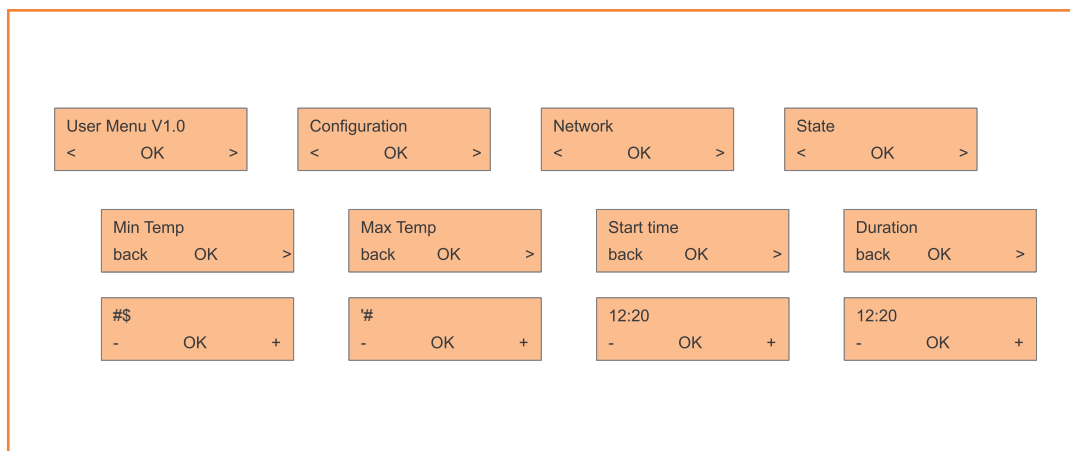


Bild 2.
Beispiel für eine
Menüstruktur.

- Maximale Dauer
- Netzwerkkonfiguration
- IP-Adresse: Dynamisch oder Statisch
- IP-Adresse
- Netzwerk-Adresse
- Gateway
- DNS-Server
- Status
- Live-Temperatur
- Prozessor-Auslastung
- Speicher-Auslastung
- Laufzeit der Maschine

Nachdem festgelegt ist, was der Benutzer sehen soll, benötigt man eine entsprechende Logik, die das Menü inklusive der Unterpunkte und der Aktionselemente realisiert. Solche Aktionselemente können sein:

- Statische Ausgabe
- Auswahl zwischen Optionen mit Bestätigung für eine Option
- Einstellen von numerischen Parametern (Zahlen)
- Einstellen von alphanumerischen Parametern
- Dynamische Ausgabe (Prozessbalken, Live-Werte, Uhrzeit u.a.)

Recht einfach umzusetzen sind die statische Ausgabe und die Auswahl (zum Beispiel Konfiguration der IP-Adresse dynamisch oder statisch). Schwieriger wird es beim Einstellen von Parametern und der dynamischen Anzeige von sich aktualisierenden Werten.

Geht man nun allzu naiv an die Sache heran, so wird das Endergebnis vermutlich ein typischer Spaghetti-Code sein. Das erfreuliche Gegenteil wäre eine flexible und erweiterbare Lösung, bei der man später fast intuitiv neue Menüpunkte einfügen kann, ohne sich tief in die Software einarbeiten zu müssen.

Lösungsvorschlag

Bei der Suche nach einer passenden Lösung sind zwei Vorschläge entstanden. Einmal die klassische Bibliothek, mit der man solch ein Programm umsetzen kann. In **Listing 1** ist der entsprechende Pseudo-Quelltext dargestellt. Mit den Funktionen `MainMenu` und `SubMenu` kann man eine Baumstruktur aufbauen, die einfach erwei-

Im Menü „Konfiguration“ lassen sich wichtige Einstellungen vornehmen. Die Minimum- und Maximum-Werte kann man entsprechend setzen, um etwa Schaltvorgänge oder ähnliches auszulösen. Für eine Netzwerkverbindung benötigt man ebenfalls ein paar Konfigurationsparameter. Zum einen muss eingestellt werden, ob die IP-Adresse dynamisch oder statisch konfiguriert wird. Bei der statischen Konfiguration muss man zusätzlich alle Parameter einzeln eingeben. Gute Software sollte natürlich darauf achten, dass nur IP-Adressen in gültigem Format eingegeben werden können.

Umsetzung des Menüs

Bei der Menüführung kann man sich von Geräten im eigenen Umfeld inspirieren lassen. Sicherlich kennt jeder von uns gelungene und weniger gelungene Ansätze.

In **Bild 2** haben wir die Menüführung skizziert, die wir nun umsetzen wollen. Es ist nicht das komplette, oben beschriebene Menü dargestellt, aber die restlichen Punkte kann man sich gut vorstellen.

Listing 1. Pseudo-Code für den Aufbau eines Menüs.

```
configuration = MainMenu(„Configuration“)
network      = MainMenu(„Network“)
state       = MainMenu(„State“)

//Menu Configuration
mintemp = SubMenu(configuration,„Min Temp“)
maxtemp = SubMenu(configuration,„Max Temp“)
start   = SubMenu(configuration,„Start Time“)
duration = SubMenu(configuration,„Duration“)

//Menu Network
ipchoose = SubMenu(network,„IP-Address: Dyn/Static“)
ip        = SubMenu(network,„IP-Address“)
netaddress = SubMenu(network,„Network“)
gateway   = SubMenu(network,„Gateway“)
dns       = SubMenu(network,„DNS-Server“)

//Menu State
temperature = SubMenu(state,„Temperature“)
cpu         = SubMenu(state,„CPU“)
ram         = SubMenu(state,„RAM“)
uptime      = SubMenu(state,„Uptime“)

...

RegisterMenu(mintemp,MinTemperature)
```

Listing 2. Pseudo-Code zum Einstellen eines Parameters.

```
Function MinTemperature()
{
    mintemp = ConfigRead(„mintemp“)
    switch(button)
    {
        case „button_left“:
            mintemp--
        case „button_ok“:
            Back()
        case „button_right“:
            mintemp++
    }
    ConfigWrite(„mintemp“,mintemp)
    Display(„%s“,mintemp)
}
```

terbar ist. Wird ein neuer Knoten angelegt, erhält man eine Referenz auf diesen als Rückgabewert. Mit einem Verweis auf diesen Wert kann man weitere Unterknoten anlegen. Mit `RegisterMenu` verbindet man die Knoten schließlich noch mit Funktionen im Code, die bei der Auswahl des jeweiligen Menü-Unterpunkts aufgerufen werden. Nun kommt die Implementierung dieser Funktionen an die Reihe. Wählt man einen Unterpunkt wie z.B. das Einstellen der minimalen Temperatur, dann muss ein Menü erscheinen, mit dessen Hilfe man einen Zahlenwert einstellen kann. Als Pseudo-Code könnte das Ganze wie in **Listing 2** aussehen.

