

e lektor

Checkliste für
Einsteiger:
- Wie funktioniert's?
- Wie setze ich das ein?
- Was kostet mich das?

FPGA

Endlich im Griff!

- Arduino @ Android | **Linux-Board im Netzwerk** | Elektronik-Grundlagen | **USB-I/O-Kabel** | LED-Weihnachtsbaum
- Internet der Dinge ● Was tut sich im Elektor-Labor?
- Bauelemente-Tipps** ● Neues aus der Community

Quarzlose 8-Bit USB PIC® Mikrocontroller senken Systemkosten und Stromverbrauch

0,25% Taktgenauigkeit ermöglichen USB-Anbindung, und der externe Quarz erübrigt sich



Microchips günstigste und kleinste USB-Mikrocontroller (MCUs) bieten 14 bis 100 Pins und sind die ersten 8-Bit-MCUs, die LCD-Ansteuerung, batteriegestützte RTCC und USB auf einem einzigen Chip unterstützen.

Microchips neueste USB PIC® MCUs verfügen über integrierte Taktquellen mit 0,25% Genauigkeit und ermöglichen so eine USB-Anbindung ohne externen Quarz. Sie sind auch die ersten USB MCUs, die 14 bis 100 Anschlüsse bieten, mit zahlreicher Peripherie ausgestattet sind und bis zu 128 KB Flash enthalten. Die eXtreme Low Power (XLP) Technologie senkt den Stromverbrauch auf nur 35 µA/MHz im Aktivmodus und auf 20 nA im Sleep-Modus.

Kostengünstig und klein

Die PIC16F145X MCUs bieten USB-Anbindung und kapazitive Touch-Sensorik, zusammen mit umfangreicher Peripherie auf einer Stellfläche bis hinab auf 4 mm x 4 mm.

Hochleistungsfähige Touch-Sensorik mit USB

Mit einer integrierten Charge Time Measurement Unit (CTMU) und 1,8- bis 5-V-Betrieb sind die PIC18F2X/4XX50 MCUs pinkompatibel zu vorherigen PIC18 MCUs, was den Übergang zu mehr Leistungsfähigkeit vereinfacht.

USB plus LCD-Ansteuerung und RTCC mit Vbat

Die PIC18F97J94 Familie bietet USB-Anbindung, LCD-Ansteuerung und einen batteriegestützten Real-Time Clock Calendar (RTCC) – und das in einem einzigen 8-Bit PIC® Mikrocontroller.

EINFACHER START IN 3 SCHRITTEN:

1. Wahl der Peripherie und Pinzahl für Ihre Anwendung
2. Kostenlose USB-Stacks und Softwaretreiber für ein schnelleres Design nutzen
3. Sofort mit der Entwicklung beginnen – mit kostengünstigen Tools

Weitere Informationen unter: www.microchip.com/get/eu8bitUSB



Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

ein SOCKEL

für alle click BOARDS

Stecken Sie einfach
das **Click™** board
Ihrer Wahl in den
fortschrittlichen
neuen **mikroBUS™**
und schon kann es
los gehen.



\$49⁰⁰



\$45⁰⁰



\$41⁹⁰



\$29⁰⁰



\$21⁰⁰



\$18⁰⁰



\$19⁰⁰



\$21⁰⁰



\$13⁹⁰



\$44⁰⁰



\$17⁹⁰



\$26⁰⁰



\$49⁰⁰



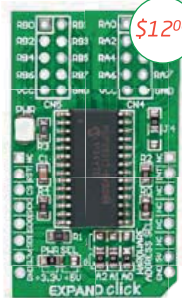
\$29⁰⁰



\$10⁰⁰



\$19⁰⁰



\$12⁰⁰



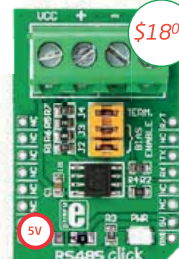
\$25⁹⁰



\$18⁰⁰



\$21⁹⁰



\$18⁰⁰



\$24⁰⁰



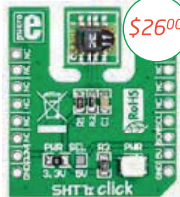
\$10⁰⁰



\$21⁰⁰



\$15⁰⁰



\$26⁰⁰



\$8⁰⁰



\$13⁹⁰



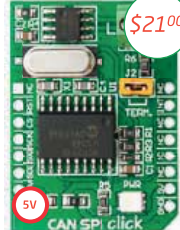
\$16⁹⁰



\$9⁰⁰



\$15⁰⁰



\$21⁰⁰

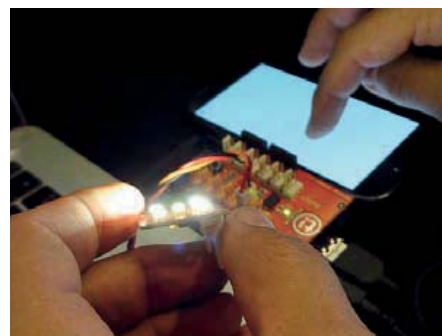
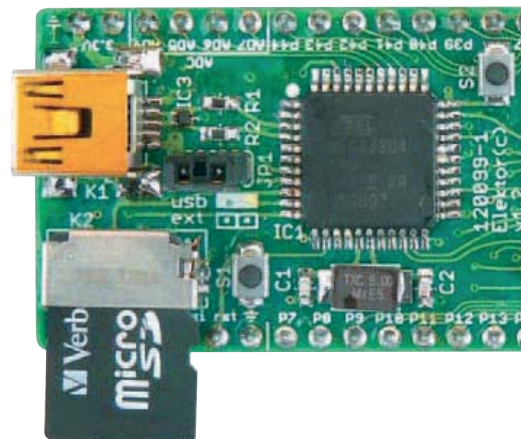


\$21⁰⁰



\$24⁰⁰

elektor



● Community

- 6 Impressum**
- 8 Elektor World**
Briefe, E-Mails und Ideen aus der Elektor-Community
- 10 Elektronik-News**
Gelungene Premiere
Arduino mit ARM

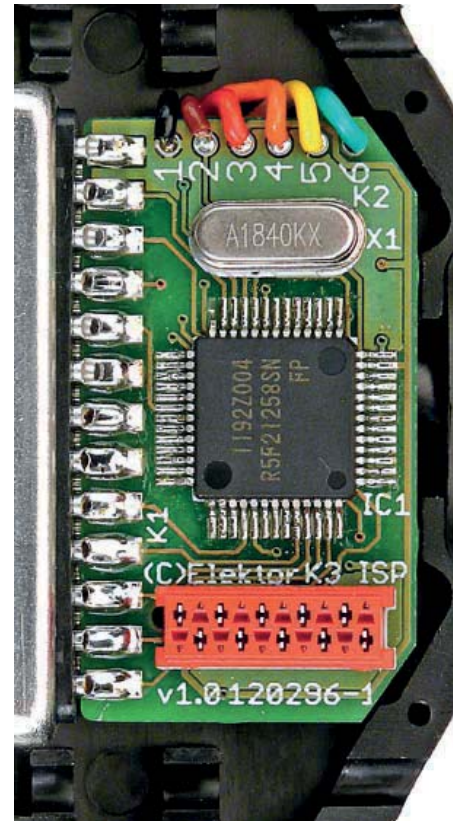
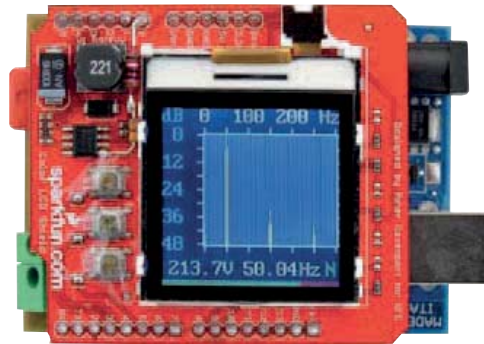


● Labs

- 12 Elektor .Labs**
Neue Projekte
auf der Community-Website
Eine Auswahl der Redaktion
- 48 Klein aber oho!**
Warum lässt sich bei unserer
Schaltung keine Firmware flashen?
- 50 Topfgucken**
Ist das nicht das neue
Elektor-Schaltnetzteil?
- 52 Massefläche weg?**
Platine mit PCB-Visualizer testen!
- 54 Bauelemente-Tipps**
Diesen Monat stehen zwei ICs
im Fokus des Interesses, mit
denen man eine kompakte,
temperaturgesteuerte
Lüfterregelung realisieren kann.

● Projects

- 14 Bauen Sie Ihren Chip!**
Zu den aktiven Komponenten,
die sich durch enorme Flexibilität,
aber auch durch hohe Komplexität
auszeichnen, zählen zweifellos die
FPGAs. Doch Schwellenangst war
gestern, unser Elektor-FPGA-Board
hilft, diese Hürde zu nehmen!
- 22 USB-IO24-Kabel**
Mit unserem Interface-Kabel lassen
sich bis zu 24 digitale Signale
einfach über die USB-Schnittstelle
eines PCs einlesen und ausgeben.
Wahlweise kann man auch bis zu
acht analoge Signale erfassen,
PWM- und Servo-Signale erzeugen
und mehr!
- 32 Netzspannungs-Analyser mit
Arduino**
Hier wird mit einem Arduino-
Board, einem Filter-Shield
und einem Display-Shield ein



Spektrumanalyse realisiert, der die „Verunreinigungen“ im Niederspannungsnetz sichtbar macht.

44 LED-Weihnachtsbaum

Die Glühlampenkettchen früherer Zeiten, die höchstens monoton blinken konnten, räumen ihren Platz für LED-Arrays und Lichtshow-Effekte!



56 Arduino auf Kurs (3b)

Diesen Monat zeigen wir, wie man mit einem Smartphone Sensoren auslesen kann, die an Arduino angeschlossen sind.

62 Embedded Linux leicht gemacht (6)

Das Baukasten-System Linux bietet sich perfekt als Plattform für Komplettlösungen im Bereich Messen, Steuern und Regeln an. In dieser Folge binden wir das Elektor-Linux-Board in ein typisches Heim-Netzwerk ein.

68 Zurück zu den Wurzeln (10)

Mit der Hochfrequenz fing alles an, die Funktechnik war der Motor der Elektronik. Auch heute noch bleiben HF-Experimente auf Mittel- und Kurzwelle gerade für Einsteiger äußerst spannend!

● Tech the Future

75 Das Internet der Dinge

Das IoT (Internet of Things) kommt! Elektor hatte die Gelegenheit, mit Experten über die Zukunft des IoT, die technischen Herausforderungen und auch eine mögliche Bedrohung zu diskutieren.



● Magazine

78 Hexadoku

Sudoku für Elektroniker

82 Nächsten Monat in Elektor

Impressum

43. Jahrgang, Nr. 504 Dezember 2012
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-medienservice.de
Es gilt die Anzeigenpreisliste Nr. 42 ab 01.01.2012

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 Schutzrechtinhabers 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.

© 2012 elektor international media b.v.
Druck: Senefelder Misset, Doetinchem (NL)

Treu sein lohnt sich

Im letzten Heft habe ich es schon angekündigt: Ab dieser Ausgabe hat unsere Zeitschrift eine frische Aufmachung. Auch die Rubriken haben sich geändert. Die Elektor-**Community** wächst international immer mehr zusammen. Hinweise und Ideen von anderen Elektronikern werden wir Ihnen daher aus aller Welt anbieten, unter dem Titel „Elektor World“. Interessante Projekte aus unserer **.Labs**-Website stellen wir in der gleichnamigen Rubrik vor. Hier erfährt man auch, an welchen Projekten unsere Entwickler gerade arbeiten und auf welche Herausforderungen sie dabei stoßen... Dazu kommen Bauteil-Tipps und Tricks, die im Entwickler-Alltag hilfreich sind. Die neue Technik-Rubrik **Tech the Future** ist nach unserer englischsprachigen Schwester-Website benannt. Meine Kollegin Tessel Renzenbrink wird hier Neuigkeiten aus der Forschung aufgreifen. Für diese Ausgabe befragte Tessel drei junge Ingenieure zum immer wichtiger werdenden „Internet der Dinge“. Es versteht sich von selbst, dass die Rubrik **Projects** das Herzstück unseres Heftes ist und den weitaus größten Raum einnimmt. In dieser Ausgabe ist uns – wie ich finde – wieder eine attraktive Auswahl gelungen. So konnte ich es kaum erwarten, dass es jetzt endlich mit dem FPGA-Projekt losgeht! Noch einmal daran erinnert sei, dass unsere Abo-Mitglieder alle zwei Wochen ein neues Extra-Projekt zugesendet bekommen. Darüber hinaus können die Abonnenten alle Hefte und Artikel in digitaler Form herunterladen, die ab dem Beginn des Abos erschienen sind (die Ausgaben sind ab dem Jahr 2000 online verfügbar). Es lohnt sich also mehr denn je, uns treu zu sein!



Jens Nickel

Unser Team

Chefredakteur:	Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)
Ständige Mitarbeiter:	Dr. Thomas Scherer, Christopher Rausch
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
Herausgeber:	Donatus Akkermans
Grafik & Layout:	Giel Dols, Mart Schroijen



Germany

Ferdinand te Walvaart
+31 46 4389417
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
+27 78 2330 694
j.dijk@elektor.com



China

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

Unser Netzwerk



Die Elektor-Community



Unsere Partner und Sponsoren



Allnet
www.synergy21.de31



Beta Layout
www.pcb-pool.com47



DesignSpark
www.designspark.com53



ELV
www.elv.de40 – 43, Beilage



Eurocircuits
www.elektorpcbservice.com49



Jackaltac
www.jackaltac.com55



LeitOn
www.leiton.de55



Linx Technologies
www.linxtechnologies.com11



Microchip
www.microchip.com2



MikroElektronika
www.mikroe.com3



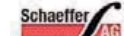
Pico
www.usbmso.com/PS17877



Reichelt
www.reichelt.de84



Ribu
www.ribu.at Beilage



Schaeffer AG
www.schaeffer-ag.de74

Sie möchten Partner werden?

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

Elektor World

Von Jan Buiting

Elektors „Verbesserter Strahlungsmesser“ im Weltraum

Hallo. Ich bin Michelle Boyce, Mitglied der SpaceConcordia.Ca. Kürzlich haben wir einen Würfel-Satelliten (ConSat-1) für den kanadischen Satelliten-Design-Wettbewerb gebaut.

Enthalten ist ein Halbleiter-Strahlungsmesser, der auf Artikeln von Elektor basiert:

- Messung von Gammastrahlen mit einem Fotodioden-Strahlungsmesser (mit BPW34) von Burkhard Kainka, Juni 2011.
- Verbesserter Strahlungsmesser für Alpha-, Beta- und Gammastrahlung von Burkhard Kainka, November 2011.

Das Projekt war Teil eines „Educational Outreach Programms“.

Viele Grüße!

Michelle Boyce

Das Strahlungsmesser-Projekt war außerordentlich erfolgreich. Als diese Nachricht aus Kanada eintraf, war uns auf einmal klar, dass Elektors Einfluss bis in die unendlichen Weiten des Weltraums reicht ;-)

Wir haben Michelle gebeten, einen Artikel für Elektor zu schreiben.

Elektors „Verbesserter Strahlungsmesser“ @ Oxford University...

Hallo Thijs, ich arbeite in der Abteilung für Physik an der University of Oxford. Wir haben einen Strahlungsmesser gebaut, um mit einem Helium-Ballon kosmische Strahlung zu messen.

Wir haben einige unterschiedliche Bausätze eingekauft, um sie in unserem Labor zu testen, dabei war auch der Strahlungsmesser von Elektor. Wir haben diverse Strahlungsquellen zur Verfügung, doch gelang es uns nicht, die Schaltung zum Laufen zu bringen.

Ich denke, dass wir die Schaltung korrekt aufgebaut und abgeschirmt haben, doch konnten wir den Fehler nicht finden. Sie triggert nämlich nur, wenn der Komparator auf L68 gesetzt ist (darüber triggert es nicht). Das Signal an K4 hat laut Oszilloskop etwa 500 mV_{SS}.

Wäre es möglich, dass wir eure Schaltung zu Vergleichszwecken ausgeliehen bekommen oder könnte ich die von mir gebaute einsenden (sie steckt in einem Metallgehäuse)?

Jeff Lidgard

Wie? Universität Oxford und funktioniert nicht? Ein funktionierender Prototyp ist schon unterwegs zu Dir, Jeff.

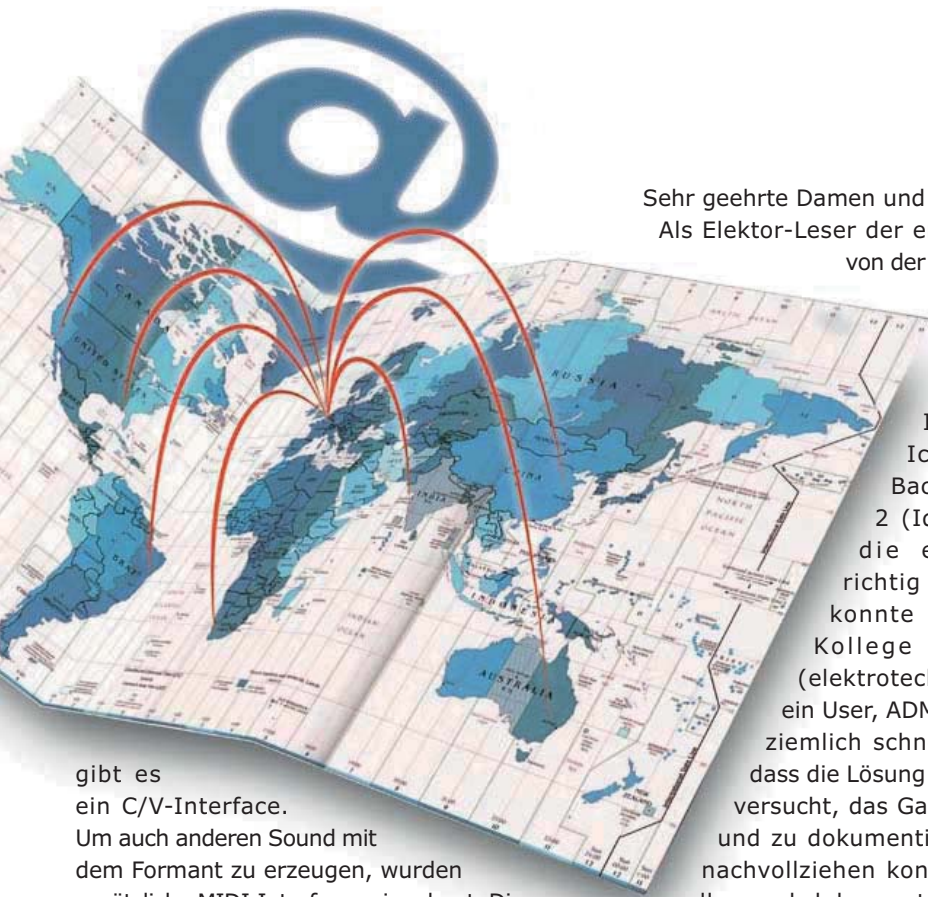
Elektor Monster-Formant auf YouTube

Siegfried Brückner hat vermutlich die größte Ansammlung von Elektor-Formant-Modulen in diesem Universum! Grob geschätzt dürfte sein Instrument drei Meter breit und 1,20 m hoch sein. Es ist mit einem Sequenzer, einem Bass-Pedal und einer Klaviatur verbunden. Insgesamt 110 Formant-Module, 500 Knöpfe, und Dutzende roter Patch-Kabel produzieren einen einzigartigen Formant-Synthesizer-Sound [1].



Siegfried schrieb uns: „In Phase 3 bestand das Projekt aus über 160 Modulen der Elektor-Formant-Bücher 1 & 2 sowie einigen Spezialitäten und diskreten Schaltungen. Alles ist über eine Modul-Wand verbunden. Über ein Achtwege-MIDI-Interface kann der Formant über MIDI und CV/Gate gesteuert werden.

Die neue Formant-Klaviatur hat nun eine vierte und fünfte Oktave sowie zusätzlich ein Pedal mit 13 Tasten. Für jede Klaviatur und jedes Pedal



Sehr geehrte Damen und Herren!

Als Elektor-Leser der ersten Stunde war ich von der erfrischenden Idee des Labyrinths recht angetan und habe Kopien bei meinen Kollegen aus dem IT-Bereich verteilt.

Ich begann mit dem Backtracing der Lösung 2 (Idee: niemand bietet die erste Lösung als richtig an!). Die Lösung 2 konnte ich nachvollziehen.

Kollege Sebastian Scholz (elektrotechnisch gesehen eher ein User, ADMIN halt) meldete sich ziemlich schnell mit der Aussage, dass die Lösung 1 richtig sei. Ich habe versucht, das Ganze nachzuvollziehen und zu dokumentieren: soweit ich das nachvollziehen konnte (ADMINs wissen alles und dokumentieren NICHTS!) sind mindestens 2 Lösungen richtig.

Dipl.-Ing. Peter Wendt

gibt es ein C/V-Interface. Um auch anderen Sound mit dem Formant zu erzeugen, wurden zusätzliche MIDI-Interfaces eingebaut. Diese ermöglichen neben dem monophonen Spiel auch Polyphonie (Expander). Die Expander-Sounds können bei Bedarf modifiziert werden.

Im April 2012 war ich zum ersten Mal in der Lage, meinen Monster-Formant anlässlich der Veranstaltung „Happy Knobbing“ in Rheinland Pfalz anderen Spezialisten vorzustellen. Das Motto dieses Jahres war nämlich „Sound Generation Using a Modular System“ [2]. Die Tonaufzeichnung ist leider nicht ganz so toll. Die nächste Vorstellung ist am 17. November in München.”

Der Formant im Elektor-Besitz konnte gerade noch vor dem Müllcontainer bewahrt werden und steht seitdem im Keller, wo er immer wieder gespielt wird [3]. Kopfhörer auf!


[1] http://youtu.be/_5cRr7og4Vc

[2] <http://youtu.be/ynQyGT6t9tU>

[3] <http://youtu.be/JOhxLE3-ecl>

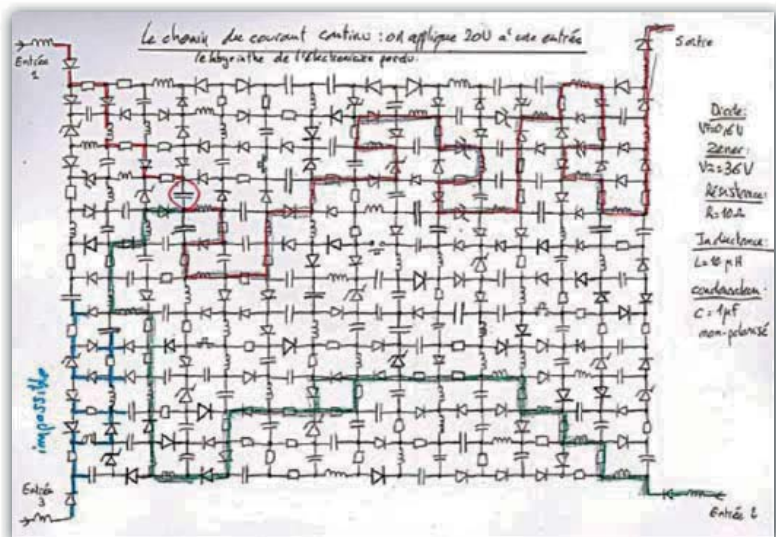
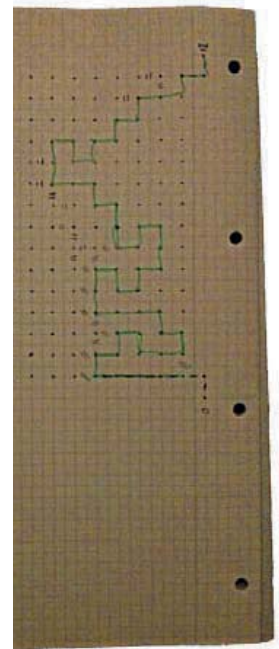
Verloren im Labyrinth

Leider hatte das in dieser Sommerausgabe abgedruckte Bauteile-Labyrinth (7-8/2012, S.60) einen kleinen Schönheitsfehler, worauf sich auch gleich mehrere Leser bei uns meldeten. Stellvertretend hier die E-Mail von Peter Wendt und eine Antwort des Rätsel-Entwicklers Sadettin Commert:

 Hallo liebe Leser,

leider war die Version des Labyrinths, die in Elektor abgedruckt wurde, nicht die endgültige, sie enthielt noch einen Fehler. Anbei ist die richtige Version, hier wurde noch ein Widerstand durch einen Kondensator ausgetauscht. Die fehlerhafte Version erlaubte zwei mögliche Lösungen.

Sadettin Commert (Frankreich)



Elektor live! - Gelungene Premiere



Ende Oktober fand der erste Seminar- und Ausstellungstag „Elektor live!“ in Deutschland statt. Im Dortmunder Kongresszentrum Westfalenhallen kamen rund 400 Elektronikfans zusammen, um die eigenen Kenntnisse zu erweitern, sich über neue Produkte zu informieren und mit Gleichgesinnten zu fachsimpeln. Eines der Highlights waren die Seminare zum Thema „Android“ - hier kamen Einsteiger genauso wie Fortgeschrittene auf ihre Kosten. Der Vortrag zur Produkt-Entwicklung mit Embedded Linux sprach die Profis unter den Besuchern an, woraus sich interessante Fachgespräche entwickelten. Zum bunten Seminarprogramm gehörten aber auch FPGAs, Röhrenverstärker, Xmega-Controller, Transistor-Grundlagen und vieles mehr.

In der Mittagspause zwischen den Seminaren kam ebenfalls keine Langeweile auf: David Cuartielles, einer der beiden Gründer des Arduino-Projekts, fand Zeit für ein kleines Interview und eine unterhaltsame Präsentation auf der Bühne. Dazu gab es interessante Dinge an den Ständen der Aussteller zu sehen. Ins Kongresszentrum gekommen waren auch Firmen wie der bekannte Distributor Reichelt, die man nur selten auf Messen findet. Und fast alle Aussteller hatten nicht nur neue Produkte mitgebracht, sondern zeigten die Boards und Tools auch in Aktion. Dicht umlagert war der Stand des Startup-Unternehmens TinkerForge: Die Besucher waren begeistert, wie leicht sich Elektronik vom PC aus steuern lässt. Auch Fräsmaschinen, Wheelies, Löt-Tools und vieles mehr konnte man live erleben und ausprobieren. Der Tool-Hersteller Arcoss hatte sogar einen kleinen Wettbewerb organisiert; die Elektor-Leser konnten hier ihre eigenen Löt- und Entlöt-Fähigkeiten testen. Und auch Elektor selbst war natürlich nicht mit leeren Händen nach Dortmund gekommen. Auf großes Interesse stieß unter anderem der E-smog-Tester Tapir, der Strahlungsmesser und das Embedded-Linux-Board.

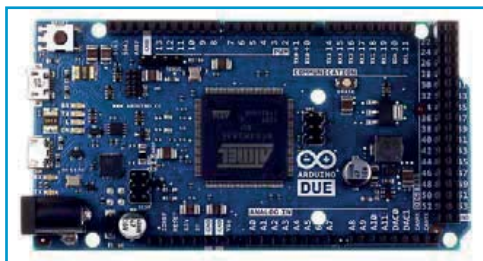
Retronik-Fans erfreuten sich am „Junior-Computer“ und dem Elektor Selbstbau-Oszilloskop. Untermauert wurde das Ganze vom ganz besonderen elektronischen Sound des Künstlers Piet-Jan Blauw. Wer wollte, durfte auch einmal selbst mit den interessanten Instrumenten Musik machen.

Insgesamt ein sehr unterhaltsamer Tag und alle Anwesenden waren sich einig: Im nächsten Jahr kommen wir wieder!

Arduino-Board mit ARM-Controller

Nach einer einjährigen Ankündigungsphase ist es nun da: Das Board „Arduino Due“. Das Controller-Board ist mit einem Cortex-M3-Controller vom Typ SAM3X8E ausgestattet, der mit 84 MHz getaktet ist, über 96 KB SRAM verfügt und von 512 KB Flash begleitet wird. Im Arduino-Store kann man das Board für 39 Euro ohne MwSt. kaufen (im Moment wird das Board allerdings als ausverkauft angegeben). Arduino Due verfügt über 54 digitale I/O-Leitungen und stellt vier Hardware-USARTs, zwei I²C-Interfaces, zwölf PWM-Ausgänge sowie zwölf 12-bittige ADC-Kanäle zur Verfügung. SPI- und JTAG-Anschlüsse sind ebenfalls vorhanden. Zwei 12-bittige analoge Ausgänge bieten per Audio-Library eine Soundausgabe. Der Controller bietet sogar ein CAN-Interface an. Hinzu kommen zwei USB-Ports, wovon der Programming-Port zur Programmierung dient. Ein Hardware-USART ist über einen ATmega16U2-Chip mit diesem USB-Ausgang verbunden, so dass man vom PC aus einfach mit dem Haupt-Controller kommunizieren kann. Der zweite USB-Port ist als USB-On-The-Go-Schnittstelle ausgeführt. Vor dem Aufstecken von Shields sollte man überprüfen, ob diese mit der für den Due geeigneten Betriebsspannung von 3,3 V kompatibel sind. Man darf auch keine 5 V an I/O-Leitungen anlegen. Bei neuen Shields kann man anhand des neuen Pins IOREF registrieren, mit welcher Spannung der jeweilige Controller arbeitet. Das Board läuft mit einer Versorgungsspannung von 7 bis 12 V und nimmt im Betrieb etwa 800 mA Strom auf.

<http://arduino.cc/en/Main/ArduinoBoardDue>



RFID-Chips in Platinen

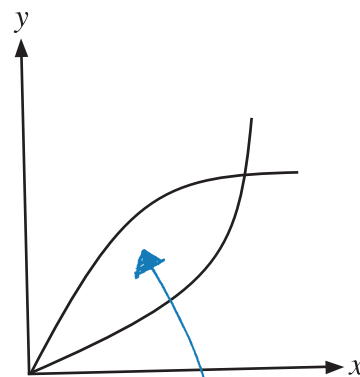
Der Platinendienstleister Beta Layout hat seine neueste Innovation „Magic PCB“ auf der diesjährigen Electronica in München vorgestellt. Die Neuheit besteht darin, dass es Beta Layout gelungen ist, Platinen mit RFID-Chips auszustatten. Die Chips werden schon bei den ersten Produktionsschritten in die Platinen integriert, sodass sie im weiteren Prozess jederzeit erkannt und verfolgt werden können. Der Vorteil von integrierten RFID-Chips liegt darin, dass sie eine sehr hohe Auslesegenauigkeit aufweisen und gleichzeitig eine Art unsichtbarer Kopierschutz darstellen. Die verwendeten RFID-Chips arbeiten im UHF-Band zwischen 860 und 960 MHz, das weltweit verwendet wird. Ein zweites Highlight am Stand von Beta Layout war das neue IDF-to-3D-Tool (IDF = Intermediate Data Format). Wenn die für das Platinen-Layout verwendete Software IDF-Daten ausgeben kann, erlaubt dieses Tool die Generierung von 3D-Modellen der bestückten Platinen. Die 3D-Ansicht ermöglicht, vorab die Ausmaße und Position von Bauteilen zu überprüfen, was gerade für Tastenfelder, Buchsen etc. von Vorteil ist. Man kann so Probleme noch rechtzeitig beheben, bevor die Platinen in Produktion gehen. Speziell für Eagle-User wurde ein ULP (User Language Program) entwickelt, das die nötigen Daten für das IDF-to-3D-Tool generiert.



www.betalayout.com

Calculus Quiz

2. Find the area enclosed by $y = \sqrt{x}$ and $y = x^2$ (4 points)



There it is
+2
Not exactly!

If only RF could be so easy.

Linx
TECHNOLOGIES

Wireless made simple®

RF Modules
Remote Controls
Antennas
RF Connectors
Custom Designs

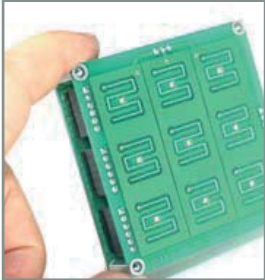
www.linxtechnologies.com

Von Clemens Valens
(Elektor .LABS)

Erstaunliches

Als ich noch Schüler war, bemerkten meine Lehrer gerne, wie schwierig es sei, mit den Fortschritten der Technik Schritt zu halten. Ich fragte mich „Wirklich?“, da es mir nicht schwer fiel, mir alle möglichen Informationen reinzuziehen, die ich finden konnte. Und nun, da meine letzten Haare auf dem Kopf damit drohen, dass sie ausgehen wollen, muss ich auch zugeben: Es ist nicht einfach zu wissen, was sich in jedem Winkel des Technikuniversums tut. Damit bin ich wohl nicht allein...

...letztlich saß ich bei einem älteren Kollegen im Auto. Er fuhr und daher konnte ich mich auf die Verkehrsschilder konzentrieren. Als wir schon nahe am Ziel waren, holte ich mein Smartphone raus und startete Google Maps. Da das Telefon schlief, musste ich es aufwecken und meinen Finger hierfür in einer bestimmten Weise über den Bildschirm führen. Dies nennt man Pattern Lock. Mein Kollege sah das und war so fasziniert davon, dass er fast auf das Auto vor uns auffuhr. Zurück auf der Straße fragte er, ob das wohl das Neueste unter den coolen Gadgets sei. Ich verneinte und erklärte, dass diese Technik schon seit ein paar Jahren benutzt wird. Im Moment, so erläuterte ich, haben wir sogar ein Projekt auf .LABS, mit dem man seinen eigenen Pattern Lock erzeugen kann. Mein Chauffeur war sehr beeindruckt. Witzigerweise erhielt ich drei Tage später eine Message von ihm, dass er das Projekt gebaut hätte und es jetzt fertig sei. Er hätte sogar ein Video davon gemacht, teilte er mit. Also schaute ich bei .LABS rein und dieses Mal war es ich, der tief beeindruckt war! Eine saubere Beschreibung, hochqualitative Fotos und ein perfektes Video – genau so, wie wir das lieben. Seine Präsentation führte sogar dazu, dass in mir der Wunsch entstand, auch so etwas zu bauen. Wundern Sie sich also nicht, wenn Sie bald mehr über dieses Projekt in Elektor lesen können. Es wäre natürlich super, wenn alle Projekte in .LABS so gut präsentiert würden. Doch das ist keine Pflicht. Wenn man nicht die Gerätschaften und das Know-How hat, Fotos in Studio-Qualität zu machen, dann tut es auch die Kamera im Smartphone. Und wenn Sie Ihre Schaltung nicht mit einem professionellen CAD-Programm erstellen, dann genügt auch eine Skizze auf Papier, die eingescannt und hochgeladen wird. Viel wichtiger ist nämlich, dass Sie dabei sind und sich mit anderen Elektronikern über interessante Projekte und gute Ideen austauschen. Wir würden uns freuen, Sie bei .LABS zu treffen.



Schauen Sie sich auch mal das Pattern-Lock-Projekt an:

www.elektor-projects.com/project/android-style-capacitive-sensing-pattern-lock.12407.html

Die Entstehung der Arten

Ich mag Wikipedia, Sie auch? Schauen Sie, was ich da ausgegraben habe:

„Die Staubläuse (Psocoptera) sind eine Ordnung der Insekten, bis zum Jahr 2005 wurden weltweit etwa 5600 Arten beschrieben. Etwa 100 Arten sind in Mitteleuropa verbreitet. Die Tiere ernähren sich von Pilzgewebe, Sporen, Flechten, Grünalgen oder Ähnlichem.“

...

Die meisten der heimischen Arten gehören in die Gruppe der Psocomorpha oder Echten Staubläuse. Bei ihnen bestehen die Antennen immer aus 13 Gliedern, die Tarsen können zwei- bis dreigliedrig sein. Hierher gehören etwa die nur parthenogenetisch lebende Art Psoculus neglectus, die häufig in Vogelnestern oder Wohnungen zu findende Lachesilla pedicularia oder der Bodenbewohner Kolbea quisquiliarum.“

Wenn Sie wie ich nicht viel mehr verstehen, als dass es da um Krabbeltiere geht, dann werden Sie sich sicher fragen, warum ich das hier zitiert habe. Nun: Kürzlich wurde die Unterordnung der Psocomorpha um eine neue

Spezies erweitert, die lauteste überhaupt, den PSoCaMorph-Synthesizer. Dieses Exemplar wurde zuerst auf .LABS gesichtet!