In dieser und jeder anderen Funktion zur Einstellung eines Parameters muss es einen lesenden und schreibenden Zugriff auf eine Konfigurationsdatei oder Datenbank geben (`ConfigRead` und `ConfigWrite`). Die Tastendrucke werden in einer switch-case-Struktur ausgewertet. Der aktualisierte Temperatur-Wert wird mit dem Befehl `Display` angezeigt.

Auf der Elektor-Website zu diesem Artikel [3] kann man eine Beispiel-Implementierung in der Sprache C herunterladen.

Alternative

Elegant könnte man ein solches Menü auch mit dem (Linux-)Dateisystem realisieren. Entsprechend der Menüstruktur legt man einen Verzeichnisbaum an. In den Verzeichnissen können dann jeweils eigene Mini-Programme liegen, die sich um die Umsetzung des Menü-Unterpunkts kümmern. Die Ausgabe des Programms wird einfach 1:1 an das Display weitergeleitet.

Weblinks

[1] sauter@embedded-projects.net

[2] www.elektor.de/120596

[3] www.elektor.de/130044

Lesen Sie die neue Elektor ein Jahr lang in der ultimativen GOLD-Mitgliedschaft und profitieren Sie von allen Premium-Vorteilen!



Die Elektor-GOLD-Jahresmitgliedschaft bietet Ihnen folgende Leistungen/Vorteile:

- Sie erhalten **10 Elektor-Hefte** (8 Einzelhefte + 2 Doppelausgaben Januar/Februar und Juli/August) pünktlich und zuverlässig frei Haus.
- **Extra:** Jedes Heft steht Ihnen außerdem als PDF zum sofortigen Download unter www.elektor-magazine.de (für PC/Notebook) oder via App (für Tablet) bereit.
- **Neu & Exklusiv:** Sie erhalten alle 2 Wochen per E-Mail ein neues Extra-Schaltungsprojekt (frisch aus dem Elektor-Labor).
- **Neu & Exklusiv:** Wir gewähren Ihnen bei jeder Online-Bestellung 10% Rabatt auf alle unsere Webshop-Produkte – dauerhaft!
- **Neu & Exklusiv:** Der Online-Zugang zum neuen Community-Bereich www.elektor-labs.com bietet Ihnen zusätzliche Bauprojekte und Schaltungsideen.
- **Extra:** Die neue Elektor-Jahrgangs-DVD (Wert: 27,50 €) ist bereits im Mitgliedsbeitrag inbegriffen. Diese DVD schicken wir Ihnen sofort nach Erscheinen automatisch zu.
- **Extra:** Top-Wunschprämie (im Wert von 30 €) gibts als Dankeschön GRATIS obendrauf!



UMWELTSCHONEND – GÜNSTIG – GREEN

Möchten Sie Elektor lieber im elektronischen Format beziehen? Dann ist die neue GREEN-Mitgliedschaft ideal für Sie! Die GREEN-Mitgliedschaft bietet (abgesehen von den 10 Printausgaben) alle Leistungen und Vorteile der GOLD-Mitgliedschaft.



Jetzt Mitglied werden unter www.elektor.de/mitglied!

Raspberry Pi:

Ein Jahr und eine Million Computer später

Interview von
Clemens Valens
Übersetzung von
Rolf Gerstendorf



Im April 2012 berichtete Eben Upton, Gründer der Raspberry Pi Foundation in einem Elektor-Interview [1] vom Produktionsstart des Einplatinencomputers. Inzwischen wurden fast eine Million Boards verkauft. Wir trafen den „R-Pi“ Hardware-Spezialisten Peter Lomas auf der Embedded World 2013 in Nürnberg. Eine gute Gelegenheit, mit ihm den neuesten Stand des Projekts zu erörtern.

C: Raspberry Pi, ein Phänomen. Es ist schon erstaunlich, was da passiert ist.

P: Das ist es! Viele Leute fragen mich, warum aus Raspberry Pi so ein Hype geworden ist, was haben wir anders gemacht als andere? Der erste wichtige Grund könnte sein: Wir hatten einen unserer ersten Prototypen in einem UK-Blog. Einer der BBC-Korrespondenten, Rory Cellan-Jones, machte ein kleines YouTube-Video, das 600.000 Aufrufe erhielt.

C: Ich habe einen Blick auf die Webseite von Raspberry Pi geworfen und finde, sie ist für eure Zielgruppe – Jugendliche – nicht so einfach zu verstehen. Es ist nicht wirklich erklärt, was Raspberry Pi genau ist. Es gibt verschiedene Distributionen, man muss Linux kennen und in Python programmieren...

P: Nun, das ist wahr, und auf seltsame Weise ist das Teil des Erfolgs, denn du musst tatsächlich aktiv werden. Um etwas mit Pi anzufangen, reicht es nicht, es aus einer Hochglanzbox auszupacken, auf den Tisch zu legen und auf die „ON“-Taste zu drücken. Du hast tatsächlich einige geistige Arbeit zu verrichten. Du musst einige Dinge herausfinden. Und ich glaube, dass es ein Vorteil ist, wenn etwas funktioniert, weil du eine Leistung erbracht hast. Du hast etwas getan. Nicht „*Wir haben etwas getan*“, DU hast es selbst gemacht und dies ist eine Genugtuung.

C: Neben dieser Komplexität ist die Tatsache, dass man in Englisch programmieren muss [wegen Python, Anmerkung der Redaktion] für die nicht-englischsprachige Bevölkerung eine weitere Hürde. Deshalb ist Arduino so erfolgreich, es machte die Programmierung einfach. Es ist nicht nur eine billige Hardware.

P: Es gibt keinen Zweifel, Arduino ist ein geniales Produkt. Und du hast Recht, man kann sehr leicht ein „*Hello World*“ aufrufen. Aber auf einem Raspberry Pi schließt du die Kabel an und steckst die Karte ein und hast ein entsprechendes „*Hello World*“, nur bei uns ist das die Scratch-Katze. Sobald du die Scratch-Katze (**Bild 1**) aufgerufen hast, hast du eine Reihe von Möglichkeiten. Man kann Scratch mit einer I/O-Schnittstelle verwenden, um eine LED zum Leuchten zu bringen; oder man kann eine Taste drücken, um die Scratch-Katze zu bewegen. Ich habe festgestellt, dass Kinder das Ganze verstehen, solange es nicht allzu kompliziert ist. Es sind eher die Erwachsenen, die da Probleme haben.

C: Ich sah mindestens drei verschiedene Distributionen für die Boards. Was sind eigentlich die Unterschiede? Warum gibt es nicht nur eine?

P: Nun, sie alle bieten leicht unterschiedliche Funktionen. Letztendlich fokussiert haben wir uns nur auf Python – das ist es, woher der Name

„Pi“ kommt. Mit den verschiedenen Paketen sind wir in der Lage „different roots“ von Menschen zu unterstützen, die Pi verwenden wollen. Jetzt haben wir das RISC-OS, das sie verwenden können. Und die Leute sind sogar mit Bare-metal-Programmierung beschäftigt. Wenn wir nur eine Distribution hätten, würden wir dies ausschließen.