Bei diesem Projekt handelt es sich um die Neuauflage eines alten analogen Synthesizers mit Hilfe modernster PSoCs (Programmable System-on-Chip). Wie auch schon dem Wikipedia-Eintrag zu entnehmen ist, sind die Tiere dieser Ordnung äußerst modular aufgebaut: Da gibt es Flügel, Beine, Antennen und viele andere Körperteile. Analoge Synthesizer sind geradezu prototypisch modular: Sie bestehen je aus einem oder mehreren VCO-, VCF- und VCA-Modulen und einigen anderen Teilen. Chips und Bugs sind beide klein und haben mehr als zwei Beine. Psocoptera erschienen vor 250 Millionen Jahren und analoge Synthesizer sind (fast) schon ausgestorben.

Auf jeden Fall können Sie jetzt selbst zum technischen Insektenkundler werden, wenn Sie beim PSoCaMorph-Projekt auf .LABS mitmachen. Dabei haben Sie sogar die Chance, ein PSoC-5-Entwicklungs-Kit zu gewinnen!



Ein PSoCaMorph lebt hier:

www.elektor-projects.com/project/the-psocamorph.12269.html



Das gefällt der Redaktion

Einige .LABS-Projekte werden vom Elektor-Team für eine Veröffentlichung ausgewählt. Von daher bitten wir Sie, dass Sie gelegentlich Ihre E-Mails durchsehen, wenn Sie ein Projekt auf .LABS gestartet haben. Wir können es nämlich schlecht veröffentlichen, ohne uns mit Ihnen abgestimmt zu haben.

Nachfolgend eine Liste interessanter Projekte, die wir gerne in Elektor veröffentlichen wollen:

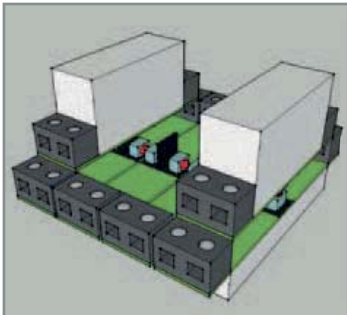
Geigerzähler-Datenlogger mit WLAN-Interface

Marcus Vohburger arbeitet an einem kompletten System für die Messung und Überwachung von Radioaktivität. Laut Marcus ist der Bau eines Geigerzählers recht einfach. Und da es so einfach ist, hat er ihn einfach erweitert ;-)

www.elektor-projects.com/project/geiger-counter-data-logger-with-wlan-interface.12264.html



elektor **e** labs



Netzschalter: Programmierbares Relais & Energiemonitor

John Hind hat dazu geschrieben: Vielleicht nicht das Neueste vom Neuesten, denn hier werden einfach fünf Schaltungen aus der Juli/August-Doppelausgabe 2012 kombiniert, um eine Art Zentrale zu bauen, die nicht nur schaltet, sondern auch misst und absichert sowie Verbrauchsdaten aufzeichnet – das Ganze zu minimalen Kosten.

www.elektor-projects.com/project/mains-gate-programmable-relay-energy-monitor.12461.html

Einfaches MIDI-Wind-Instrument

Dieses Projekt basiert auf Experimenten des „Original Posters“ (OP), die in einem überkomplexen System mündeten. Wenn dieses Projekt veröffentlicht wird, modernisieren und vereinfachen wir es zuvor. Dazu könnten wir etwas Unterstützung von Ihnen gebrauchen. Wenn jemand einen brauchbaren Wind-Messer bzw. Luftdruck-Sensor kennt...

www.elektor-projects.com/project/simple-midi-wind-instrument.12277.html



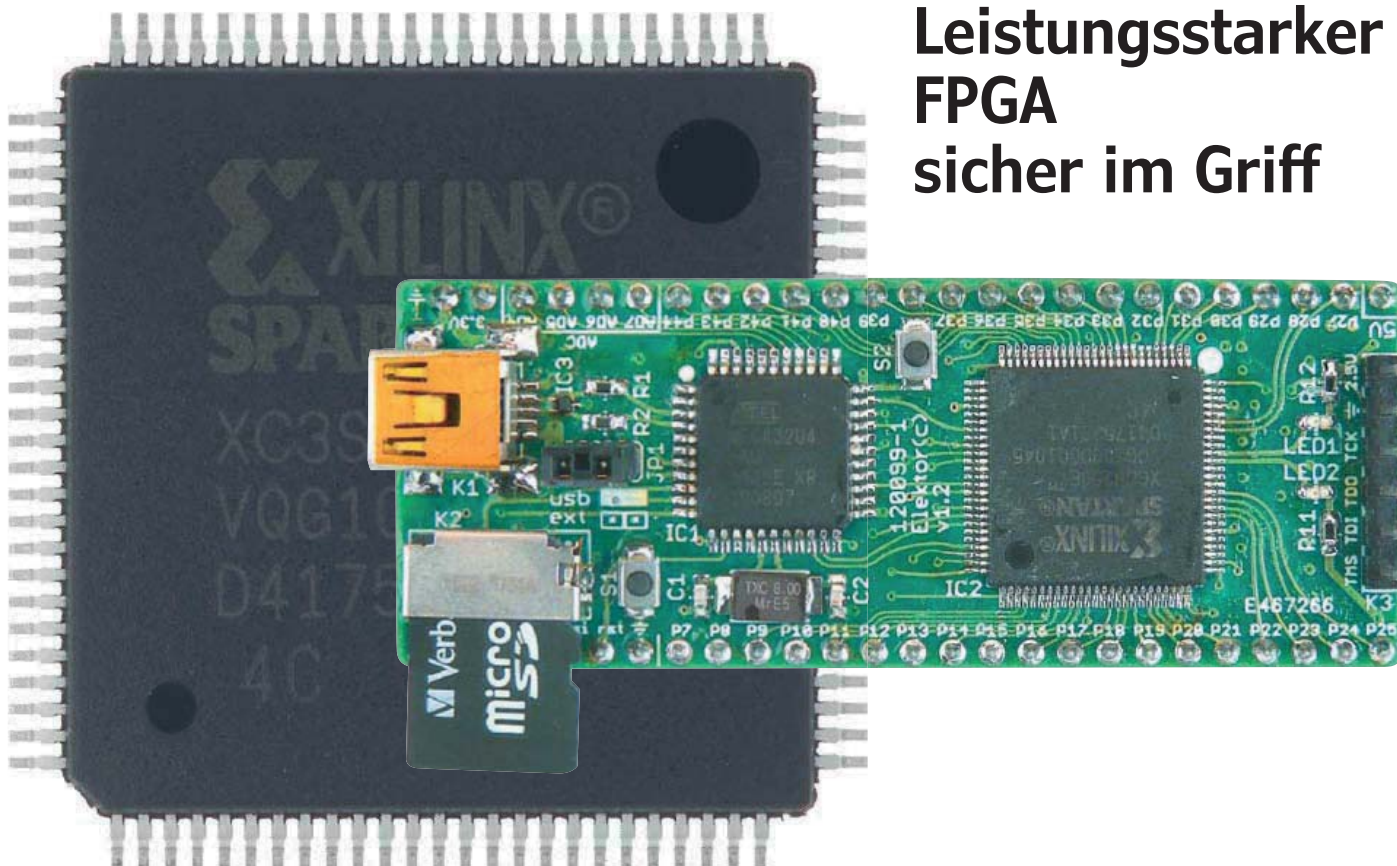
Windrichtung und -Geschwindigkeit messen

Zwischen zwei Spray-Dosen sitzt eine kleine Platine mit nur vier Bauteilen: Ein absichtlich erhitzter Transistor und drei NTCs. Wind verschiebt die Luftsäule, die in der Mitte vom Transistor erhitzt wird und die Änderung wird dann mit den drei NTCs detektiert. Nette Idee, oder?

www.elektor-projects.com/project/wind-direction-and-speed-instrument.12480.htm

Bauen Sie Ihren Chip!

**Leistungstarker
FPGA
sicher im Griff**



Text:
Clemens Valens
(Elektor-Labor)

Projekt:
Raymond Vermeulen
(Elektor-Labor)

Zu den aktiven Komponenten, die sich durch enorme Flexibilität, aber auch durch hohe Komplexität auszeichnen, zählen zweifellos die FPGAs. Das Grundkonzept erlaubt dem Entwickler eine Vorgehensweise, die dem Konstruieren maßgeschneiderter Chips gleichkommt. In dieser Artikelfolge wollen wir uns auf das Terrain der programmierbaren Logik-Bausteine begeben, die Basis ist das im Elektor-Labor entwickelte FPGA-Board. Schwellenangst war gestern, das Elektor-FPGA-Board hilft, diese Hürde zu nehmen!

Die Kurzbezeichnung FPGA steht für *Field Programmable Gate Array*, was wörtlich „Feldprogrammierbare Anordnung logischer Ports“ bedeutet. Eine freiere und vermutlich bessere Übersetzung lautet „Baustein mit digitalen logischen Funktionen, die vom Anwender konfigurierbar sind“. FPGAs vereinen heute eine Vielzahl logischer Ports, Flipflops, Speicherelemente und noch einiges mehr auf dem gleichen Chip, vom Anwender können sie ähnlich dem Baukastenprinzip zu digitalen Systemen zusammengestellt

werden. Aus dieser grundlegenden Eigenschaft resultiert der generelle Unterschied zu den Mikrocontrollern: FPGAs führen keine Programme aus, sie müssen als komplexe digitale Schaltungen betrachtet werden.

Erste Ansätze

Programmierbare Logik bestand zunächst aus einer Matrix, in der mehrere Dutzend logischer Ports versammelt waren. Die Ein- und Ausgänge konnten weitgehend frei miteinander gekoppelt

werden. Unterschiedliche logische Funktionen ließen sich mit dem gleichen Chip realisieren, die Vielzahl diskreter logischer Bausteine auf überdimensionalen Platinen war Vergangenheit. Hinzu kam, dass Funktionen durch Neuprogrammieren des Chips schnell angepasst werden konnten, ein eventuelles Durchtrennen von Leiterbahnen und Verlegen neuer Leitungen entfiel. Die Klasse der programmierbaren logischen Chips ist dicht bei den ROMs (*Read-Only Memory*) und den EPROMs (*Erasable Programmable Read-Only Memory*) angesiedelt, die verwandte Eigenschaften haben. Ein einfaches Beispiel ist ein AND-Port mit den Eingängen A und B sowie dem Ausgang Q. Zu diesem Port gehört folgende Wahrheitstabelle:

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Wenn die Eingänge A und B in A1 und A0 umbenannt werden und Ausgang Q die Bezeichnung D0 erhält, deutet dies auf zwei Adressleitungen hin, die vier Speicherplätze adressieren. Werden in die Speicherplätze die Werte 0, 0, 0 und 1 gesetzt, entsteht eine AND-Funktion. Die Speicherplätze können auch mit anderen Werten belegt werden, die Anzahl der Adressleitungen ist ebenfalls variierbar. Nach diesem Grundprinzip lassen sich auf unkomplizierte Weise auch hochkomplexe logische Funktionen zusammenbauen. Allerdings ist das Speichern eines Ergebnisses, das eine Funktion liefert, noch nicht möglich. Programmierbare Logik-Bausteine wurden deshalb durch Register ergänzt, die Ergebnisse zwischenspeichern, so dass sie von anderen Teilen des gleichen logischen Systems weiterverarbeitet werden können. Die einfachste Art eines Registers ist das Flipflop, es kann genau 1 bit speichern. In **Bild 1** ist ein solches 1-bit-Speicherelement, wie es in FPGAs vorkommt, schematisch dargestellt.

Flipflops können die Grundelemente vieler Funktionen sein, sie können beispielsweise zu Zählern oder Registern beliebiger Breite zusammengeschaltet werden. Das Verzögern von Datenflüssen ist möglich, und auch Schieberegister lassen sich realisieren, die parallele oder serielle Datenformate ineinander konvertieren.

Fazit: Eine aus logischen Ports bestehende Matrix ist der richtige Ansatz, doch erst wenn Speicherelemente hinzukommen, sind neue Ziele erreichbar.

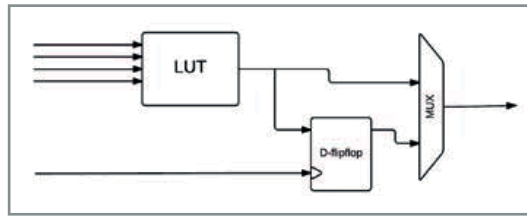


Bild 1.
Stark vereinfachte
blockschematische
Darstellung einer logischen
Zelle in einem FPGA.

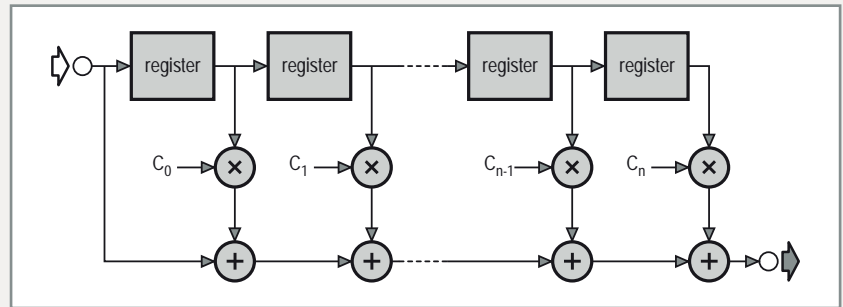
Fortentwicklung

Die Komplexität programmierbarer logischer Bausteine wuchs im Laufe der Jahre stetig. Das ist bereits an der Kurzbezeichnung CPLD einer Gruppe solcher Bausteine erkennbar, die *Complex Programmable Logic Device* bedeutet. Die fortschreitende Entwicklung machte es nicht nur möglich, sondern auch notwendig, die Bausteine so zu gestalten, dass sie eher mit den RAMs (*Random Access Memory*) als den ROMs (*Read Only Memory*) verwandt sind. Allerdings verlieren RAMs nach Abschalten der Betriebsspannung ihren Inhalt, und die neuen programmierbaren Logikbausteine verlieren ihre Konfiguration. Die Idee zur Lösung dieses Problems besteht darin, den Chip bei jedem Start innerhalb des Systems zu konfigurieren. Dies war die Geburtsstunde des FPGA. Die spezifische Konfiguration wird oft dadurch gesichert, dass die zugehörigen Daten in einem externen, nicht flüchtigen Speicher stehen und vom FPGA nach jedem Reset neu gelesen werden. Hier liegt die Assoziation zu einem Mikroprozessor nahe, dessen Programm ebenfalls in einem separaten, nicht flüchtigen Speicher untergebracht ist. Inzwischen existieren auch FPGAs, die ihre Konfiguration ohne externe Komponenten nicht-flüchtig speichern können.

Im Lauf der Jahre ist es den FPGA-Herstellern gelungen, eine enorme Anzahl Gatter und Flipflops auf einem einzigen Chip unterzubringen. Die immer noch fortschreitende Integration macht einerseits neue Anwendungen mit extrem schnellen Datenflüssen möglich, andererseits schafft sie neue Probleme, beispielsweise hinsichtlich der internen Chip-Organisation. Das wahlfreie Verknüpfen etlicher Millionen Ports erfordert eine kaum noch überschaubare Anzahl von Verbindungswegen. Die „Leitungen“ fallen lang aus, wenn Ausgänge links oben auf dem Chip mit diagonal gegenüberliegenden Eingängen gekoppelt werden müssen. Bei hohen Datengeschwindigkeiten können die Weglängen Ursache für Verzögerungen und Deformationen der Signale bis zur Unbrauchbarkeit sein. Die Synchronisation der beteiligten Ports ist nur noch mit großer Mühe möglich.

Bild 2.

Digitale Filter bestehen generell aus Verzögerern, Multiplizierern und Zählern. Verzögerer sind mit Registern realisierbar, während die Multiplizierer und Zähler aus logischen Funktionen und Registern aufgebaut werden können. Da dies alles in einem einzigen FPGA implementierbar ist, sind FPGAs für die digitale Signalverarbeitung ideal.



Die beschriebene Situation hat dazu geführt, dass sich eine allgemein gebräuchliche FPGA-Architektur herausgebildet hat. Die Anordnung der Funktionsblöcke auf dem Chip ist nicht willkürlich, sondern wohlüberlegt. Die Hersteller haben vor Augen, dass sich Standard-Produkte auf möglichst breiter Front einsetzen lassen, so dass die Stückzahlen steigen. In den Typen, die für spezielle Aufgaben konzipiert sind, findet der Anwender darüber hinaus besondere Funktionen vor. Eigentlich ist nur dies der Grund, warum kein FPGA-Typ einem anderen vollständig gleicht. Die Funktionsblöcke, die gegenwärtig in praktisch jedem FPGA implementiert sind, lassen sich wie folgt klassifizieren:

- Logik
- I/O
- Arithmetik
- Speicher
- Taktung

Außer Funktionen dieser allgemeinen Kategorien können weitere Komponenten integriert sein, zum Beispiel Prozessoren, Flash-Speicher und auch Peripherie wie Ethernet-Controller, Speichersteuerungen, A/D-Wandler oder Module für die serielle Datenkommunikation (SPI, I²C, USB, CAN oder andere). Mit diesem Funktionsumfang rückt der FPGA in die Nähe der autonomen Chipsysteme, sie werden auch „System-on-Chip“ (SoC) genannt.

Funktionsblöcke

Logik

Die logischen Blöcke machen den umfangreichsten Teil eines FPGAs aus, in der Malerei wären sie die „Leinwand“ (englisch *fabric*), auf der ein Künstler ein Gemälde malt. Komponenten von der Art der Matrix, die zu Beginn betrachtet wurde, gehören diesem Bereich an. Ein logischer Block ist eine konfigurierbare Ansammlung von Ports und Registern, wobei einer so genannten LUT (*Look-Up Table*) eine Schlüsselrolle zukommt. Die LUT entspricht exakt dem Beispiel mit dem AND-Port, nur ist sie meistens etwas größer, und sie hat mehr Eingänge. Anders betrachtet ist sie ein Register, beispielsweise mit der Breite 16 bit, mit dem logische Funktionen ausgeführt werden können. LUTs können auch als Schieberegister oder als Speicher dienen. Im zweiten Fall entsteht so genanntes „verteiltes“ RAM (*Distributed RAM*), der Begriff soll den Unterschied zu den Speicherblöcken andeuten. Ergänzt durch Multiplexer und weitere logische Komponenten beherrschen logische Blöcke auch niedere mathematische Operationen, was ihre Verwendbarkeit und Flexibilität steigert.