C: Okay. Ich habe die Distribution ausgesucht und möchte nun eine LED ansteuern, dann muss ich einen Treiber oder so etwas öffnen, nehme ich an?

P: Nun, du hast die Bibliothek und musst nur einen Bibliotheks-Aufruf tätigen. Wieder ist es aber nicht ganz so einfach: Du musst die Bibliothek suchen und finden und herunterladen. Jetzt kommt die Erweiterungsplatine Pi-Face [3] mit einer interaktiven Bibliothek ins Spiel, die zu Scratch führt. Das Gertboard [4] ist ein weiteres Add-on-Board, mit Treiberbibliotheken und einigen dazugehörigen Tutorials. Letztendlich kann man die GPIOs dann genauso einfach wie bei der „Bare-metal“-Programmierung ansprechen [also ohne ein Betriebssystem als Zwischenschicht, wie bei einem einfachen Mikrocontroller, Anmerkung der Redaktion].

C: Ich habe mir einige der Projekte angesehen und bin von der Vielzahl der Media-Player überrascht. Mit dieser Art von Anwendung bewirbt zum Beispiel RS das Board. Sind Sie davon nicht enttäuscht? Raspberry Pi wird von vielen Menschen nur als eine billige Plattform für eine Linux-Applikation angesehen, mit der sie ein Media-Center realisieren können.

P: Ich weiß genau, was du meinst. Ich sollte enttäuscht sein, dass einige Leute das Board nur kaufen, um es zu einem Media-Center zu machen und sonst nichts damit tun. Aber wenn nur fünf oder zehn Prozent der Menschen, die dies getan haben, denken „Na ja, das war aber einfach, vielleicht bekomme ich auch die eine oder andere Anwendung damit hin“, dann ist es doch ein Erfolg.

C: Kommen wir zu der Hardware auf dem Board.

P: Aber gerne!

C: Also, du hast einen Broadcom-Prozessor aus- gesucht, weil Eben bei Broadcom gearbeitet hat?

P: Er arbeitet immer noch bei Broadcom. Es wäre schwer für mich zu leugnen, dass es nicht der Grund für diese Entscheidung war, als Eben sagte: „Oh schau mal, hier ist ein phantastischer Chip. Er kann alles, was wir wollen, warum nehmen wir ihn nicht?“

Wir haben unseren guten Ruf auf die Webseite genagelt und versprochen, dass das Board 35 \$ kosten würde, da gab es keinen Weg zurück. Aber die grundlegenden Zahlen in unserer Kalkulation sahen plausibel aus, wir hatten nur eine Menge Arbeit, das Ganze mit den Preisen auch wirklich auf den Punkt zu bringen. Also, ich glaube, wenn wir einen anderen Weg gegangen wären, vielleicht mit Samsung, hätte es das Budget gesprengt.

C: Hat Broadcom in irgendeiner Weise geholfen?

P: Jeder Halbleiterhersteller half dem Projekt, indem er die Chips zur Verfügung stellte und dies zu einem guten Preis. Denn das große Problem ist, dass man nicht von vornherein das Volumen abschätzen kann. Also versuchst du, konservativ zu sein und von einigen tausend Boards auszu- gehen. Schnell waren wir bei 20.000 Boards, aber nirgendwo in unseren kühnsten Träumen hatten wir damit gerechnet, schon am Starttag 200.000 Boards zur Verfügung stellen zu müssen und innerhalb eines Jahres eine Million ver- kaufen zu können. Das hat in vielerlei Hin- sicht geholfen, denn es hat den Preis aller Bauteile nach unten getrieben. Ich würde nicht behaupten, es hätte die Anbieter von Komponenten, die vom ersten Tag an uns glaubten, nicht gefreut, damit etwas Geld zu verdienen. Wir folgten immer dem Grundprinzip der Nachhaltigkeit, dass die Foundation, die Community, die die Boards kauft und unsere Lieferanten alles tun, damit sich das Projekt finanziell trägt. Es wäre eine totale Katastrophe gewesen, wenn etwa Broad- com gesagt hätte: „Ich sag euch was, Leute, lasst euch die ersten 20.000 Prozessoren schenken“. Wir hätten dem Board allen möglichen Schnick- schnack hinzufügen können und nach dem Ver- kauf der ersten 20.000 Exemplare den Preis um 12 \$ anheben müssen. Ich glaube, das wäre das Ende von Raspberry Pi gewesen.

C: Broadcom hält sich aus irgendeinem Grund zum Prozessor ziemlich bedeckt. Ich glaube, du musstest ein NDA (Non-Disclosure Agreement) unterschreiben, um alle Details zu erhalten.

P: Immerhin haben wir es geschafft, die ARM- Seite öffentlich zu machen, aber die GPU [Graphi- cal Processing Unit] musste geschlossen bleiben.

C: Ich nehme an, du hast Zugang zu allen Informationen?



Bild 1.
Die Scratch-Katze,
eine Multi-Media-
Programmierungsumgebung für
Kinder.

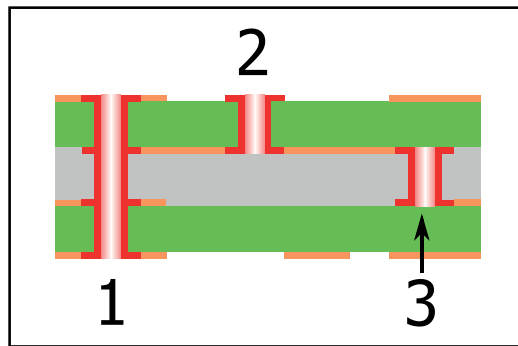


Bild 2. Drei verschiedene Vias: 1 – normal, 2 - blind; 3 - buried. Stellen Sie sich eine 6-Lagen-Platine mit *blind vias* zwischen den Layern 1 und 2 vor. Diese beiden Lagen müssen separat als doppelseitige Platine gefertigt und dann mit den anderen vier Layern verbunden werden. Diese Anordnung wird dann wieder gebohrt und durchverbunden. Insgesamt sind zwei Bohr- und Beschichtungsverfahren erforderlich anstelle von nur einem, wenn keine *blind* oder *buried vias* vorhanden sind.

P: Wir müssen den Zugang haben. Aber meine persönliche Sorge wäre auch, wenn wir uns da durcharbeiten und anfangen, etwa eigene Versionen von Bibliotheken zu veröffentlichen, dann könnten wir am Microsoft-Syndrom erkranken mit DLLs, die im Konflikt zueinander stehen, was eine enorme Komplexität zur Folge hätte. Und ich glaube auch, der Video-Kern ist das Kronjuwel des geistigen Eigentums von Broadcom. Es gibt viele andere Halbleiterhersteller, wo dies in Silizium realisiert ist, und niemand beschwert sich, nicht genau zu erfahren, wie das Stück Silizium funktioniert. Ich denke, solange es eine Schnittstelle gibt, die du benutzen kannst, solltest du das wie ein Stück Silizium ansehen. Du gibst ihr Befehle, sie macht einen Job für dich und reagiert mit einer Antwort.