Da in einem FPGA sehr viele derartige Blöcke verfügbar sind, können Gruppen zu Multiplizierern oder Zählern zusammengeschaltet werden. Digitale Filter bestehen aus Speicherelementen, Multiplizierern und Zählern, mit ihnen sind auch

Tabelle 1. Funktionsblöcke des FPGA Xilinx Spartan-3E XC3S250E-4VQG100C

Ports	250000
Logische Blöcke	612
I/O-Blöcke	66, davon 7 nur als Eingang
Arithmetische Blöcke	12 Multiplier
Speicherblöcke	12, insgesamt 221184 bit (216 Kbit)
Verteilter Speicher	39168 bit (38,25 Kbit)
Taktblöcke	4

Die Anzahl der I/O-Blöcke hängt von der Gehäusevariante ab, die Angabe bezieht sich auf die Ausführung VQ100 mit 100 Anschlüssen. Ein logischer Block, bei Xilinx *Configurable Logic Block* (CLB) genannt, besteht aus vier *Slices* (Segmente), die jeweils zwei 16-bit-LUTs einschließen. Die Hälfte ist als verteiltes RAM nutzbar, die andere Hälfte für logische Funktionen. Zu jedem *Slice* gehören zwei Flipflops, so dass jeder CLB insgesamt über acht Flipflops verfügt (siehe Bild 3).

digitale Fourier-Transformationen ausführbar (**Bild 2**). Logische Blöcke stellen folglich alle Komponenten bereit, um digitale Signale zu filtern oder zu transformieren. Allerdings werden immer mehrere Blöcke benötigt, ein einzelner Block genügt für solche Prozesse nicht. Die digitale Signalverarbeitung gehört zu den starken Domänen der FPGAs. Wenn ein FPGA zwischen einen A/D-Wandler und einen D/A-Wandler geschaltet wird, kann dies zu überraschenden Ergebnissen führen, beispielsweise wenn das Eingangssignal ein Video- oder Audio-Stream ist.

Aktuelle FPGAs vereinen mehr als eine Million logischer Blöcke auf ihrem Chip, zum Beispiel sind im Virtex-7 XC7V2000T von Xilinx 1,2 Millionen LUTs und 2,4 Millionen Flipflops integriert. Dieser FPGA-Chip besteht aus fast sieben Milliarden Transistoren, während es der 10-Kern-Mikroprozessor Xeon Westmere-EX von Intel auf „nur“ 2,6 Milliarden Transistoren bringt.

I/O

Die logischen Blöcke müssen ebenso wie die übrigen Funktionsblöcke mit der FPGA-Umgebung über die Chip-Anschlüsse kommunizieren.

Die Brücke zur Außenwelt wird von den so genannten I/O-Blöcken geschlagen. Im erwähnten FPGA-Typ XC7V2000T von Xilinx sind 1200 I/O-Blöcke integriert.

Ein I/O-Block in einem FPGA besteht aus einer Vielzahl von Transistoren in einer Push-Pull-Konfiguration. Die I/O-Blöcke unterstützen nicht nur diverse Schnittstellen-Standards, zum Beispiel DDR, PCI, LVTTTL, LVCMOS oder LVDS, sie können Datenströme mit Höchstgeschwindigkeiten sowohl unsymmetrisch (*single-ended*) als auch symmetrisch (*differential*) übertragen. Um ihre Aufgaben zu erfüllen, machen sie Gebrauch von Verzögerungsleitungen, Synchronisationsflipflops und auch Parallel-Seriell-Wandlern (*Serializer-Deserializer, SerDes*). Selbstverständlich sind Pullup- und Pull-down-Widerstände ebenso wie Leitungsabschlusswiderstände auf dem Chip integriert.

I/O-Blöcke sind in Banken organisiert, die sich an unterschiedliche Betriebsspannungen anpassen lassen. Dadurch können die Banken gleichzeitig mehrere Schnittstellenstandards unterstützen, auch dies trägt zur Flexibilität der FPGAs bei. Die Leistungsaufnahme und die Störemission

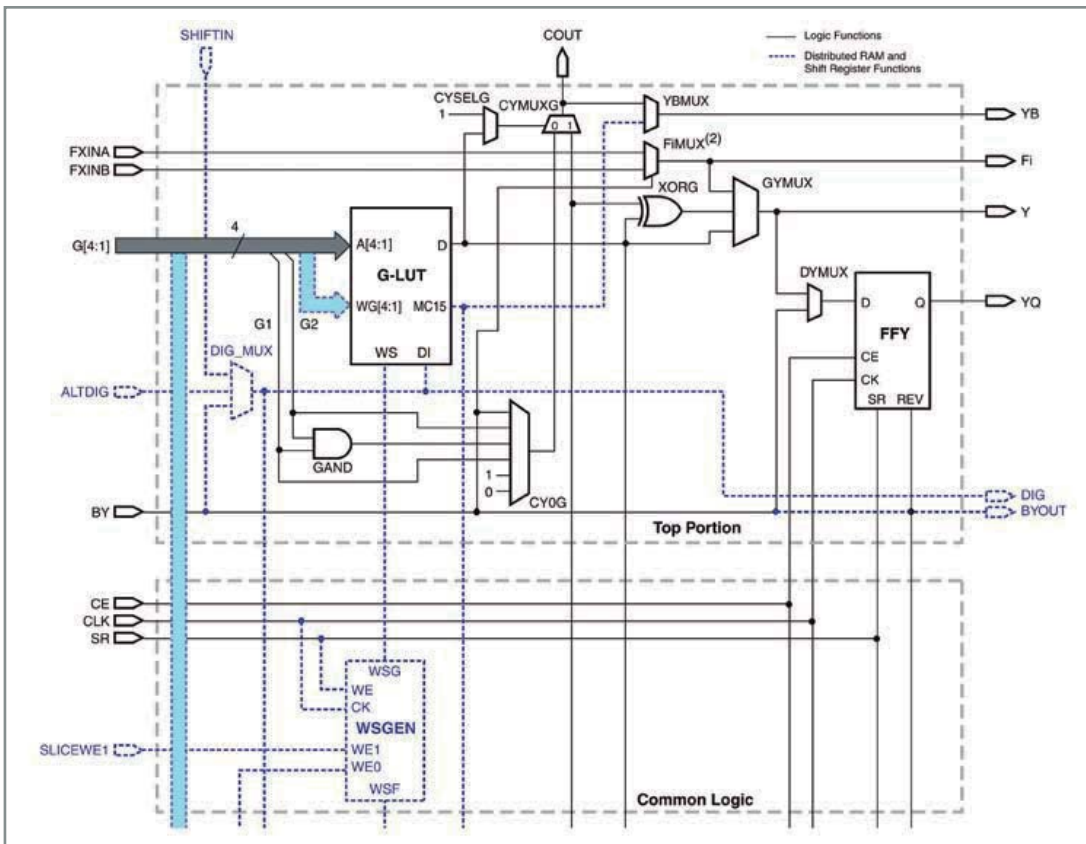


Bild 3.
Funktionsschema eines Abschnitts aus einem Slice. Der fehlende Teil ist die „Bottom Portion“, sie ist weitgehend mit der umrahmten „Top Portion“ identisch.

FPGA-Board

Aus dem Blockschema des FPGA-Boards in **Bild 4** geht hervor, dass ein ATmega32U4 und eine Micro-SD-Karte zum System gehören. Die vollständige Schaltung ist in **Bild 5** wiedergegeben. Der FPGA ist der XC3S250E-4VQG100C aus der Familie „Spartan-3E“ von Xilinx. Auf dem Chip befinden sich 250000 *System Gates*, 5508 *Logic Cells* und 612 *Logic Units*, die maximale Taktfrequenz beträgt 572 MHz. Weitere Eckdaten dieses Typs sind in **Tabelle 1** zusammengefasst. Um die Platinenfläche klein zu halten, haben wir eine FPGA-Variante mit geringen Abmessungen gewählt. Die Bauform 100-VQFP hat 100 Anschlüsse für Oberflächenmontage (SMD), davon gehören 66 Anschlüsse zu Eingängen oder Ausgängen. Das dürfte auch für die Realisierung komplexer Systeme genügen.

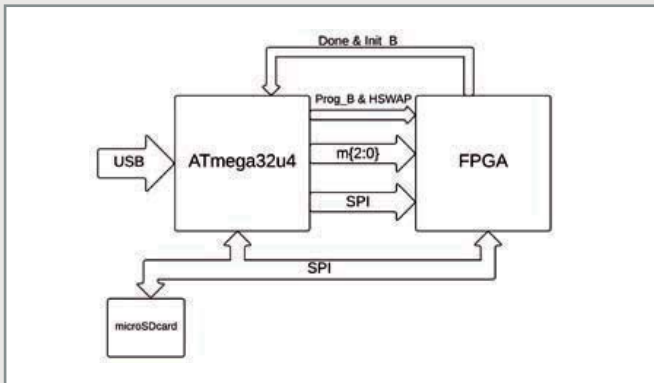


Bild 4.

Bausteine des Elektor-FPGA-Boards: Auf dem Board befinden sich außer dem FPGA ein Mikrocontroller und eine Steckfassung für eine Micro-SD-Karte.

Der Mikrocontroller ATmega32U4 übernimmt die Rolle eines Vermittlers. Er ist für den Datenaustausch mit dem PC zuständig und sorgt dafür, dass der PC mit der Micro-SD-Karte wie mit einem Massenspeicher (*Mass Storage Device*) umgehen kann. Seine wichtigste Aufgabe ist jedoch, die FPGA-Konfigurationsdatei von der Micro-SD-Karte zu lesen und in den FPGA zu laden. Dieser Mikrocontroller kombiniert die bekannten ATmega-Eigenschaften mit der Funktionalität einer USB-Schnittstelle. Als 8-bit-Typ ist er zwar vergleichsweise langsam, doch genügt er für seinen Einsatzzweck voll. Die Firmware steht auf der Elektor-Projektseite [1] zum Download bereit. Der ATmega auf der bestückten und getesteten FPGA-Platine, die der Elektor-Shop liefert, ist bereits programmiert. Die Fassung der Micro-SD-Speicherkarte belegt auf der Pla-

tine nur wenig Fläche. Die Karte arbeitet im SPI-Modus, von den meisten Typen wird dieser Modus unterstützt. Die nicht benutzten Leitungen liegen über Pullup-Widerstände an +3,3 V, so dass keine undefinierten Signalzustände auftreten können. Ein Fassungskontakt erkennt, ob eine Karte eingesteckt ist. Die Spannungsversorgung ist für eine Platine dieses Formats vergleichsweise aufwändig. Der FPGA benötigt die drei Betriebsspannungen 3,3 V, 2,5 V und 1,2 V, die Spannung 3,3 V versorgt auch die Micro-SD-Karte und den Mikrocontroller. Die Entkopplungskondensatoren tragen maßgeblich zur Systemstabilität bei. Wenn Jumper JP1 an seinem Platz ist, wird das FPGA-Board über die USB-Leitung mit Strom versorgt. Die USB-Spannung +5 V liegt auch an Kontakt P26 des Steckverbinders K5. Beim Umgang mit dieser Spannung ist Vorsicht geboten: Die I/O-Anschlüsse des FPGA vertragen +5 V nicht! Wenn JP1 offen ist, kann das Board über P26 von K5 mit +5 V aus einem externen Netzteil versorgt werden. Eine USB-Verbindung ist dann nicht zwingend notwendig.

Wegen der zahlreichen Leitungswege auf enger Fläche hat das Elektor-Labor eine vierlagige Platine entwickelt (**Bild 6**). Die Platine muss nicht in Eigenregie angefertigt und bestückt werden, denn der Elektor-Shop (siehe [1]) liefert das fertig montierte und getestete FPGA-Board zu einem attraktiven Preis.

Konfiguration laden

Normalerweise können FPGAs nicht dauerhaft konfiguriert werden, dies muss nach jedem Reset neu geschehen. Die meisten FPGA-Entwicklungsboards benutzen hierzu einen PC und einen JTAG-Programmer, dies ist hier über Anschluss K3 ebenfalls möglich. Systeme mit FPGAs verwenden zum Speichern der Konfiguration oft Flash-Speicher, der mit dem SPI-Bus verbunden ist. In der Startphase liest der FPGA die Konfiguration aus dem Flash-Speicher. Auf dem Elektor-FPGA-Board setzt der Mikrocontroller die Konfiguration nach dem SPI-Protokoll in den FPGA, dieser Modus wird auch *Serial-Slave-Mode* genannt. Die Methode wird überwiegend in größeren Systemen mit mehreren FPGAs praktiziert, bei nur einem FPGA ist sie ebenfalls anwendbar.

Auch mit der Micro-SD-Karte kann der Mikrocontroller über SPI kommunizieren. Damit die Karte unter einem PC-Betriebssystem einsatzfähig ist, muss sie für die Dateisysteme FAT16 oder FAT32 formatiert werden. Der Mikrocontroller muss in der Lage sein, mit dem Dateisystem umzugehen, damit er die Karte lesen kann. Da die Karte auch über das PC-Betriebssystem zugänglich sein soll, muss der Mikrocontroller die so genannten

SCSI-Kommandos interpretieren und ausführen. Das gewählte Systemkonzept macht den Arbeitsablauf einfach. Nachdem in dem Tool Xilinx ISE die Konfigurationsdatei erstellt wurde, wird sie im Windows Explorer in „config.bin“ umbenannt und auf die Micro-SD-Karte übertragen. Im Explorer erscheint das FPGA-Board als USB-Massenspeicher mit zugewiesenem Laufwerksbuchstaben. Nachdem die Datei übertragen ist, wird das „USB-Massenspeichergerät“ (die FPGA-Karte) vom Betriebssystem abgemeldet. Der Mikrocontroller lädt die Konfiguration in den FPGA, sobald das FPGA-Board neu startet.

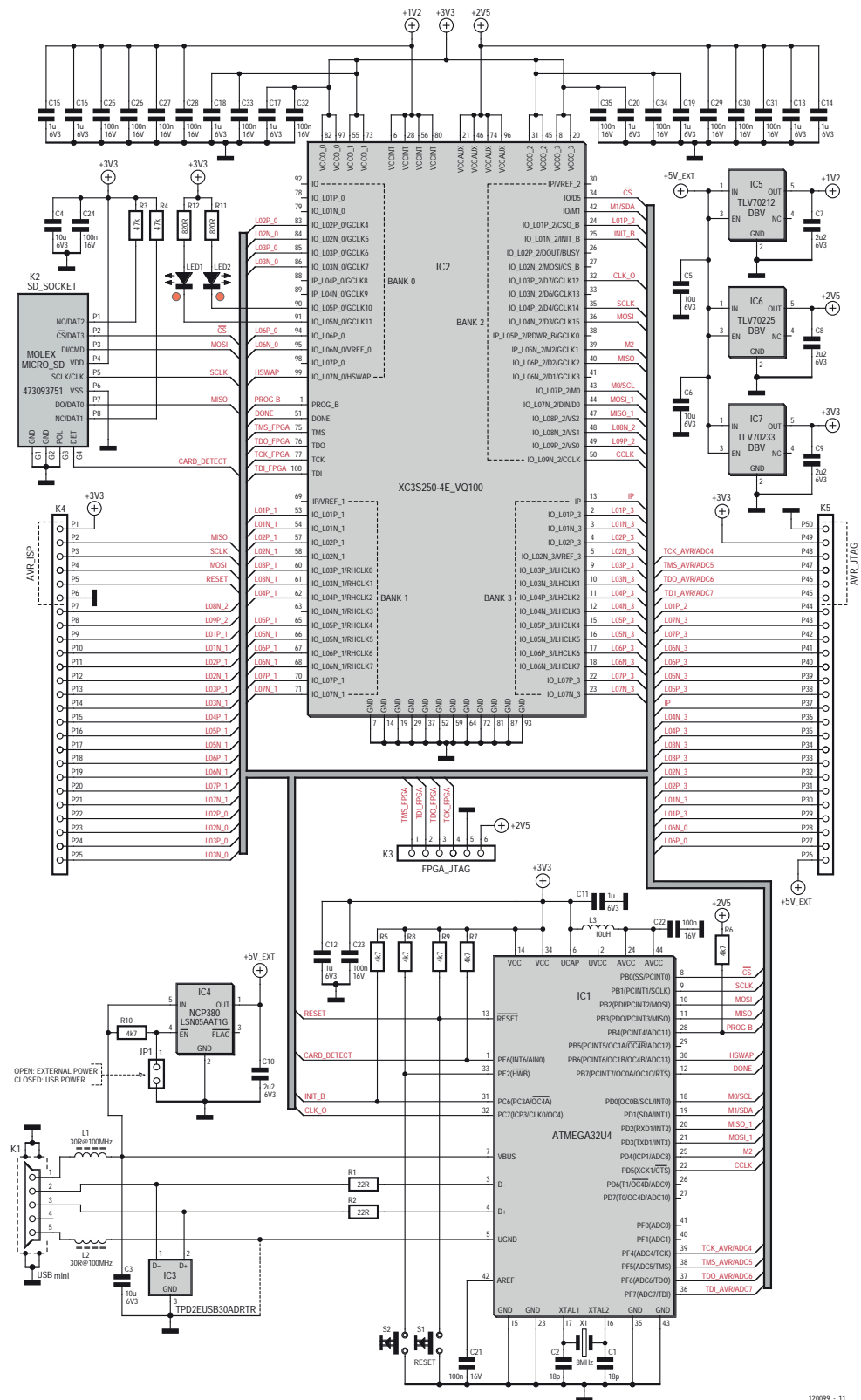
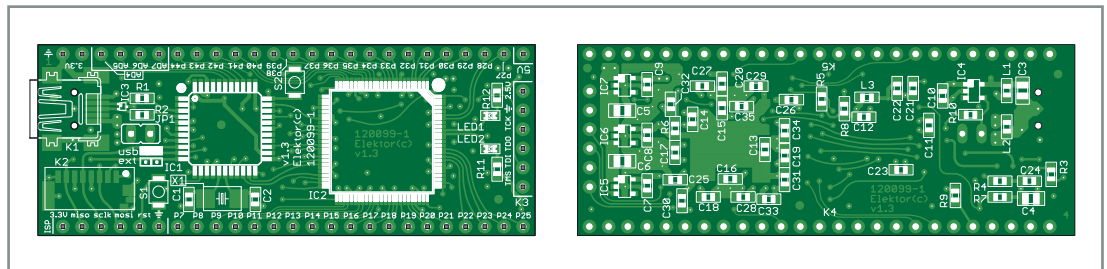


Bild 5.
Die Schaltung des Elektor-FPGA-Boards passt gerade auf eine Druckseite, die zugehörige Platine ist nur 65 mm mal 25 mm groß.