C: War es für dich nicht ein Glücksspiel? Hattest du von Tag 1 an Zugriff auf alles oder musstest du Broadcom erst überzeugen, dass du Tausende von Boards verkaufen könntest, bevor sie mit dir reden würden?

P: Eben und noch einige andere der Jungs, die das Projekt unterstützen, arbeiten ja für Broadcom. Einige von ihnen haben sogar Teile von diesem Chip entwickelt und kennen ihn natürlich im Detail. Das ist sicherlich immer ein Vorteil.

C: Und wenn Eben und die anderen nicht für Broadcom gearbeitet hätten...

P: „Hätten wir einen anderen Chip verwendet?“ Nun, ich spekuliere mal, ich wäre herumgelaufen und hätte geschaut und verglichen und spätes-

tens, wenn der Preisfaktor ins Spiel gekommen wäre, wäre ich wieder bei unserem Chip gelandet, der beinahe ideal zu unseren Anforderungen passt. Ein Knackpunkt dabei war zweifellos die HDMI-Unterstützung. Aus technischer Sicht war eine der Herausforderungen die Leiterbahnführung unter dem BGA, weil *blind vias* und *buried vias* sehr teuer sind (**Bild 2**).

C: Wie viele Layer hat die Platine?

P: Sechs, die Layer-Anzahl ist Standard. Der einzige kleine Trick war, wir haben „blind vias“ nur auf die Layer eins und zwei platziert - so hatten wir zwar eine zusätzliche Bohrstufe, aber nur eine Bonding-Stufe. Das hat dann zwei Cents gekostet. Aber, weil die nächste Ebene darunter eine Massefläche war, musste ein Großteil der Anschlüsse aus dem Broadcom-Prozessor nur einen Layer tiefer gehen. Und das bedeutete, dass ich Raum darunter zum Routen anderer Dinge hatte.

C: Haben sie nicht Richtlinien bei Broadcom?

P: Oh, klar haben sie Richtlinien! „Verwenden Sie blind und buried vias und vias in Pads“. Unser erster Prototyp berücksichtigte all dies, aber die Kosten wären auf 100...110 \$ gestiegen. Also haben wir die Machete ausgepackt und all die Dinge, die wir nicht brauchten, weggehackt. Du hast alle Funktionen, die du wünschst, die Leistung, die du dir vorstellst, aber nichts extra.

C: Wie viele Versuche hast du bis zum ersten funktionsfähigen Board gebraucht?

P: Wir haben Prototypen gebaut. Sie haben nicht funktioniert. Wir mussten eine Modifikation vornehmen, um den Broadcom-Chip mit Strom zu versorgen. Ich glaube, ich hatte das Datenblatt missverstanden.

C: Wie bitte? Das Problem war, dass der Chip nicht mit Strom versorgt wurde?

P: Wenn der Chip gar nicht versorgt worden wäre, kein Problem. Aber die Hälfte des Chips lief, die andere nicht, weil es einige interne Spannungsversorgungen gibt, die extern versorgt werden müssen. Ich habe da etwas durcheinander gebracht. Aber jemand hat entdeckt, dass lediglich eine Durchkontaktierung mit einer kleinen Kupferfläche direkt daneben verbunden werden musste.

C: Nur ein Klecks Lötzinn?

P: Es war der sprichwörtliche Klecks Lötzinn. Ein Klecks Lötzinn und alles begann zu leben! Eines

der ersten Videos auf der Website zeigt Eben und mich, wie wir kleine Stückchen Lötstopplack abkratzen und Lötzinn auf die Stelle klecksen.

C: Ich habe Kritik von Unternehmen gehört, ihr tötet das Geschäft. Ihre Kunden erwarten jetzt Boards, die die Hälfte kosten. „Wenn Raspberry Pi nur 25 \$ kostet, warum soll ich für deine kleine Platine dann 100 \$ bezahlen?“ Was denkst du dazu?

P: Nun, meine Antwort darauf wäre, ihren Kunden sagen: „Bestellen Sie eine Million Exemplare und Sie können es zu diesem Preis haben.“ Es ist alles eine Frage der Skalierbarkeit. Wir konnten wegen der Unterstützung, die wir hatten, so skalieren. Aber ich glaube nicht, dass wir ernsthaften Schaden im Markt anrichten.

C: Was denkst du über die Zukunft von Raspberry Pi?

P: In unserer Branche garantiert Moores Gesetz, das alles nach zwei Jahren hoffnungslos veraltet ist. Also denken wir zwar über die Zukunft nach,

aber nicht mehr. Wir versuchen, unser Educational-Release zu verbessern. Ich will nicht behaupten, dass Raspberry Pi perfekt ist. Wir haben nur eine Modifikation am Board vom Design bis zur Veröffentlichung und nur einige kleinere Änderungen an der Version 2 vorgenommen. Einige davon beheben ein paar Unregelmäßigkeiten, einige andere waren nötig, um unserem neuen Produktionspartner Sony entgegenzukommen. Sony benötigte einige geringfügige Änderungen an der Platine, um sie einfacher fertigen zu können.

(130101)

Links & Quellen

- [1] Interview mit Eben Upton:
www.elektor.de/120228
- [2] Scratch: <http://scratch.mit.edu/>
- [3] Pi-Face: <http://pi.cs.man.ac.uk/interface.htm>
- [4] Gertboard: http://elinux.org/RPi_Gertboard

Anzeige

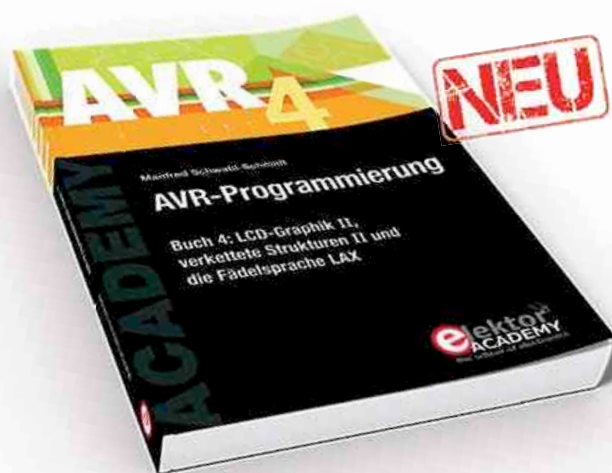
AVR-Programmierung 4

LCD-Graphik II, verkettete Strukturen II und die Fädelsprache LAX

In diesem vierten Band zur Programmierung von AVR-Mikrocontrollern wird die LCD-Graphik aus Buch 3 weiterentwickelt. Hinzu kommen das Füllen von Polygonen, die Zuordnung von Pixelkoordinaten zu Graphikobjekten und die Verwendung des Displays als Textfenster. Der Einstieg in die Animationstechniken reicht von einem einfachen blinkenden Cursor über die Simulation einer rollenden Billardkugel und Truchet-Kacheln bis zu einer animierten Sanduhr.