Bild 6.
Die vielen Leitungen lassen sich bei der kleinen Fläche nur mit einer vierlagigen Platine realisieren. Die Platine ist aufgebaut und getestet lieferbar.



sind vom Anwender beeinflussbar, indem er die Flankensteilheit der Signale und den Strom der Ausgangsleitungen konfiguriert.

Arithmetik

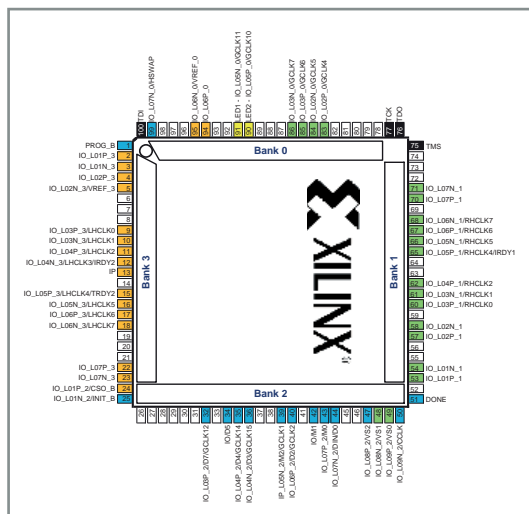
Da FPGA-Anwender logische Funktionsblöcke häufig für arithmetische Operationen einsetzen, haben die FPGA-Hersteller in viele Typen vollständige arithmetische Einheiten implementiert, die

bequemer konfigurierbar sind. Meistens handelt es sich um Multiplizierer, manchmal sind auch DSP-Funktionen vorhanden, die oft vorkommende Operationen wie $a + (b \cdot c) \rightarrow a$ (*multiply-accumulate*) beherrschen.

Speicher

Außer dem verteilten RAM in Form von LUTs sind in FPGAs auch RAM-Sektionen integriert, deren Gesamtgröße im Megabit-Bereich liegen kann. Das Unterbringen von Speicher auf dem Chip macht extrem hohe Datengeschwindigkeiten möglich. Auch funktional vollständige Mikrocontroller-Systeme sind mit einem einzigen FPGA ohne externe Komponenten realisierbar. Wenn RAM-Funktionsblöcke mit zwei voneinander unabhängigen Ports ausgestattet sind, können sie die Funktion von Datenpuffern (FIFOs) übernehmen. Damit sind beispielsweise Systemkomponenten synchronisierbar, die mit unterschiedlichen Geschwindigkeiten arbeiten. Die Speicherwortbreite ist variabel, so dass zum Beispiel 1-bit-Speicher und 32-bit-Speicher gemischt werden können.

Bild 7.
Anschlüsse des FPGA-Typs
XC3S250E-4VQG100C von
Xilinx. Die Farben haben
folgende Bedeutungen:
Orange und Grün führen zu
den Board-Steckverbindern,
Blau ist die Schnittstelle
mit dem Mikrocontroller,
an Gelb sind zwei LEDs
angeschlossen, und Schwarz
ist die JTAG-Schnittstelle,
sie ist über einen eigenen
Steckverbinder zugänglich.



Taktung

Obwohl es theoretisch möglich ist, asynchron ablaufende Funktionen (ohne Taktsignal) zu implementieren, stellt dieser Ansatz für die meisten komplexen Aufgaben keine gangbare Lösung dar. Wegen der unterschiedlich langen Wege, die von den Signalen auf dem Chip zurückgelegt werden müssen, kommt es zu Laufzeitdifferenzen. Flanken, die an der Quelle exakt gleichzeitig auftreten, kommen an der Senke zu unterschiedlichen Zeiten an. Dadurch entstehen nicht definierte Zustände, die zu Fehlfunktionen führen. Die Lösung dieses Problems besteht darin, die Signale zu definierten Zeiten gewissermaßen abzutasten, ähnlich wie ein Foto einen visuellen Zustand zu einem definierten Moment einfriert. Diese Methode geht zwar auf Kosten der Geschwindigkeit, doch sie hält die Signalfanken synchron.

Bild 8.
Kontaktbelegungen der Board-Steckverbinder. Die orangen und grünen Pins stimmen mit Bild 7 überein. Der Mikrocontroller kann über diese Steckverbinder programmiert werden. Die Anschlüsse des AVR-JTAG-Ports sind auch als analoge Eingänge nutzbar.

1	3.3V			GND	50
2		MISO		3.3V	49
3	AVRISP	SCLK	TCK_AVR/ADC4	AVR_JTAG	48
4		MOSI	TMS_AVR/ADC5		47
5		RESET	TDO_AVR/ADC6		46
6	GND		TDI_AVR/ADC7		45
7	43	IO_L08N_2/VS1	IO_L01P_2/CS0_B	24	44
8	49	IO_L09P_2/VS0	IO_L07N_3	23	43
9	53	IO_L01P_1	IO_L07P_3	22	42
10	54	IO_L01N_1	IO_L06N_3/LHCLK7	18	41
11	57	IO_L02P_1	IO_L06P_3/LHCLK6	17	40
12	58	IO_L02N_1	IO_L05N_3/LHCLK5	16	39
13	60	IO_L03P_1/RHCLK0	IO_L05P_3/LHCLK4/TRDY2	15	38
14	61	IO_L03N_1/RHCLK1	IP	13	37
15	62	IO_L04P_1/RHCLK2	IO_L04N_3/LHCLK3/IRDY2	12	36
16	65	IO_L05P_1/RHCLK4/IRDY1	IO_L04P_3/LHCLK2	11	35
17	66	IO_L05N_1/RHCLK5	IO_L03N_3/LHCLK1	10	34
18	67	IO_L06P_1/RHCLK6	IO_L03P_3/LHCLK0	9	33
19	68	IO_L06N_1/RHCLK7	IO_L02N_3/VREF_3	5	32
20	70	IO_L07P_1	IO_L02P_3	4	31
21	71	IO_L07N_1	IO_L01N_3	3	30
22	83	IO_L02P_0/GCLK4	IO_L01P_3	2	29
23	84	IO_L02N_0/GCLK5	IO_L06N_0/VREF_0	95	28
24	85	IO_L03P_0/GCLK6	IO_L06P_0	94	27
25	86	IO_L03N_0/GCLK7		5V	26

Für die korrekte Synchronisation ist das Taktsignal zusammen mit diversen Flipflops verantwortlich. Allerdings unterliegt das Taktsignal den gleichen physikalischen Gesetzen wie andere Signale, auch hier entstehen Laufzeitdifferenzen. Die FPGA-Hersteller begegnen diesem Problem, indem sie dem Anwender konfigurierbare Verzögerungs- und Schiebemechanismen an die Hand geben. Taktsignale können auch über unabhängige interne Taktnetze transportiert werden. Ein weiterer Grund, komplexe Taktungen auf dem Chip zu integrieren, sind unterschiedlich schnelle Subsysteme (so genannte Taktdomänen). Die Taktung ist konfigurierbar, indem Taktsignale mit unterschiedlichen Frequenzen und Phasen von einem Generaltakt abgeleitet werden. Die Taktsignal-Frequenzen können bis in den GHz-Bereich reichen.

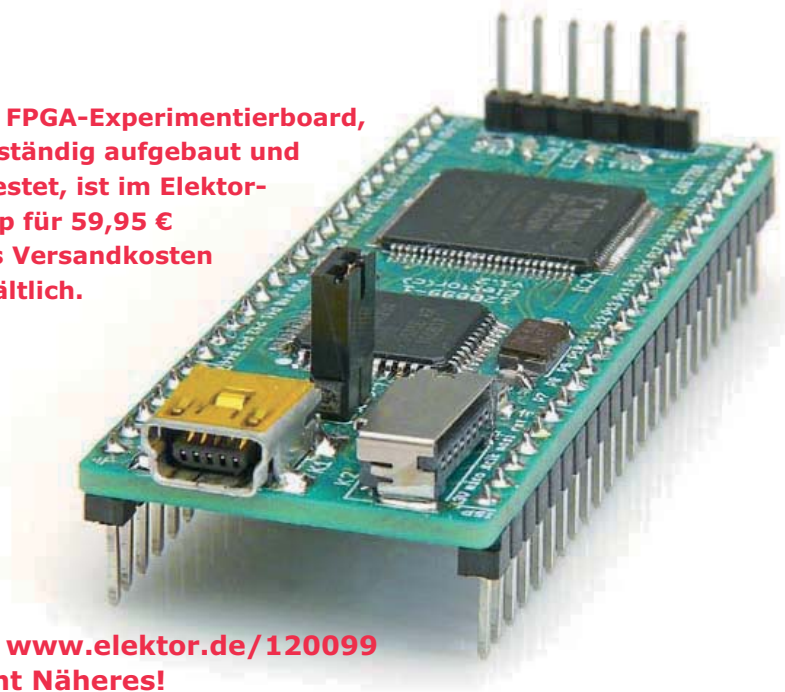
Die SD-Karte

Nach den allgemeinen Betrachtungen wollen wir uns an die ersten praktischen Versuche mit einem FPGA wagen. Zu diesem Zweck hat unser Entwickler Raymond Vermeulen ein FPGA-Experimentierboard entworfen, das einen schwerwiegenden Nachteil vieler anderer FPGA-Boards geschickt umgeht. Gemeint ist das oft recht komplizierte Verfahren, die entwickelte Konfiguration in den FPGA zu laden und zu erhalten. Ein typisches Merkmal vieler FPGA-Software-Tools ist ihre hohe, stark einarbeitungsbedürftige Komplexität. Hinzu kommt, dass zum Laden der Konfigurationsdatei ein externes JTAG-Programmiergerät notwendig ist. Unsere alternative Methode zielt darauf ab, dass ein PC das FPGA-Board wie einen intelligenten USB-Stick einstuft. Physisch ist der USB-Stick eine auf dem FPGA-Board installierte Micro-SD-Karte. Auf die SD-Karte kann die Konfigurationsdatei wie jede andere Datei geschrieben werden. Immer wenn das System gestartet wird, kopiert der FPGA die Konfigurationsdatei von der SD-Karte in seinen flüchtigen Konfigurationsspeicher. Damit ist gleichzeitig das Problem der Treiber für spezielle Programmier-Hardware gelöst.

Im nächsten Monat

Das Konfigurieren eines FPGAs ist keine ganz einfache Aufgabe. Deshalb werden wir uns in Folgebeiträgen mit den Schritten beschäftigen, die zu praktikablen realen Anwendungen führen. Voraussichtlich im nächsten Monat zeigen wir, wie das Gratis-Tool Xilinx ISE Webpack installiert wird. Als ersten Gehversuch lassen wir die

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



Auf www.elektor.de/120099 steht Näheres!

LEDs des FPGA-Board blinken. Die Zeit bis dahin können Sie zum Download der Xilinx-Software [2] nutzen. Die kostenlose Webpack-Version hat einen Umfang von etwas mehr als 6 GB. Alternativ können Sie die Software auf DVD (**Bild 9**) bei Xilinx kostenlos bestellen. In der Regel kommt sie innerhalb von zwei Wochen ins Haus.

(120099)gd

Weblinks

- [1] www.elektor.de/120099
- [2] www.xilinx.com/support/download/index.htm

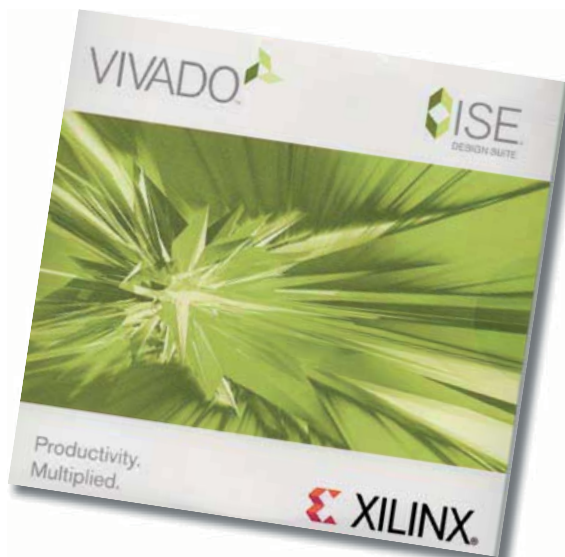


Bild 9.
Die kostenlose DVD von Xilinx, auf der sich das FPGA-Tool „ISE Webpack“ sowie weitere FPGA-Tools befinden.

USB-IO24-Kabel

Einfach messen, steuern und regeln mit dem PC

Von Dr.-Ing. Uwe Altenburg (D)

Mit dem hier vorgestellten Kabel lassen sich bis zu 24 digitale Signale einfach über die USB-Schnittstelle eines PCs einlesen und ausgeben. Wahlweise kann man auch bis zu acht analoge Signale erfassen, PWM- und Servo-Signale erzeugen und mehr. Im Artikel stellen wir ein einfaches Protokoll zur Ansteuerung des Kabels vor und gehen auch auf die PC-Programmierung in der Sprache C# ein.

Heutzutage gibt es Computer in jeder erdenklichen Form – sei es als Desktop, Laptop oder als Tablet. Das wirkt sich erfreulicherweise auch positiv auf die Preise aus. Wer vor einigen Jahren noch einen ausgemusterten Mini-Tower verwendet hat, um die Beleuchtung des geliebten Aquariums zu steuern, greift heutzutage vielleicht zu einem

schicken kleinen Box-PC und entwickelt sich eine schöne Oberfläche mit Touch-Bedienung.

Allerdings ist die PC-Anbindung eigener Elektronik über USB nicht ganz einfach, und zur Erzeugung von zeitkritischen Signalen ist ein PC auch nicht die erste Wahl. Darüber hinaus muss man sich Gedanken über ein Kommunikations-Protokoll machen sowie erst einmal eine geeignete Programmiersprache finden. Für all diese Herausforderungen bieten wir hier eine Lösung an!

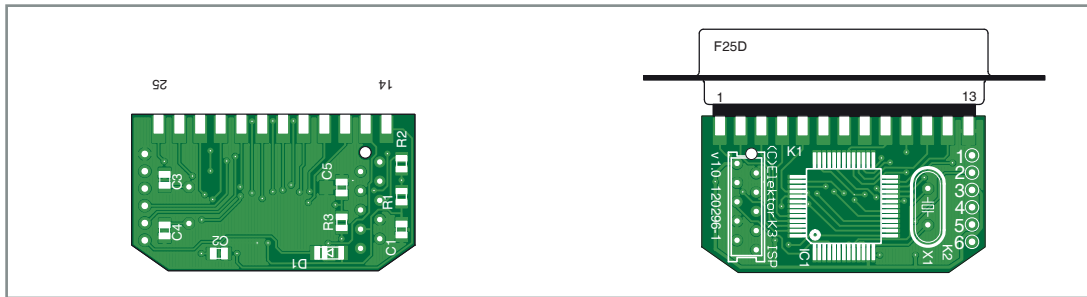


Bild 1.
Die Platine findet noch Platz
im Gehäuse der Sub-D-
Buchse.

Stückliste

Widerstände (SMD 0805, 5 %):

R1 = 100 k
R2 = 4k7
R3 = 100 Ω

Kondensatoren:

C1, C2, C5 = 100 n, 50 V, 10 %, X7R, SMD 0805
C3, C4 = 22 p, 50 V, 5 %, NP0, SMD 0805

Halbleiter:

D1 = LL4148 Minimelf
IC1 = R8C/25 SMD TQFP52

Außerdem:

USB/TTL-Adapterkabel TTL-232R-5V-WE, erhältlich
bei Elektor: 080213-71 [6]
K1 = DSub-25-Steckverbinder weiblich, für
Lötmontage (77SD B25S Amphenol)
K3 = 10-pol.-Federleisten-Steckverbinder MicroMatch,
für Lötmontage mit Anschlüssen unten (8-215079-0
TE Connectivity/Amp)
X1 = Quarz 18,432 MHz, 18 pF, 50 ppm
Haube für DSub-25-Steckverbinder, Flip Top (A-FT 25
Assmann)

Optional:

10-pol.-Stecker für Federleiste MicroMatch (8-215083-
0 TE Connectivity/Amp)

Schaltung

Von der britischen Firma FTDI werden seit einigen Jahren USB/TTL-Adapter-Kabel (TTL-232R-5V-WE [1]) angeboten, die auch bei Elektor erhältlich sind [2]. Mit Hilfe dieser Kabel, die neben den Signalen RxD, TxD, RTS und CTS auch eine 5-V-Betriebsspannung zur Verfügung stellen, wird es sehr einfach, einen Mikrocontroller an einen PC anzuschließen. Die Kabel enthalten den USB-Seriell-Wandler FT232R, der sich im vergossenen USB-Stecker befindet. Es bietet sich geradezu an, dass man am anderen Ende des Kabels eine 25-polige Sub-D-Buchse vorsieht, in deren Gehäuse man gleich einen Mikrocontroller unterbringt. Der hier verwendete 8-bit-Mikrocontroller ist ein Derivat der in Elektor schon mehrfach vorgestellten R8C-Familie. Der R8C25 [3] bietet mit 64 kB Flash und 3,5 kB RAM einen ausreichend großen Speicher. Mit seinem 10-bit-Analogwandler sowie einer leistungsfähigen Timer-Struktur ist er ideal für Steuerungsaufgaben geeignet. Die 25 Anschlüsse der Sub-D-Buchse sind genug, um sämtliche Pins von drei Controller-I/O-Ports sowie die erforderliche Masseverbindung nach außen zu führen. Im Gehäuse der Buchse ist genug Platz, um darin eine kleine Leiterplatte mit den Maßen 37x20 mm unterzubringen (**Bild 1**). Neben dem Mikrocontroller befinden sich auf der Platine nur noch ein 18,432-MHz-Quarz, ein Steckverbinder

zur Programmierung sowie einige wenige diskrete Bauelemente. Es ist geplant, die Platine fertig bestückt anzubieten, mehr darüber findet man auf der Website zu diesem Artikel [6]. An die bestückte Platine muss man nur noch ein FTDI-Kabel anlöten und das Ganze in ein passendes Gehäuse einbauen.

Schaltungstechnisch (**Bild 2**) gibt es nur wenige Besonderheiten. Der Mikrocontroller wird über das USB-Kabel versorgt. Über die Diode D1 ist die RTS-Leitung des USB-Seriell-Kabels mit dem Reset-Pin des Mikrocontrollers verbunden. Somit ist es möglich, über die USB-Schnittstelle einen Hardware-Reset des Mikrocontrollers auszulösen. Die Signale TxD und RxD des USB-Seriell-Kabels werden für die Kommunikation zwischen PC und Mikrocontroller verwendet.

Eigenschaften

- 24 digitale Ein-/Ausgänge oder alternativ bis zu
- 8 analoge Eingänge (10 bit Auflösung)
- 8 PWM-Ausgänge (10 bit Auflösung)
- 8 RC-Servo-Signale (10 bit Auflösung)
- 4 Zählereingänge (16 bit Up/Down)
- 2 ICs MAX7219 für bis zu 128 LEDs ansteuerbar

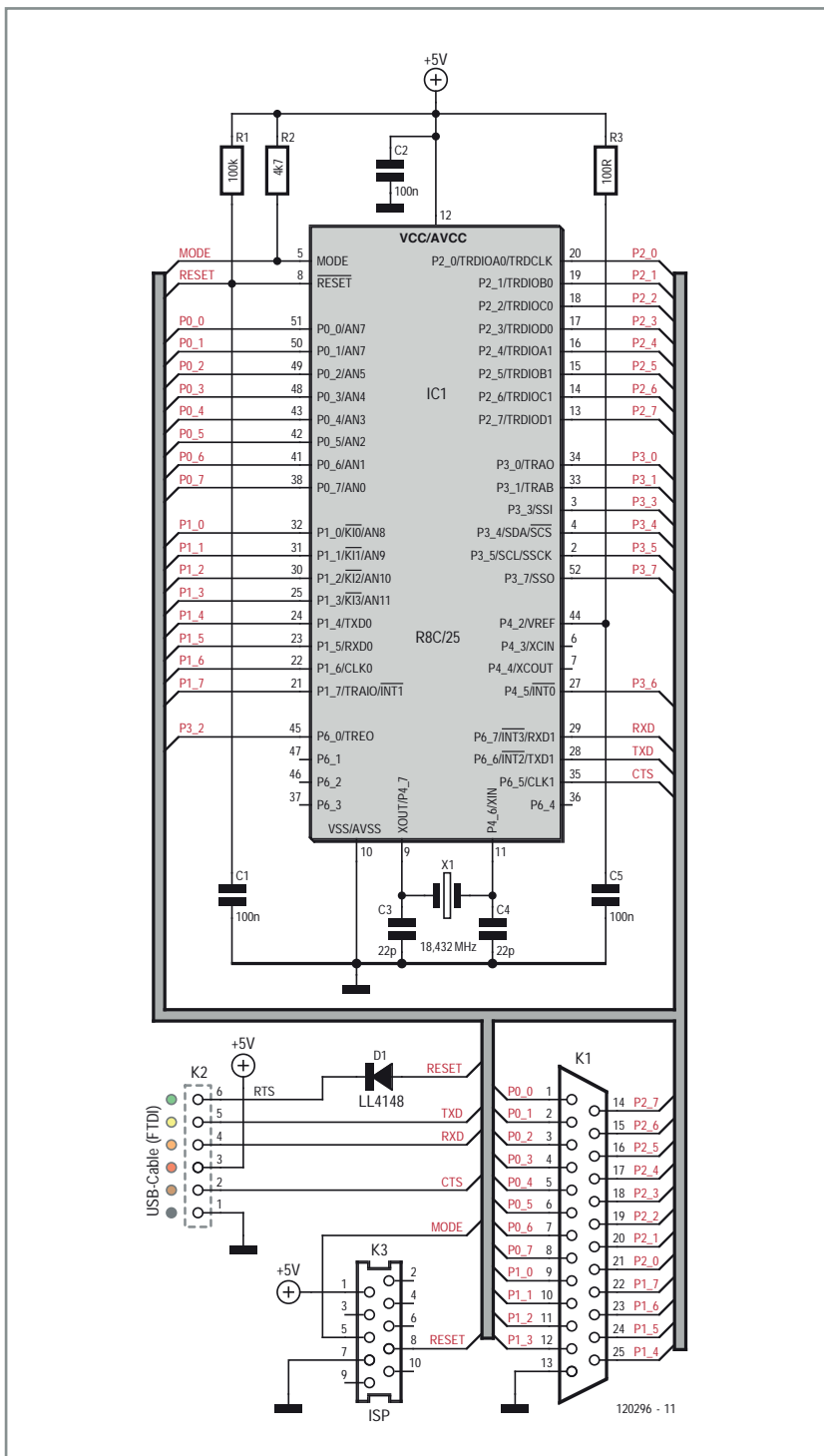


Bild 2.
Neben dem Mikrocontroller benötigt man nur noch einen Quarz sowie einige wenige diskrete Bauelemente.

Die Ports P0 bis P2 des Mikrocontrollers sind auf die Sub-D-Buchse geführt (**Bild 3**). Damit stehen alle wesentlichen Signale des Mikrocontrollers zur Verfügung.
Die Pins von Port0 können als Analogeingänge konfiguriert werden, sodass bis zu acht Analog-

signale erfasst werden können. Der Referenzspannungseingang des Analog/Digital-Wandlers ist standardmäßig über eine RC-Kombination mit der 5-V-Versorgung verbunden, was einen Messbereich von 0...5 V mit ausreichender Genauigkeit ermöglicht. Für den Fall, dass man auf eine höhere Genauigkeit bei der Messung analoger Signale Wert legt, kann aber auch eine Referenzspannungsquelle vom Typ LM4040 bestückt werden. Port1 bietet alternativ bis zu vier Interrupt-Eingänge, über die beispielsweise schnelle Zähler realisiert werden können. Zusätzlich befindet sich an Port1 eine serielle Schnittstelle, die im UART- und im SPI-Modus betrieben werden kann. An Port2 befinden sich die Ausgänge der beiden 16-bit-Timer des R8C25. Damit ist es möglich, bis zu acht PWM-Signale gleichzeitig zu erzeugen. Ein PWM-Signal kann beispielsweise für die Drehzahlsteuerung von Gleichstrommotoren oder die Generierung von Analogsignalen verwendet werden. Damit bietet unser USB-IO24-Kabel eine Vielzahl von Möglichkeiten! Wie sieht es nun aber mit der Software aus?

Firmware

Der Autor hat eine Firmware für den Controller erstellt, welche eine Ansteuerung vom PC aus sehr einfach macht (der Controller auf der bestückten Platine ist bereits mit dieser Firmware vorprogrammiert). Die Datenübertragung zwischen PC und Mikrocontroller erfolgt mit einer Datenrate von 115200 Baud bei 8 Datenbits, 1 Stoppbit und ohne Parität (115200,8,N,1). Statt eines binären Datenprotokolls mit einem festen Rahmen (der zum Beispiel Art und Länge der Daten sowie eine Prüfsumme enthält) werden zur Kommunikation einfache Textbefehle genutzt. Dieser Ansatz ist flexibel und leicht verständlich; die Kommunikation ist darüber hinaus sehr transparent. Und man benötigt zum Absetzen von Kommandos nichts weiter als ein einfaches Terminalprogramm, was das Ganze unabhängig vom verwendeten Betriebssystem macht.

Nach dem Reset beginnt der Mikrocontroller mit der Initialisierung. Viel ist dabei nicht zu tun, weil die genaue Funktion der Ports und Timer erst durch die Kommandos festgelegt wird. Lediglich die Takterzeugung und natürlich die serielle Schnittstelle für die Verbindung zum PC müssen initialisiert werden. Daneben besteht das Hauptprogramm nur noch aus einer Endlos-Schleife – dem Kommando-

Interpreter (**Listing 1**). Ein Kommando ist dabei eine Folge von ASCII-Zeichen, auf die am Schluss ein CR (Carriage Return) folgt. Genau genommen können auch mehrere Kommandos aufeinander folgen und nur das letzte von einem CR abgeschlossen werden. Das bietet mehr Flexibilität bei der Anwendung. Die Zeichenkette *X10 P0.0=1 P0.0=0 <CR>* ist dabei ein gültiges Kommando und bedeutet: „Setze 10 Mal Port0.0 auf 1 und wieder zurück“.

Die Funktion `ReadCmd()` liest eine Kommando-sequenz bis zum CR ein. Aus der Sequenz werden mit `SkipBlanks()` alle Leerzeichen entfernt und anschließend wird die Funktion `Execute()` aufgerufen. Wenn die Funktion alle Kommandos erfolgreich ausführen konnte, gibt sie einen positiven Wert zurück, anderenfalls eine Null – entsprechend wird dann *OK* oder *ERROR* zum PC zurückgesendet.

Die Ausführung einer Kommando-sequenz kann eine gewisse Zeit in Anspruch nehmen. In der Zeit dürfen aber schon weitere Zeichen über die serielle Schnittstelle vom PC aus gesendet werden. Bei der eingestellten Baudrate von 115200 Baud

Listing 1. Main-Routine mit Kommando-Interpreter.

```
// --- Main ---
VOID main()
{
    InitCrystal();           // init crystal
    InitUart();              // init uart
    EI;                      // ints

    SendString("\xCPIOCable V1.5\r\nOK\r\n");// version

    for (;;)                // endless..
    {
        ReadCmd();           // read command
        SkipBlanks();        // remove spaces

        if (Execute())       // execute commands
            SendString("\r\nOK\r\n");
        else
            SendString("\r\nERROR\r\n");
    }
}
```

dauert die Übertragung eines Zeichens gerade mal 90 µs! Der Empfang erfolgt deshalb in einer Interrupt-Routine, die die Zeichen in einem Ringpuffer ablegt. `ReadCmd()` liest die Zeichen dann aus dem Ringpuffer.

Listing 2. Interpretation des Port-Kommandos.

```
// --- Macros ---
#define Digit(p)    (*p++ - '0')           // get a digit
#define GetIf(p,c)  (*p == c ? p++, 1 : 0) // get if char

// --- Read port ---
BYTE GetPort(BYTE nPort)
{
    switch (nPort)
    {
        case 0: return p0;                // p0 input
        case 1: return p1;                // p1 input
        case 2: return p2;                // p2 input
    }
}

// --- Write port ---
VOID SetPort(BYTE nPort,BYTE nBits,BYTE nMask)
{
    BYTE nSet = nBits & nMask;

    switch (nPort)
    {
        case 0: p0 = p0 & ~nMask | nSet; break; // p0 output
        case 1: p1 = p1 & ~nMask | nSet; break; // p1 output
        case 2: p2 = p2 & ~nMask | nSet; break; // p2 output
    }
}

// --- Port command [Pn|Pn=v|Pn.b|Pn.b=v|Pn~|Pn.b~] ---
BOOL PortCmd()
{
    BYTE nPort = Digit(pCmd);              // port
```

```
BYTE nMask = 0xFF;
BYTE nPin  = 0;

if (GetIf(pCmd, '.'))                    // '.'
{
    nPin = Digit(pCmd);                  // pin
    nMask = 1 << nPin;                  // mask
}

if (nPort < 3 && nPin < 8)                // check
{
    BYTE nValue;

    if (GetIf(pCmd, '='))                // '='
    {
        nValue = GetValue() << nPin;     // value
        SetPort(nPort,nValue,nMask);    // set port
    }
    else if (GetIf(pCmd, '~'))            // '~'
    {
        nValue = GetPort(nPort);         // get port
        SetPort(nPort,~nValue,nMask);    // set invert
    }
    else
    {
        nValue = GetPort(nPort) & nMask; // get port
        nValue = nValue >> nPin;         // get value
        SendValue(nValue);              // send value
    }

    return TRUE;
}

return FALSE;
```

Tabelle 1. Kurzreferenz der unterstützten Kommandos

A..	An	$n = 0..7$	Pin An wird als Analogeingang konfiguriert und der Analogwert wird im Bereich 0..1023 zurückgegeben.
C..	$Cn, Cn = x$ $Cn+, Cn-$	$n = 0..3$ $x = 0..65535$	Der aktuelle Wert des Zählers CNTn wird zurückgegeben bzw. kann auf einen Wert im Bereich 0..65535 gesetzt werden. Mit + oder - wird die Zählrichtung festgelegt.
P..	$Pn, Pn = x$ $Pn.b, Pn.b = y$ $Pn\sim, Pn.b\sim$	$n = 0..2$ $b = 0..7$ $x = 0..255$ $y = 0..1$	Ein 8 bit breiter Port wird gelesen bzw. geschrieben (oder nur ein einzelner Pin) davon. Mit ~ wird der Port bzw. ein Pin invertiert.
S..	$Sn, Sn = x$	$n = 0..7$ $x = 0..1023$	Einer der PWM-Pins wird auf die Ausgabe eines RC-Servo-Signals programmiert. Mittelstellung des Servos ist bei $x = 512$.
W..	$Wn, Wn = x$	$n = 0..7$ $x = 0..1023$	Einer der PWM-Pins wird auf die Ausgabe eines Pulsweiten-Signals programmiert. Die Pulsbreite hat einen Bereich von 0..1023.
D..	$Dn =$ x, x, x, x, x, x, x, x	$n = 0..1$ $x = 0..255$	Ansteuerung eines MAX7219, an den man bis zu 64 LEDs anschließen kann. Es werden gleichzeitig 8 Datenbytes x gesendet.
R..	$Rn, Rn = x$	$n = 0..6$ $x = 0..255$	Lesen bzw. Schreiben der Steuerregister: R0: Datenrichtung von P0, 0=Input,1=Output R1: Datenrichtung von P1, 0=Input,1=Output R2: Datenrichtung von P2, 0=Input,1=Output R3: <i>reserviert</i> R4: PWM-Frequenz, 0=1 kHz, 1=2 kHz, 2=4 kHz R5: PWM-Verzögerung, 0=keine .. 99=langsam R6: Helligkeit 1. MAX7219, 0=dunkel .. 15=hell R7: Helligkeit 2. MAX7219, 0=dunkel .. 15=hell
T..	Tn	$n = 0..6500$	Zeitverzögerung von $n * 10 \mu s$.
X..	Xn	$n = 0..65535$	Alle Kommandos der Befehlszeile, die nach dem X folgen, werden n Mal wiederholt.
?	?		Es wird eine Kurzreferenz aller implementierten Befehle ausgegeben.

Die möglichen Kommandos werden anhand des ersten Zeichens unterschieden. So beginnt das Port-Kommando mit einem P , die Abfrage der Analogeingänge mit A , das Setzen eines Zählers mit C usw., wobei jeweils die Groß- als auch Kleinschreibung zulässig ist. Für jedes Kommando gibt es eine spezielle Funktion im Code, welche die mögliche Syntax erkennt. Im **Listing 2** sind die Funktionen für das Port-Kommando zu erkennen. Im einfachsten Fall wird der aktuelle Zustand eines Ports mit $P0$, $P1$ oder $P2$ abgefragt. Als Ergebnis sendet der Interpreter eine Zahl von 0..255 zurück. Um den Zustand eines Ports zu ändern, kann man dem Port auch einen Wert zuweisen ($P0 = 0..255$). Ob es sich um eine Zuweisung oder eine Abfrage handelt, erkennt die

Funktion `PortCmd()` also am „=“. Darüber hinaus ist es auch möglich, nur einzelne Bits eines Ports zu adressieren. Dazu wird nach der Portnummer die Pinnummer angegeben $P0.0$, $P0.1$.. $P0.7$. Auch hier erkennt die Funktion am „.“, ob der gesamte Port oder nur ein Pin gemeint ist. Schließlich kann man mit dem Kommando auch ein Portpin toggeln – das wird durch die Angabe von „~“ erreicht ($P0.0\sim$).

Befehlssatz

Mit den bereits beschriebenen Port-Kommandos lässt sich sehr gut eine digitale I/O-Karte mit Relais und Optokopplern betreiben. Allerdings muss dazu noch die Datenrichtung der einzelnen Pins des USB-IO24-Kabels konfiguriert werden. Nach einem

Reset sind alle Pins grundsätzlich auf Eingang programmiert. Dieser Umstand muss beim Entwurf eigener Hardware berücksichtigt werden – Leistungsstufen müssen dann unbedingt einen sicheren Zustand einnehmen. Mit dem Kommando *R* lassen sich verschiedene Steuerregister beschreiben. Die Register R0, R1 und R2 beinhalten die Datenrichtungen der drei Ports. Bit 0 von R0 legt die Datenrichtung des Pins P0.0 fest, Bit 1 die des Pins P0.1 und so fort. Eine 1 bedeutet Ausgang und eine 0 Eingang. Der Befehl *R0 = 15<CR>* ändert die Datenrichtung der unteren vier Pins von Port0 auf Ausgang. Alternativ kann man auch schreiben *R0 = \$0F<CR>* oder *R0 = %00001111<CR>*. Mit dem Dollarzeichen werden hexadezimale Zahlen eingeleitet und mit „%“ binäre Zahlen.

In **Tabelle 1** sind alle derzeit unterstützten Kommandos des USB-IO24-Kabels aufgeführt. Neben den bereits erwähnten Kommandos für Ports und Register gibt es weitere elementare Befehle. Für Sensor-Anwendungen besonders wichtig ist sicher das Kommando *A* zur Abfrage der Analogeingänge. Bis zu acht analoge Werte lassen sich digitalisieren. Die entsprechenden Pins AI0..AI7 müssen dazu nicht extra konfiguriert werden – sie werden beim ersten Kommando *An* automatisch auf Analogeingang programmiert. Da es prinzipiell möglich ist, mehrere Befehle auf einmal zu senden, ist es auch möglich, mehrere Analogeingänge mit einem Kommando abzufragen. Die Befehlsfolge: *A0 A1 A2<CR>* fragt die ersten drei Analogeingänge ab und sendet die Werte in einer Antwort zurück. Mehrere Werte in einer Antwort werden jeweils durch ein Leerzeichen getrennt.

Das Pendant zur Analogeingabe wäre eine Analogausgabe. Der verwendete Mikrocontroller unterstützt zwar keine direkte Analogausgabe, man kann jedoch über den Umweg der Erzeugung von PWM-Signalen bis zu acht Analogwerte generieren. Die PWM-Signale werden mit dem Kommando *Wn* eingeschaltet. Wie bei den Analogeingängen ist auch hier keine extra Konfiguration erforderlich; über das Steuerregister R4 kann man aber die PWM-Frequenz einstellen. Um aus einem PWM-Signal einen Analogwert abzuleiten, benötigt man einen Tiefpass. Im einfachsten Fall reicht dazu eine RC-Kombination; besser ist jedoch die Verwendung eines Operationsverstärkers, um einen Tiefpass höherer Ordnung aufzubauen – z.B. ein Butterworth-Filter.