Aufbauend auf der Darstellung der inneren Mechanik von Fädelsprachen im vorigen Band wird die Fädelsprache LAX vorgestellt und implementiert. Aufbau, Verhaltensweise und Konstruktion der Sprach-elemente sind so genau beschrieben, dass eigene Ideen zur Sprachgestaltung erfolgreich umgesetzt werden können.

Der dritte Teil des Buches setzt die Einführung in die Welt der verketteten Listen fort. Hier werden die Betriebsalgorithmen einiger Listenvariationen diskutiert und implementiert. Höhepunkt ist jedoch die Anwendung der vorgestellten Techniken in einem ausführlichen Beispiel, das die Umsetzung einer Variation der Netzplantechnik in AVR-Code behandelt.



321 Seiten (kart.) • Format 17 x 23,5 cm
ISBN 978-3-89576-232-1
€ 46,00 • CHF 57,10



Weitere Infos & Bestellung unter
www.elektor.de/avr-prog4

Hexadoku Sudoku für Elektroniker

Wer seine kleinen grauen Zellen frisch halten will, muss diese regelmäßig trainieren. Hierfür eignen sich nicht nur die Elektronik-Projekte in unserem Heft, sondern auch unser Hexadoku. Nun müssen Sie nur noch die richtigen Zahlen herausfinden. Wer uns die Hex-Ziffern in den grauen Kästchen zuschickt, kann einen von vier Gutscheinen gewinnen.

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist. Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien)

genau einmal vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst - sprich die Zahlen in den grauen Kästchen herausfindet - kann wie jeden Monat einen Hauptpreis oder einen von drei Trostpreisen gewinnen!

Mitmachen und gewinnen!

Unter allen internationalen Einsendern mit der richtigen Lösung verlosen wir einen **Eurocircuits/Elektor-PCB-Service-Gutschein im Wert von 100 €** und drei **Elektor-Bücher-Gutscheine im Wert von je 50 €**.

Einsenden

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail, Fax oder Post an:
Elektor – Redaktion – Süsterfeldstr. 25 – 52072 Aachen
Fax: 0241 / 88 909-77 E-Mail: hexadoku@elektor.de
Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 31. Mai 2013!

Die Gewinner des Hexadokus aus der März-Ausgabe stehen fest!

Die richtige Lösung ist: **48C57**.

Der Eurocircuits/Elektor-PCB-Service-Gutschein im Wert von 100 € geht an: Yves Printems (Frankreich).

Einen Elektor-Gutschein über je 50 € haben gewonnen: Torsten Clever, Peter Raue und Arwin Vosselman.

Herzlichen Glückwunsch!

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

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

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.

• **Subscribe** to *audioXpress* magazine!

Do your **electronics speak** to you? Are the words **"audio," "vacuum tubes,"** and **"speaker technology"** music to your ears?

Then you should be **reading audioXpress!**

Recently acquired by The Elektor Group, *audioXpress* has been providing engineers with incredible audio insight, inspiration and design ideas for over a decade. If you're an audio enthusiast who enjoys speaker building and amp design, or if you're interested in learning about tubes, driver testing, and vintage audio, then *audioXpress* is the magazine for you!

What will you find in *audioXpress*?

- In-depth interviews with audio industry luminaries
- Recurring columns by top experts on speaker building, driver testing, and amp construction
- Accessible engineering articles presenting inventive, real-world audio electronics applications and projects
- Thorough and honest reviews about products that will bring your audio experiences to new levels

Choose from print delivery, digital, or a combination of both for maximum accessibility.

Subscribe to *audioXpress* at www.audioamateur.com today!

audioXpress



Von
Reginald Neale (USA)

Der Fluch des Sammelns



Fairchild 7050. Dieses Digitalvoltmeter benutzte ich in den 60er Jahren. Als eines der ersten setzte es auf „dual-slope integration“, was zu einem stabilen Nullpunkt führte. Da die Integrationszeit von der Netzfrequenz abhängt, werden Effekte von Netzstörungen schön unterdrückt. Bild: Marvin Collins.



V-742 Dosimeter. Das Skalenbild ist das was man sieht, wenn man das Dosimeter ins Licht hält und am anderen Ende durchs Okular schaut. Das Modell V-742 ist dazu gedacht, per Clip an der Kleidung befestigt zu werden und die kumulative Strahlungsdosis zu erfassen. So ein Dosimeter wird mit einer hohen Spannung geladen, was die Strichlinienanzeige zurücksetzt. Strahlung sorgt dafür, dass die Ladung reduziert wird, wobei die Anzeige dann nach rechts wandert. Das Instrument benötigt ein Spezialladegerät.



Brown Converter. Bei diesem „Konverter“ handelt es sich um einen Satz vibrierender Kontakte, eingesetzt in einem synchronen Detektor-Verstärker-System.



Klassischer Empfänger für Funkamateure. Ein Collins 2051 restauriert und fotografiert von Marvin Collins, W6OQI.

Heathkit. In den USA ein großer Name in Sachen Selbstbau-Elektronik. Dieses Labornetzteil für Röhren kaufte ich in den späten 50er Jahren.



Fluke 1941A Digitalzähler der frühen 70er Jahre mit Nixie-Röhren.

Weitere virtuelle Museen im Web

Wenn Ihnen das gefällt, interessieren Sie sich vielleicht auch für diese Seiten:

www.conradhoffman.com
www.oneillselectronicmuseum.com/page10g.html
www.electrictuff.co.uk/
www.tubecollector.org/list.php?L=:
www.unusualmuseums.org/
<http://wps.com/archives/tube-datasheets/>
www.ieee-virtual-museum.org/
www.seas.upenn.edu/~museum/
<http://frank.yueksel.org>
www.heathkit-museum.com/
http://semiconductormuseum.com/Museum_Index.htm
<http://silicongenesis.stanford.edu>

Webseiten über Webseiten... und sie ändern sich natürlich unentwegt. Warnung: Es sind unglaubliche Zeitschlucker- aber das wissen Sie vermutlich schon...

Die hier abgebildeten Teile sammelten sich bei mir im Laufe meines Berufslebens als Elektroniker, da ich wohl unfähig war, irgendetwas Elegantes, Exotisches, Tolles, Merkwürdiges, Ungewöhnliches oder etwas von außergewöhnlicher Qualität wegzuerwerfen. Dabei spielte es keine Rolle, ob man das jemals wieder gebrauchen könnte. Diese Retronik-Ausgabe sei deshalb all jenen gewidmet, die mit einem ähnlichen Laster geschlagen sind.