Mit den bisher vorgestellten Befehlen lassen sich bereits viele Standard-Anwendungen mit digitaler oder analoger Ein- und Ausgabe abdecken. Wenn man bedenkt, dass man theoretisch bis zu 127 USB-IO24-Kabel an einen Computer anschließen kann, wären das $127 * 24 = 3048$ digitale Ein- und Ausgänge! Dennoch bedarf es mitunter spezieller Lösungen für Anforderungen, wie sie gerade im Bereich eines Hobby-Elektronikers häufig vorkommen. Genau dafür wurden weitere leistungsfähige Befehle implementiert, die nachfolgend an Beispielanwendungen erläutert werden sollen.

Befehle für RC-Servos

Die aus dem Modellbau bekannten und in allen Leistungsstufen erhältlichen RC-Servos werden inzwischen auch gern für Robotik-Projekte oder andere mechanische Antriebe verwendet. Sie regeln ihre Position entsprechend eines vorgegebenen Sollwertes. Der Sollwert für RC-Servos wird durch einen Impuls mit einer Breite von 1..2 ms codiert, der alle 20 ms wiederholt wird (**Bild 4**). Diese Impulsfolge zu erzeugen, ist eine Aufgabe, die man nicht direkt dem PC überlassen kann. Das Timing ist sehr kritisch – insbesondere bei der Impulsbreite. Schon geringes Jittern führt am Servo zu deutlich hörbarem Brummen (und damit verbunden zu einer erhöhten Leistungsaufnahme). Deshalb gibt es das spezielle Kommando *Sn*, mit dem man alternativ zu den PWM-Signalen bis zu acht Servo-Signale generieren kann. Es ist auch möglich, PWM- und Servo-Signale gleichzeitig zu erzeugen. Auf Grund der Timerstruktur im R8C25 müssen dann aber immer vier Pins von Port2 das gleiche Timing haben – also die Pins P2.0...P2.3 generieren PWM-Signale, während die Pins P2.4...P2.7 für Servo-Signale zuständig sind. Neuere RC-Servos zeichnen sich durch eine sehr hohe Stellgeschwindigkeit aus. Das ist prinzipiell ein Vorteil, wobei dann für gewollt langsame Bewegungen sehr viele Sollpositionen vorgegeben werden müssen. Dabei kommt es jedoch

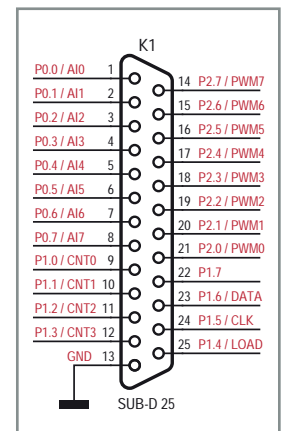


Bild 3.
Pinbelegung der Sub-D-Buchse.

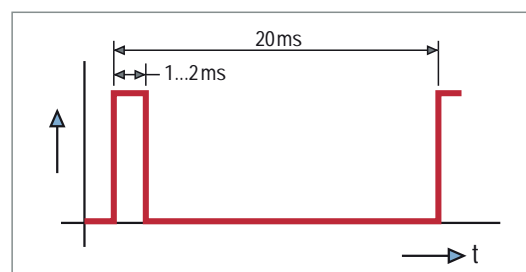


Bild 4.
Impulsfolge für RC-Servos.

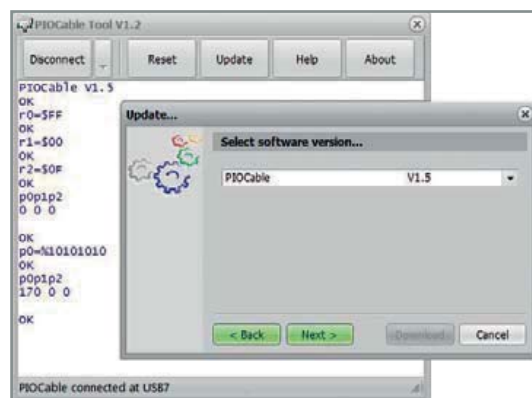
zu einer Art Ruckeln der Servos, was bei einem Roboter-Arm zu erheblichen Schwingungen führen kann. Mit dem Steuerregister R5 kann man deshalb die Stellgeschwindigkeit gezielt reduzieren. Die Verzögerung geht direkt in die Impulserzeugung mit ein und reduziert somit das Ruckeln der Servos ganz erheblich. Der Wert in R5 wirkt sich gleichzeitig auch auf die Erzeugung der PWM-Signale aus.

Position oder Drehzahl

Ein weites Gebiet ist die Ansteuerung von Gleichstrommotoren mit kleiner Leistung, wie sie in

der Achse anbringt, z.B. eine Taktscheibe, die eine Lichtschranke unterbricht. Die dabei entstehenden Impulse (Flanken) müssen dann gezählt werden. Sie ergeben akkumuliert die Position oder bezogen auf ein Zeitintervall die Drehzahl. Beides lässt sich mit dem Kommando *Cn* ermitteln. Das USB-IO24-Kabel besitzt vier Eingänge CNT0...CNT3, die controller-intern jeweils mit einem 16-bit-Zähler verbunden werden können. Der Befehl *C0+<CR>* aktiviert den Zähl Eingang CNT0, wobei der Zähler vorwärts zählen soll; mit *C0-<CR>* würde er dagegen rückwärts zählen. Der Zählerstand wird mit *C0<CR>* ausgelesen. Die Zählrichtung kann jederzeit umgeschaltet werden, wobei der aktuelle Zählerstand unverändert bleibt. Damit wird es möglich, dass der Zähler immer der aktuellen Position einer Achse folgt. Man muss lediglich vor dem Richtungswechsel des Motors die Zählrichtung umkehren. Um dagegen die Drehzahl zu ermitteln, setzt man den Zähler zunächst mit *C0=0<CR>* auf 0 und liest ihn nach einer geeigneten Zeit wieder aus. Der Zählerwert ist dann ein Maß für die Drehzahl des Motors.

Bild 5.
Die PC-Software „PIOCable-Tool“.

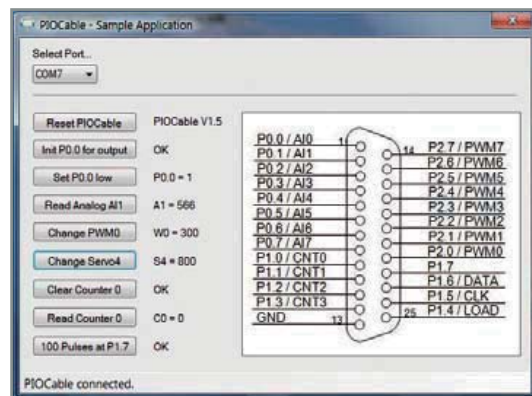


LED-Anzeigen

Mit dem MAX7219 [4] bietet Maxim einen universellen Schaltkreis an, mit dem man bis zu 64 LEDs ansteuern kann. Dazu ist bis auf einen Widerstand keine äußere Beschaltung nötig. Elektor hat bereits eine universell verwendbare 7-Segment-Anzeige mit diesem Chip vorgestellt [5], häufig werden damit aber auch Matrix-Anzeigen realisiert.

An unser USB-IO24-Kabel können zwei dieser Chips (kaskadiert) angeschlossen werden. Damit ist es möglich, bis zu 128 einzelne LEDs anzusteuern (oder zwei 8-stellige Ziffernanzeigen oder auch Kombinationen aus beidem). Die Daten werden seriell an die Anzeigen übertragen. Dazu müssen lediglich die drei Leitungen DATA, CLK und LOAD mit den gleichnamigen Pins des MAX7219 verbunden werden. Mit dem Kommando *Dn* werden dann acht Datenbytes auf einmal übertragen – z.B. *D0=48,109,121,51,91,95,38,105<CR>*. Die Bytes werden vom MAX7219 hintereinander in dessen Registern Digit 0 bis Digit 7 abgelegt; jedes Bit entspricht dabei einer angeschlossenen LED. Wie die genaue Zuordnung aussieht, steht im Datenblatt des Chips.

Bild 6.
Demo-Applikation in C#.



vielen Projekten zum Einsatz kommen. Im einfachsten Fall reicht dafür ein Bipolar- oder bei höheren Strömen auch ein Feldeffekttransistor; und wenn die Drehrichtung umgeschaltet werden soll, benötigt man noch ein Relais oder gleich eine Vollbrücke. Die Drehzahl wird dann mit Hilfe eines PWM-Ausgangs variiert. Die Schwierigkeiten beginnen, wenn man die Drehzahl regeln oder eine genaue Position anfahren möchte. Dazu ist es erforderlich, dass man einen Impulsgeber auf

PIOCable-Tool

Wie kompliziert ist die Inbetriebnahme? Woher

bekommt man die erforderlichen Treiber? Muss man erst selbst ein Programm schreiben, um die Funktionen testen zu können? Alles typische Fragen, wenn man neue Hardware an den PC anschließen möchte. Zum Glück ist die Inbetriebnahme des hier vorgestellten USB-IO24-Kabels sehr einfach. Nach dem Anschließen des USB-Steckers an den PC sucht Windows nach einem passenden Treiber. Bei den Treibern von FTDI muss man (wenn eine Internet-Verbindung besteht) meist keine weiteren Aktionen mehr ausführen, sie werden automatisch installiert. Bei Problemen findet man auf der Webseite von FTDI Hinweise zur Installation der benötigten VCP-Treiber (VCP = Virtual Com Port).

Anschließend startet man das PC-Programm PIOCable-Tool, welches speziell für die Inbetriebnahme und den Test aller Funktionen erstellt wurde (**Bild 5**) und von der Website zu diesem Artikel [6] herunterladbar ist. Auch der Sourcecode des PIOCable-Tools ist hier zu finden. Das Tool ist mit Delphi XE2 erstellt worden – der Quellcode sollte sich aber grundsätzlich auch mit früheren Delphi-Versionen übersetzen lassen.

Wenn man die Software startet und den „Connect“-Button betätigt, wird das Kabel automatisch gesucht. Sollte die Suche erfolglos bleiben, dann muss man im Geräte-Manager nachsehen, über welchen COM-Port das Kabel adressiert wird und diesen dann manuell im PIOCable-Tool einstellen.

Das PIOCable-Tool ist im Grunde ein sogenanntes Terminal-Programm. Alle Eingaben werden direkt zum Mikrocontroller des USB-IO24-Kabels gesendet, der dann seinerseits direkt antwortet. Somit kann man alle Befehle manuell eingeben und dabei (zum Beispiel mit einem Oszi) die Reaktionen an den Pins testen.

Da die Entwicklung der Kabel-Firmware weitergeht, ist es möglich, mit dem PIOCable-Tool neue Software-Versionen in den Mikrocontroller zu übertragen. Der Mikrocontroller auf der bestückten Platine wird mit einem Bootloader ausgeliefert. Spätere Software-Updates findet man ebenfalls bei Elektor [6] oder auch auf der TinyBasic-Website des Autors [7].

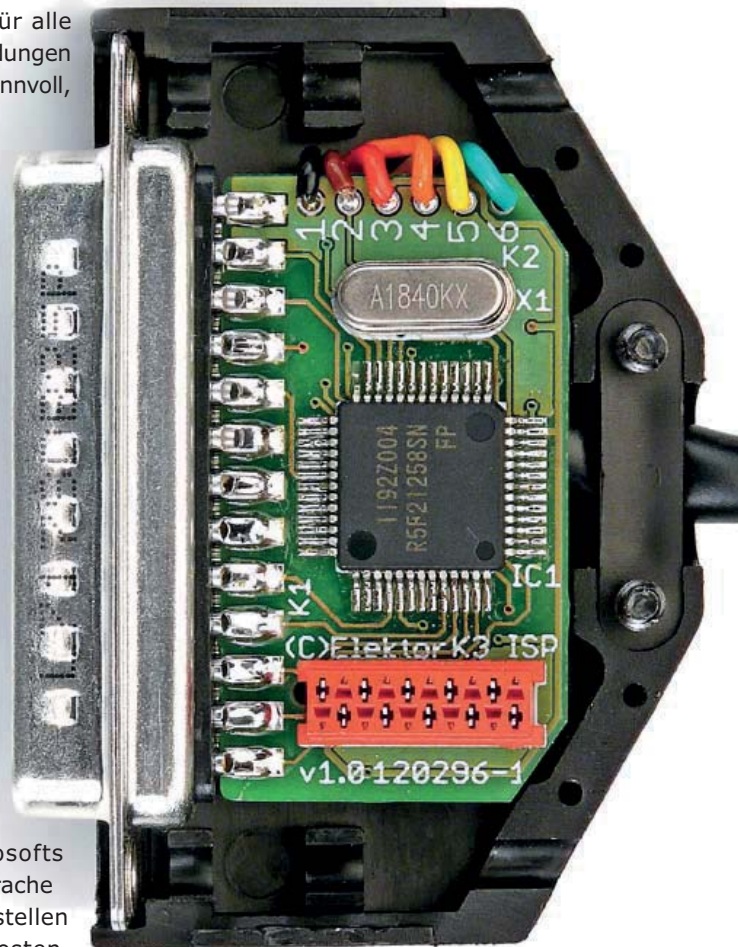
Wer eine selbst entwickelte Firmware aufspielen will, kann das natürlich ebenfalls mit dem Bootloader tun. Die Entwicklungsumgebung „High-performance Embedded Workshop“ sowie den passenden C-Compiler kann man von der Renesas-Website herunterladen [8]. Der Controller lässt

sich aber auch ohne Bootloader programmieren, einen passenden Programmer vorausgesetzt. Man muss sich dann ein kleines Adapterkabel anfertigen, das mit einem MicroMatch-Stecker ausgestattet ist (siehe Stückliste unter „Optional“). Der Bootloader (und die Original-Firmware) werden beim konventionellen Programmieren natürlich überschrieben.

Demo-Applikation

Für einfache Anwendungen (wie vielleicht einen Servo-Tester) reicht es aus, wenn man die Befehle mit Hilfe des PIOCable-Tools eingibt. Für alle anderen Anwendungen ist es natürlich sinnvoll, eine passende PC-Software zu erstellen, die dann die entsprechenden Befehle an das USB-IO24-Kabel sendet. Für die Delphi-Programmierer unter den Lesern empfiehlt es sich dabei, den Quellcode des PIOCable-Tools als Basis zu verwenden.

Besonders einfach lassen sich eigene Anwendungen aber auch mit Microsofts Programmiersprache C# für .NET erstellen (es gibt eine kostenlose „Express“-Version der Entwicklungsumgebung). Neben den üblichen Kontroll-Elementen wie Menüs, Buttons oder Labels bietet .NET auch schon eine komplette Seriell-Interface-Komponente, über die unser Kabel sehr einfach angesprochen werden kann. Zunächst zieht man eine Komponente vom Typ *SerialPort* auf ein Formular. Die Komponente muss einen Namen bekommen, z.B. „PIOPort“. Alle anderen Einstellungen macht man dann im



Quelltext. Dazu zieht man einen Button auf das Formular und schreibt in dessen OnClick-Routine die folgenden Zeilen:

```
PIOPort.ReadTimeout = 100;  
PIOPort.BaudRate = 115000;  
PIOPort.NewLine = „\r\n“;  
PIOPort.PortName = „COM5“;  
PIOPort.Open();
```

Als PortName muss man den Port eintragen, über den das Kabel aktuell adressiert wird. Wenn alles geklappt hat, ist damit die Verbindung zur Hardware hergestellt.

Der Autor hat eine Demo-Applikation in C# erstellt [6], einen Screenshot sieht man in **Bild 6**. Im Code werden die obengenannten Zeilen aufgerufen, wenn der Anwender in der DropDown-Box eine der verfügbaren Schnittstellen auswählt. Die Einstellung `PIOPort.NewLine = „\r\n“` teilt der Komponente `PIOPort` mit, dass die nachfolgenden Aufrufe von `PIOPort.WriteLine()` und `PIOPort.ReadLine()` immer mit der Zeichenfolge `<CR><LF>` enden sollen. Beim Senden eines Befehls muss man diese Zeichen also nicht extra angeben; und beim Lesen einer Antwort wird bis zum Ende einer Zeile gelesen.

Damit lässt sich nun beispielsweise eine einfache Funktion schreiben, die den Portpin P0.0 auf 1 setzt. Man zieht wieder einen Button auf das Formular und schreibt folgende Zeile in die OnClick-Routine:

```
PIOPort.WriteLine(„r0=1p0.0=1“);
```

Das Kommando setzt zunächst den Pin auf Ausgang und dann sofort auf High – fertig! Etwas

komplizierter wird es, wenn man z.B. Analogwerte lesen möchte. Hier muss man zunächst die Antwort abwarten und diese dann mit `PIOPort.ReadLine()` einlesen. Wie das geht, sieht man im Code der Demo-Applikation.

Nun steht der Programmierung einer eigenen Mess- und Steuersoftware nichts mehr im Wege! (120296)

Weblinks

- [1] www.ftdichip.com/Products/Cables/USBTTLSerial.htm
- [2] www.elektor.de/080213
- [3] www.renesas.com/products/mpumcu/r8c/r8c2x/r8c25/index.jsp
- [4] www.maximintegrated.com/datasheet/index.mvp/id/1339
- [5] www.elektor.de/081154
- [6] www.elektor.de/120296
- [7] www.tinybasic.de
- [8] www.renesas.eu/products/tools/ide/ide_hew/index.jsp



Spannung reduzieren - Mit den neuen Komponenten von Synergy21

ALLNET bietet ab sofort unter seiner Marke "SYNERGY21" ausgewählte Produkte aus dem Bereich elektronischer Komponenten und Bauelemente an.

Wie bereits bei unseren eigenen Produkten setzten wir auch hier hohe Maßstäbe an Qualität und Leistungsfähigkeit unserer Komponenten. Somit eignen sich ALLNET Komponenten der Marke "SYNERGY21" sowohl für den industriellen Sektor als auch für den anspruchsvollen Heimanwender.

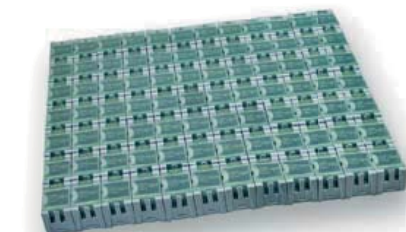
Widerstandssortiment



84 Stück



8000 Stück



Widerstandssortiment	Best.-Nr.	Preis
E12 SMD 0402 1% SET	Art. 94966	49,00E
E12 SMD 0603 1% SET	Art. 94968	51,00E
E12 SMD 0805 1% SET	Art. 94969	55,00E
E12 SMD 1206 1% SET	Art. 94970	65,00E

Überzeugen Sie sich selbst!