Galvanometer. Dieses Gerät von Leeds & Northrup kennt fast jeder, der „früher“ in einem wissenschaftlichen Labor tätig war. Per Lichtstrahl als masseloser Zeiger kann man damit sehr kleine Ströme detektieren. Das Teil ist etwa 20 cm hoch. Heutige Galvanometer werden hauptsächlich bei Scannern eingesetzt. Sie sind viel kleiner und stabiler und um Größenordnungen schneller. Bild: Conrad R. Hoffman.

Bücher!

Einige Bücher die sich mit der hier beschriebenen Technik beschäftigen:

70 Years of Radio Tubes, John W. Stokes, Vestal Press 1982

Procedures in Experimental Physics, J. Strong, Prentice-Hall 1938

The Amateur Scientist, C. L. Stong, Simon & Schuster 1960

Reference Data for Radio Engineers, Federal Telephone and Radio Corp. 1946

Geigerzähler aus den späten 40ern. Nach dem Zweiten Weltkrieg war die Öffentlichkeit von „Atomkraft“ fasziniert. Es gab sogar Leute, die mit solchen Apparaten nach Uran suchten.



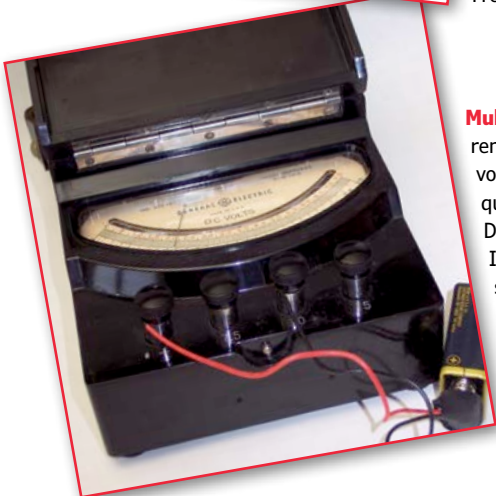
Philco-Oszilloskop. Es wurde zur TV-Reparatur in den 50er Jahren eingesetzt. Weder Zeitbasis noch Spannungsbereiche sind kalibriert. Auch die Synchronisation ist primitiv. Der graublaue Hammerschlaglack ist typisch für Messgeräte aus dieser Zeit. Bild: Marvin Collins.



Grid-Dipper aus den 50ern. Ein kontaktloses Messverfahren zur Erfassung von Resonanzfrequenzen. Die auf das Messgerät gesteckte Testspule muss mit der zu messenden Schaltung gekoppelt sein. Bei der Abstimmung mit dem rechten Drehknopf wird das Zeigerinstrument irgendwo einen scharfen Ausschlag machen, und dann kann man auf der oberen Skala die Frequenz ablesen. Für den Bereich von 0,6...300 MHz sind mehrere Spulen erforderlich.

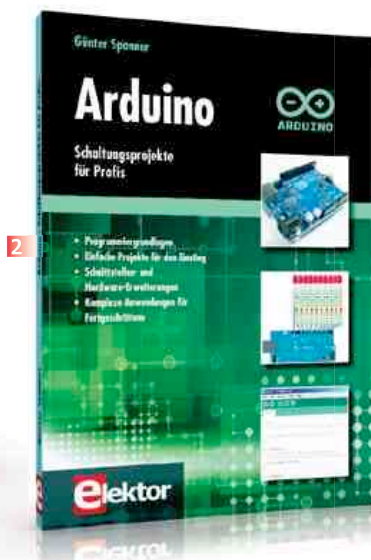
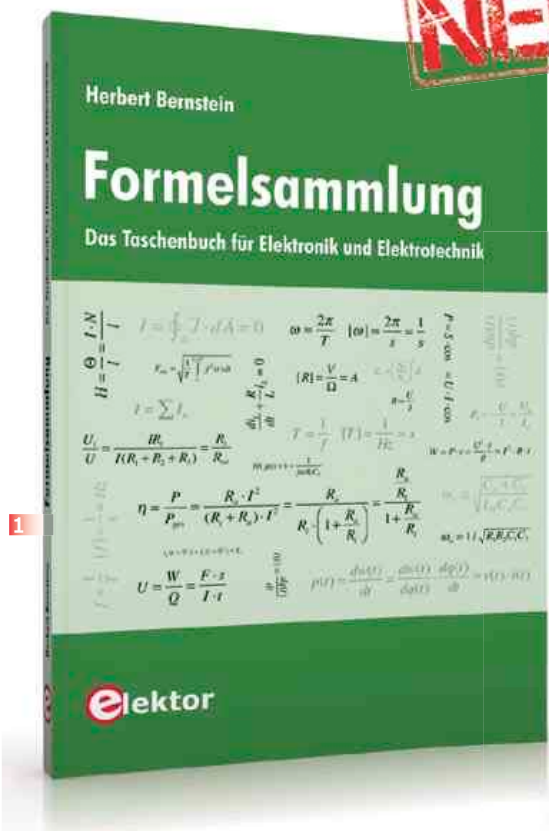


Multimeter von GE. Mehrbereichs-Labormessgerät aus den 50er Jahren. Ein modernes DVM hat eine standardisierte Eingangsimpedanz von 10 M Ω in allen Messbereichen, was die zu messende Spannungsquelle kaum belastet. Ganz anders die elektromechanische Messung: Die Impedanz hängt vom Messbereich ab und beträgt hier 200 Ω/V . Im 7,5-V-Bereich macht das 15 k Ω – niedrig genug, um reale Messungen zu beeinträchtigen. Ein weiteres Problem analoger Messgeräte: Der Zeiger liegt über der Skala. Deshalb der verspiegelte Bogen in der Skala. Schaut man so drauf, dass der Zeiger sein Spiegelbild verdeckt, wird der Parallaxe-Fehler vermieden.



Retronik ist eine monatliche Rubrik, die antiker Elektronik und legendären Elektor-Schaltungen ihre Referenz erweist. Beiträge, Vorschläge und Anfragen telegrafieren Sie bitte an Jan Buiting (editor@elektor.com).

NEU



Taschenbuch für Elektronik und Elektrotechnik

1 Formelsammlung

Diese „Formelsammlung“ beinhaltet alle wichtigen Details für Ingenieure, Techniker, Meister und Facharbeiter in der Elektrotechnik und Elektronik, die in Forschung, Entwicklung und Service tätig sind. Die logische Gliederung in zehn Kapitel vereinfacht das Nachschlagen und Aufsuchen der gewünschten Themen. In den einzelnen Kapiteln finden Sie immer die notwendigen mathematischen und physikalischen Formeln sowie die wichtigsten Tabellen.

271 Seiten (kart.) • ISBN 978-3-89576-251-2
€ 29,80 • CHF 37,00

Schaltungsprojekte für Profis

2 Arduino

Für den großen Erfolg der Arduino-Plattform lassen sich zwei Ursachen finden. Zum einen wird durch das fertige Board der Einstieg in die Hardware enorm erleichtert; der zweite Erfolgsfaktor ist die kostenlos verfügbare Programmieroberfläche. Unterstützt wird der Arduino-Anwender durch eine Fülle von Software-Bibliotheken. Die täglich wachsende Flut von Libraries stellt den Einsteiger vor erste Probleme. Nach einfachen Einführungsbeispielen ist der weitere Weg nicht mehr klar erkennbar, weil oft detaillierte Projektbeschrei-

bungen fehlen. Hier setzt dieses Buch an. Systematisch werden Projekte vorgestellt, die in verschiedene Themengebiete einführen. Dabei wird neben den erforderlichen theoretischen Grundlagen stets größter Wert auf eine praxisorientierte Ausrichtung gelegt.

270 Seiten (kart.) • ISBN 978-3-89576-257-4
€ 39,80 • CHF 49,40

Apps programmieren – Schritt für Schritt

3 Android

Smartphones und Tablet-Computer mit dem Betriebssystem Android finden immer weitere Verbreitung. Die Anzahl der Anwendungsprogramme – die sogenannten Applikationen oder kurz Apps – mit denen sich die Geräte individuell an die Vorlieben und Wünsche ihrer Benutzer anpassen lassen, steigt täglich an. Man ist bei der Individualisierung seines Smartphones aber nicht auf fix und fertige Applikationen beschränkt. Es ist einfacher als man denkt, Android-Geräte selber zu programmieren und eigene Apps zu schreiben. Dieses Buch bietet eine Einführung in die Programmierung von Apps auf Android-Geräten. Es erklärt leicht nachvollziehbar die Funktionsweise des Android-Systems und Schritt für Schritt die Programmierung von Applikationen.

256 Seiten (kart.) • ISBN 978-3-89576-252-9
€ 34,80 • CHF 43,20

Bestücke und getestete Platine

4 Elektor-Linux-Board

Linux läuft heutzutage auf den unterschiedlichsten Geräten – sogar in Kaffeemaschinen. Es gibt daher viele Elektroniker, die an Linux als Basis für eigene Controller-Projekte interessiert sind. Eine Hürde ist jedoch die scheinbar hohe Komplexität, außerdem sind Entwicklungsboards oft recht teuer. Mit diesem kompakten Modul, das bereits für modernste Embedded-Projekte fertig bestückt ausgestattet ist, gelingt der Linux-Einstieg ideal und preiswert zugleich.

Art.-Nr. 120026-91 • € 64,95 • CHF 80,60

Theorie und Praxis mit WinFACT und Multisim

5 Regelungstechnik

Die heutige Regelungstechnik hat Verknüpfungspunkte mit fast jedem technischen Gebiet. Ihre Anwendungen reichen von der Elektrotechnik über die Antriebstechnik und den Maschinenbau bis hin zur Verfahrenstechnik. Will man nun die Regelungstechnik anhand der fachlichen Regeln dieser einzelnen Gebiete erklären, so müsste man von einem Regelungstechniker verlangen, jedes Fachgebiet, in dem er Regelungen vornehmen will, fundiert zu beherrschen. Dies ist aber bei dem heutigen Stand der Technik nicht möglich. Bei der Regelung einer Antriebsaufgabe, einer



Druck- oder einer Temperaturregelung tauchen Gemeinsamkeiten auf, die man mit einer einheitlichen Vorgehensweise beschreiben kann. Die Grundgesetze der Regelungstechnik gelten in gleicher Weise für alle Regelkreise, ganz unabhängig davon, wie verschieden sie im Einzelnen auch apparativ aufgebaut sein mögen. Dieses Buch richtet sich an den Praktiker, der gründlicher in die Regelungstechnik eindringen möchte, auf ausschweifende theoretische Exkursionen in die Mathematik aber gerne verzichten kann.

365 Seiten (kart.) • ISBN 978-3-89576-240-6
€ 49,00 • CHF 60,80

Einstieg in die Praxis

6 LabVIEW 1

Das LabVIEW-Programmpaket ist ein international anerkannter Standard zur Entwicklung und Gestaltung von Messgeräten und Prozesssteueroberflächen. Seine Universalität konfrontiert den LabVIEW-Einsteiger allerdings mit einer unübersichtlichen Vielfalt von Funktionen, die er ohne fundierte Anleitung kaum überblicken kann. Hier setzt diese neue mehrteilige Lehrbuchreihe an: Von Grund auf werden in einfach nachvollziehbaren Schritten der Aufbau, die Struktur und die Verwendung von LabVIEW erklärt, in praktischen Beispielen dargestellt und mit Übungen vertieft. Der erste Band



erläutert die Grunddatentypen und die zugehörigen numerischen Grundfunktionen ebenso ausführlich wie die elementaren Programmstrukturen.

240 Seiten (kart.) • ISBN 978-3-89576-253-6
€ 34,80 • CHF 43,20

35 Einsteiger-Projekte in C

7 ARM-Mikrocontroller 1

Die Projekte in diesem Buch sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Das heißt nicht, dass diese Projekte einfach sind. Sie sind aber einfach zu verstehen. Es wird beispielsweise die USB-Verbindung zur Kommunikation benutzt, eine Methode, die im mbed-Board so einfach integriert ist, dass sie sich auch für ein Einsteiger-Buch eignet. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung. Es bedeutet, dass man keinerlei Software auf seinem PC installieren



muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC.

261 Seiten (kart.) • ISBN 978-3-89576-262-8
€ 39,80 • CHF 49,40

Kompletter Elektor-Jahrgang 2012 auf DVD

8 Elektor-DVD 2012

Die neue Elektor-Jahrgangs-DVD enthält alle Artikel des Jahrgangs 2012. Sie verfügt über eine sehr übersichtlich gestaltete Benutzeroberfläche. Mit der Elektor-DVD 2012 können Sie Platinenlayouts in perfekter Qualität drucken; diese Layouts mit einem Zeichenprogramm verändern; die Schnellsuchfunktion benutzen, mit der Sie in den einzelnen Artikeln oder im ganzen Jahrgang nach Wörtern, Bauteilen oder Titeln suchen können; Schaltbilder, Platinenlayouts, Illustrationen, Fotos und Texte exportieren.

ISBN 978-90-5381-273-0

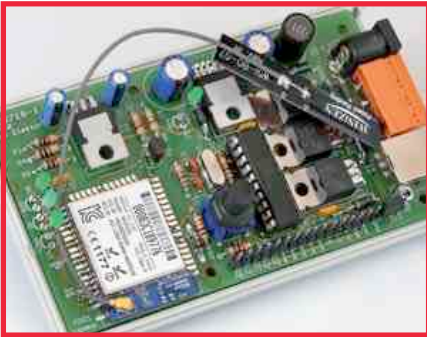
€ 27,50 • CHF 34,10

Weitere Informationen zu unseren Produkten sowie das gesamte Verlagssortiment finden Sie auf der Elektor-Website:

www.elektor.de/shop

Elektor-Verlag GmbH
Süsterfeldstr. 25
52072 Aachen
Tel. +49 (0)241 88 909-0
Fax +49 (0)241 88 909-77
E-Mail: bestellung@elektor.de

●Nächsten Monat in Elektor



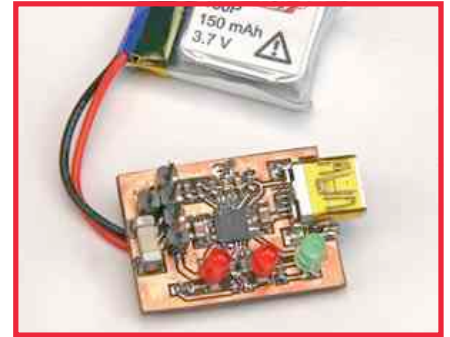
WLAN-Controller-Board

Steuerungen für LED-Beleuchtungen gibt es bereits in vielen Varianten. Doch auch hier möchte Elektor etwas Besonderes beitragen: Im nächsten Heft erscheint ein Controller-Board, mit dem über WLAN beispielsweise ein LED-Strip gesteuert werden kann. Auch Leistungsrelais, Garagentorantriebe, Roboter oder Modellbahnanlagen gehören zu den potentiellen Objekten. Universell wird das Board durch den Bootloader im Mikrocontroller, einen Erweiterungsport auf der Platine sowie Schutzdioden für den Anschluss induktiver Lasten.



Breitbandiger FM-Sender

Der Frequenzbereich dieses kleinen Senders ist das 70-cm-Amateurfunkband. Mit dem Sender können breitbandig FM-Signale, beispielsweise Audiosignale von HiFi-Anlagen oder ähnlicher Quellen übertragen werden. Pluspunkte sind nicht nur die hohe Übertragungsqualität, sondern auch die Fähigkeit zum Duplex-Betrieb. Im 70-cm-Band steht dem breitbandigen Duplex-Betrieb nichts im Weg, weil dieses Band einen weiten Frequenzumfang hat. Auch zwischen mehreren Stationen ist ein gleichzeitiger Duplex-Betrieb möglich.



Lithium-Ionen-Akkulader

Die Neuverwendung von Li-Ion-Akkus, die aus veralteten oder defekten mobilen Geräten stammen, ist nicht immer problemlos. Oft ist die Ladeüberwachung im auszumusternden System integriert, ein gesonderter Akkulader ist nicht vorhanden. Hier hilft der Bau eines globalen Laders für gebrauchte (oder auch neue) Li-Ion-Zellen weiter. Der vorgestellte Li-Ion-Akkulader kommt ohne Mikrocontroller aus, er arbeitet mit einem speziellen Baustein aus dem Sortiment von Maxim. Den Ladezustand des Akkus signalisiert dieser Baustein über eine LED-Reihe.

Elektor Juni 2013 erscheint am 22. Mai 2013.

Änderungen vorbehalten!

Rund um die Uhr und
sieben Tage die Woche

Projekte, Projekte, Projekte:
www.elektor-labs.com
Machen Sie mit!

CIRCUIT CELLAR

Subscribe Now!



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

**12 issues
per year for just**

Print OR Digital: **\$50** :: Combo (Print + Digital): **\$85**

www.circuitcellar.com/subscription

- ✓ über 40 Jahre Erfahrung ✓ über 40.000 Produkte am Lager
✓ schneller 24-Std.-Versand ✓ kein Mindermengenaufschlag

Kundenbewertungen:



Rund 98 % unserer Kunden sind vom reichelt-Service überzeugt*

* Quelle: Shopauskunft.de (21.02.2013)

Messtechnik – präzise & preiswert!

Professionelle Qualität zu attraktiven Preisen!

UNI-T

Digital-Speicher-Oszilloskop

mit hervorragenden Leistungsdaten und USB-Schnittstelle!

- ✓ extra großes Full-Color-Display
- ✓ 25 MHz Bandbreite
- ✓ 2 Kanäle



249,00

UTD 2025 CL



UNSERE BESTSELLER

UNI-T Digital-Multimeter mit USB-Schnittstelle

- LC-Display, hintergrundbeleuchtet, 6000 Counts
- Grundgenauigkeit: 0,5 %
- Sicherheit: CAT II 600 V, CAT III 300 V
- Widerstands-, Frequenz-, Kapazitäts-, Temperatur- sowie AC/DC-Strom- & Spannungsmessung



UT 61C

44,95

Infrarot-Thermometer

- Messbereich: -18 bis 380 °C
- Entfernungs-/Messpunktverhältnis: 10:1
- Festwertemission: 0,95
- Auflösung: 0,1 °C
- Wiederholungsgenauigkeit: ± 0,5 °C oder ± 0,5 %
- Messgenauigkeit: ± 2 °C/± 2 %



UT 300B

24,95

AC/DC-Digital-Zangenmessgerät

- bis 400A
- Klemme: Ø 28 mm
- 3 ¼-stelliges LC-Display, 3999 Counts
- Sicherheit: IEC 61010, CAT II 600V, CAT III 300V



UT 203

39,95

Handheld-Digital-Speicher-Oszilloskop

Verbindet die Funktionalität eines Oszilloskops mit der einer Digital-Multimeters.

- 1-Kanal-DSO mit 25 MHz Analogbandbreite
- Abtastrate: 200 MS/s (Echtzeit)
- hochauflösendes LC-Display, 8,89 cm/3,5", 320 x 240 Pixel
- Digital-Multimeter: 4000 Counts
- USB-Schnittstelle, inkl. Software



UTD 1025 CL

295,95

Echtheffektivwert-Digital-Multimeter

- hintergrundbeleuchtetes LC-Display, 40.000 Counts
- Grundgenauigkeit: 0,025 %
- Sicherheit: CAT III 1000 V, CAT IV 600 V
- USB-Schnittstelle, inkl. Software
- Datenloggerfunktion: 100 Werte
- Widerstands-, Kapazitäts-, Frequenz-, Temperatur- sowie AC/DC-Strom- und Spannungsmessung



UT 71C

135,00

Echtheffektivwert-Digital-Tischmultimeter

Universalmessgerät mit einer Top-Ausstattung und einer Multifunktionsanzeige mit 40.000 Digits!

- 4 ¾-stelliges Dual-Display, 40.000 Counts
- Grundgenauigkeit: 0,025 %
- Widerstands-, Kapazitäts-, Frequenz-, Temperatur- sowie AC/DC-Strom- und Spannungsmessung
- Dioden- und Durchgangsprüfung
- Relativwertmessung • Datenloggerfunktion
- USB-/RS232-Schnittstelle, inkl. Software
- Sicherheit: CAT I 1000 V, CAT II 600 V



179,00

UT 804

Jetzt bestellen! www.reichelt.de

Bestell-Hotline: +49 (0)4422 955-333

reichelt elektronik
Ihr kompetenter Partner für

Bauelemente • Stromversorgung • Messtechnik • Werkstattbedarf
Haus- & Sicherheitstechnik • Netzwerk- & PC-Technik • Sat-/TV-Technik

Neue Katalogausgabe!
Kostenlos – Jetzt anfordern!