- SET 1** 80 x 100 Dickfilmwiderstände der E12 Serie mit $\pm 1\%$ Widerstandstoleranz, 0.0625W SMD 0402
- SET 2** 80 x 100 Dickfilmwiderstände der E12 Serie mit $\pm 1\%$ Widerstandstoleranz, 0.1W (1/10W) SMD 0603
- SET 3** 80 x 100 Dickfilmwiderstände der E12 Serie mit $\pm 1\%$ Widerstandstoleranz, 0.125W (1/8W) SMD 0805
- SET 4** 80 x 100 Dickfilmwiderstände der E12 Serie mit $\pm 1\%$ Widerstandstoleranz, 0.25W (1/4W) SMD 1206

1 Ohm	1,2 Ohm	1,5 Ohm	1,8 Ohm	2,2 Ohm	2,7 Ohm	3,3 Ohm	3,9 Ohm	4,7 Ohm	5,6 Ohm	6,8 Ohm	8,2 Ohm
10 Ohm	12 Ohm	15 Ohm	18 Ohm	22 Ohm	27 Ohm	33 Ohm	39 Ohm	47 Ohm	56 Ohm	68 Ohm	82 Ohm
100 Ohm	120 Ohm	150 Ohm	180 Ohm	220 Ohm	270 Ohm	330 Ohm	390 Ohm	470 Ohm	560 Ohm	680 Ohm	820 Ohm
1k Ohm	1,2k Ohm	1,5k Ohm	1,8k Ohm	2,2k Ohm	2,7k Ohm	3,3k Ohm	3,9k Ohm	4,7k Ohm	5,6k Ohm	6,8k Ohm	8,2k Ohm
10k Ohm	12k Ohm	15k Ohm	18k Ohm	22k Ohm	27k Ohm	33k Ohm	39k Ohm	47k Ohm	56k Ohm	68k Ohm	82k Ohm
100k Ohm	120k Ohm	150k Ohm	180k Ohm	220k Ohm	270k Ohm	330k Ohm	390k Ohm	470k Ohm	560k Ohm	680k Ohm	820k Ohm
1M Ohm	1,2M Ohm	1,5M Ohm	1,8M Ohm	2,2M Ohm	2,7M Ohm	3,3M Ohm	Blank	Blank	Blank	Blank	R0

Bedrahtete E12 Serie



84 Stück



8000 Stück

Widerstandssortiment	Best.-Nr.	Preis
E12 Bedrahtet $\pm 5\%$ Set	Art. 97227	35,00E



80 x 100 Widerstände, bedrahtete der E12 Serie mit $\pm 5\%$ Widerstandstoleranz, 0.25W

1 Ohm	1,2 Ohm	1,5 Ohm	1,8 Ohm	220 Ohm	270 Ohm	330 Ohm	390 Ohm
2,2 Ohm	2,7 Ohm	3,3 Ohm	3,9 Ohm	470 Ohm	560 Ohm	680 Ohm	820 Ohm
4,7 Ohm	5,6 Ohm	6,8 Ohm	8,2 Ohm	1k Ohm	1,2k Ohm	1,5k Ohm	1,8k Ohm
10 Ohm	12 Ohm	15 Ohm	18 Ohm	2,2k Ohm	2,7k Ohm	3,3k Ohm	3,9k Ohm
22 Ohm	27 Ohm	33 Ohm	39 Ohm	4,7k Ohm	5,6k Ohm	6,8k Ohm	8,2k Ohm
47 Ohm	56 Ohm	68 Ohm	82 Ohm	10k Ohm	12k Ohm	15k Ohm	18k Ohm
100 Ohm	120 Ohm	150 Ohm	180 Ohm	22k Ohm	27k Ohm	33k Ohm	39k Ohm
47k Ohm	56k Ohm	68k Ohm	82k Ohm				
100k Ohm	120k Ohm	150k Ohm	180k Ohm				
220k Ohm	270k Ohm	330k Ohm	390k Ohm				
470k Ohm	560k Ohm	680 Ohm	820k Ohm				
1M Ohm	1,2M Ohm	1,5M Ohm	1,8M Ohm				
2,2M Ohm	2,7M Ohm	3,3M Ohm	R0				
Blank	Blank	Blank	Blank				

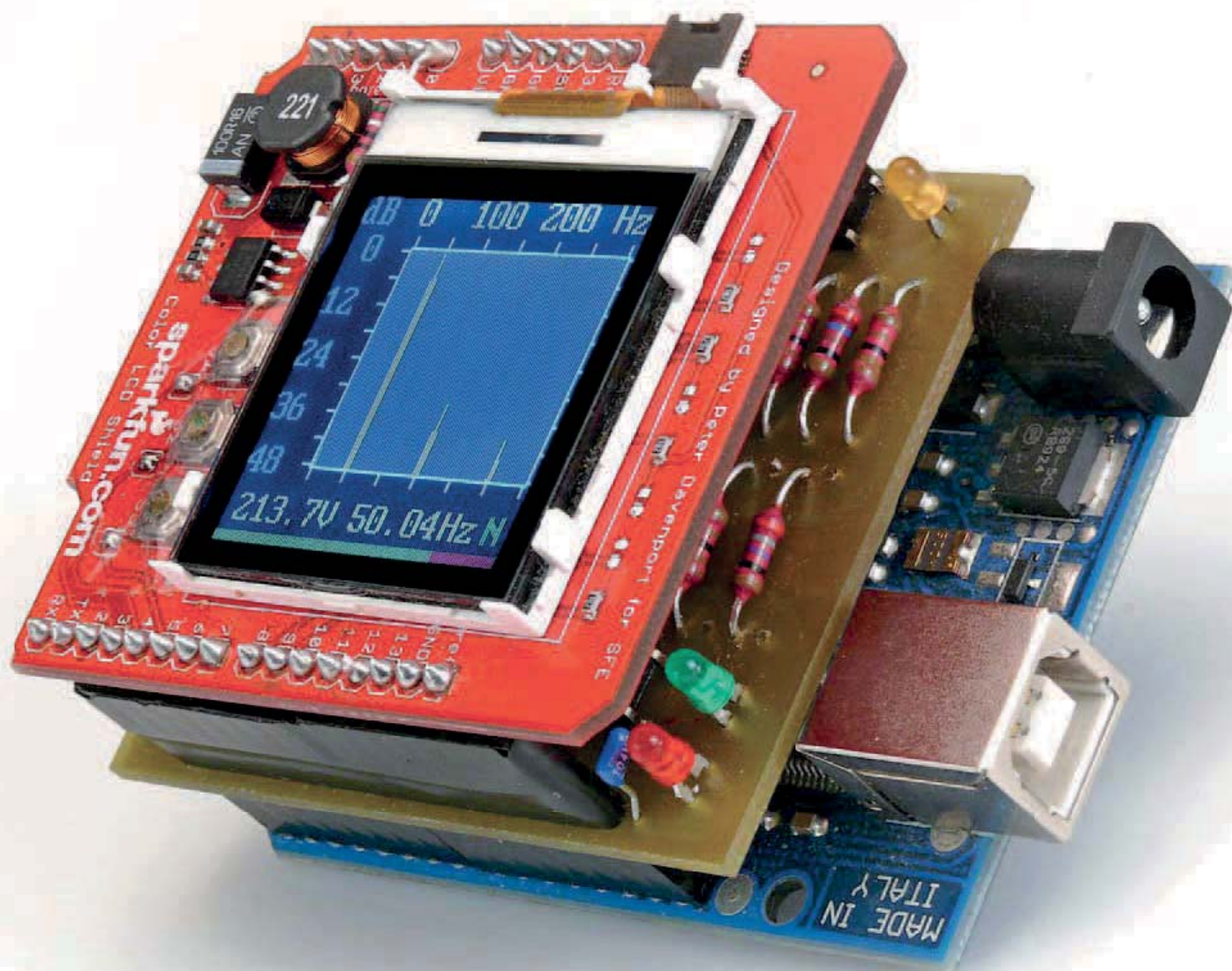


Netzspannungs- analyser mit Arduino

Oberwellenspektrum auf dem Farbdisplay

Von Fidelis Theinert
(NL)

Ein vollwertiger Spektrumanalyser mit einem kleinen Arduino-Board, geht das? Dieser Beitrag zeigt, dass auch das nicht unmöglich ist. Voraussetzung ist nur, dass die Software zur Rechenleistung passt. Hier wird mit einem Arduino-Board, einem Filter-Shield und einem Display-Shield ein Spektrumanalyser realisiert, der mit der Methode der Schnellen Fourier-Transformation die „Verunreinigungen“ im Niederspannungsnetz sichtbar macht.



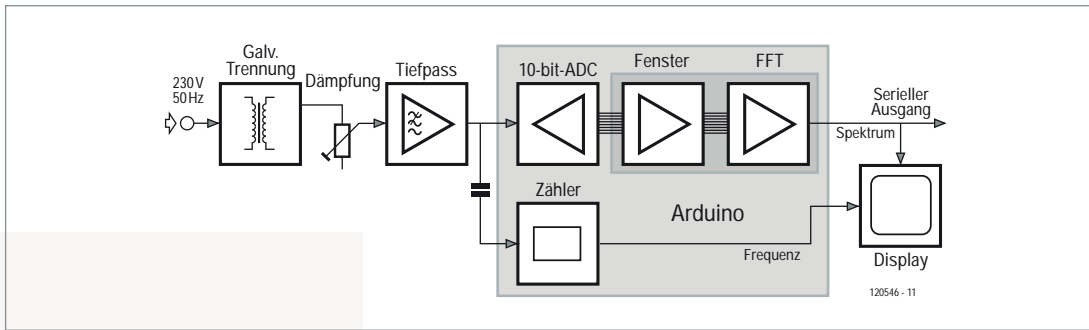


Bild 1.
Hardware-Konfiguration des
Netzspannungsanalysers.

Die beliebten Arduino-Boards sind leistungsstärker und vielseitiger als der erste Blick vermuten lässt. Der Netzspannungsanalyser beweist, dass in einem mäßig komplexen 8-bit-Mikrocontroller ein erstaunliches Potential steckt. Andererseits hat natürlich auch dieses Potential seine Grenzen. Außerdem soll dieser Beitrag zeigen, wie der Mikrocontroller seine Aufgabe möglichst effizient erfüllen kann und welche Tools zum Realisieren ähnlicher Projekte kostenfrei erhältlich sind. Die Hard- und Software des Analysers ist modular strukturiert, so dass die Komponenten unkompliziert in andere Projekte eingebunden werden können. Der Analyser wurde auf einem Arduino Duemilleanova, UNO R1, R2 und R3 erfolgreich getestet.

Der Netzspannungsanalyser wertet die Wellenform der Netzspannung nach der mathematischen Methode der Schnellen Fourier-Transformation (FFT) aus. Die Oberwellen der nur theoretisch exakt sinusförmigen Netzspannung werden grafisch auf einem Farbdisplay dargestellt. Die Grafik gibt Aufschluss über wichtige Eigenschaften des vom Energieversorger gelieferten Produkts. Bei Elektromotoren und anderen nicht rein ohmschen Lasten sind die Oberwellen für unerwünschte Verlustwärme verantwortlich, die der Kunde bezahlen muss. Auch die Wirkungsgrade schaltender Stromversorgungen, wie sie beispielsweise in jedem Desktop-PC verbaut sind, werden nachteilig beeinflusst.

Fourier-Analyse

Die Schnelle Fourier-Transformation (*Fast Fourier Transform, FFT*) ist ein mathematisches Verfahren, das diskrete Signale aus der Zeitebene in die Frequenzebene transformiert. Der Gewinn an Rechengeschwindigkeit kann verglichen mit der Diskreten Fourier-Transformation (*Discrete Fourier Transform, DFT*) und bei hoher Sample-Anzahl N erheblich sein. Die FFT unterteilt eine

Transformation der Länge N in zwei separate Transformationen der Länge $N/2$. Der Zeitgewinn beim Berechnen resultiert daraus, dass die Anzahl der mathematischen Operationen reduziert wird. Andererseits muss die Anzahl N der Samples eine Potenz von 2 sein (64, 128, 256 und so weiter), das ist in der Praxis jedoch nur selten ein Nachteil.

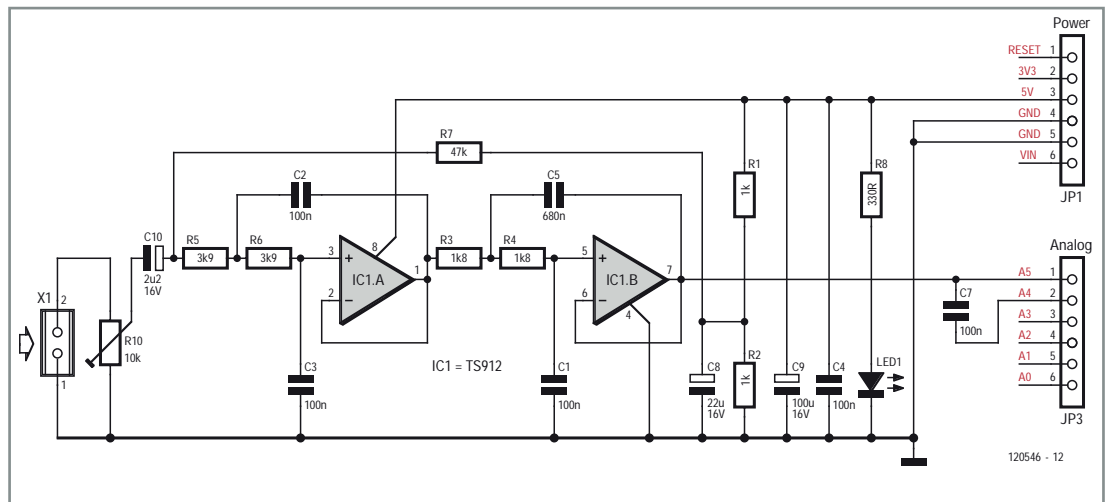
Nach dem Abtasttheorem von Nyquist und Shannon darf die Bandbreite der zu analysierenden Wellenform nicht höher als die halbe Abtastfrequenz sein. Umgekehrt muss die Abtastfrequenz also mindestens das Zweifache der höchsten vorkommenden Frequenz betragen. Mit der Abtastfrequenz 800 Hz sind folglich Signale bis zur Bandbreite 400 Hz analysierbar. Das Spektrum der Netzspannung, Frequenz 50 Hz, lässt sich mit der Abtastfrequenz 800 Hz bis zur siebten Oberwelle in seine Komponenten zerlegen ($7 \cdot 50 \text{ Hz} = 350 \text{ Hz}$).

Hardware

Zur Hardware gehören drei Platinen: Das Arduino-Board [1], das vom Autor entwickelte Filter-Shield sowie ein LC-Display-Shield [2]. Die Shields können als Sandwich auf dem Arduino-Board montiert werden (Display-Shield oben!), so dass ein kompaktes Modul ohne frei zu verlegende Leitungen entsteht. **Bild 1** zeigt die Konfiguration der Hardware.

Der serielle Port kann die Verbindung zu einem PC herstellen, der die Daten vom Arduino übernimmt und auswertet. Für die grafische Darstellung genügt ein einfaches Terminalprogramm wie zum Beispiel „Putty“. Auf dem PC-Bildschirm erscheint die Grafik mit geringer Verzögerung, da die Übertragungsgeschwindigkeit vom Arduino zum PC maximal 57600 Baud beträgt. Das Darstellen auf dem PC-Bildschirm ist immer dann eine Alternative, wenn ein Display-Shield nicht zur Verfügung steht.

Bild 2.
Antialiasingfilter
(Butterworth-Tiefpass
350 Hz).



Das Abkappen spektraler Anteile über 400 Hz im Eingangssignal übernimmt ein Sallen-Key-Tiefpass vierter Ordnung, auch „Antialiasing-Filter“ genannt, die Bandbreite beträgt 350 Hz. Die Schaltung des Tiefpasses geht aus **Bild 2** her-

vor. Die beiden Butterworth-Sektionen sind mit einem zweifachen Rail-to-rail-Opamp des Typs TS912 bestückt. Die Betriebsspannung dieses Opamps kann im Bereich 2,7...16 V liegen, die vom Arduino-Board bereitgestellte Spannung 5 V

FFT-Implementierung

Der implementierte Butterfly-Algorithmus von Cooley-Tukey gehört zur Gruppe der so genannten *In-place* Transformationen. Dies bedeutet, dass das Ergebnis nach den Rechenoperationen im gleichen Puffer steht, in dem vorher die Abtastwerte des Eingangssignals gespeichert waren. Die Abtastwerte werden folglich vom Ergebnis überschrieben. Der Vorteil liegt in der Halbierung des für den Puffer notwendigen Speicherplatzes, eine willkommene Eigenschaft, insbesondere weil der Arbeitsspeicher des Mikrocontrollers ATmega328 nur 2048 Bytes groß ist. Wie schon beschrieben, muss die Anzahl der Abtastwerte gleich einer Potenz der Zahl 2 sein. Diese Voraussetzung erfüllen die $2^8 = 256$ Abtastwerte. Die Auflösung des ADCs beträgt 1024 Schritte, so dass zum Speichern jedes Abtastwerts zwei Bytes erforderlich sind. Insgesamt werden folglich 512 Bytes belegt. Während der Rechenoperationen ist ferner ein Puffer nötig, der die Imaginärteile der Zahlenwerte aufnimmt, denn Fourier-Transformationen arbeiten stets mit komplexen Zahlen. Dieser Puffer muss den gleichen Umfang wie der Puffer der Realteile haben, auch hier werden 512 Bytes belegt. Insgesamt beträgt der Speicherbedarf 1024 Bytes, wenn 256 Abtastwerte für die FFT gespeichert werden sollen. Dies ist bereits die Hälfte des verfügbaren Arbeitsspeichers.

In einer FFT kommen nicht nur die mathematischen Grundoperationen vor, auch Winkelfunktionen (Sinus, Cosinus) und Exponentialfunktionen (Potenz, Wurzel) müssen

berechnet werden. Um diese Aufgaben mithilfe der Bibliothek mathematischer Funktionen des GCC-Compilers zu lösen, ist es sinnvoll, die Abtastwerte nicht als Integer-Variablen, sondern als Fließkomma-Variablen mit einfacher Genauigkeit zu speichern. Eine solche Variable belegt jedoch vier Bytes, so dass ein Puffer für 256 Abtastwerte 2048 Bytes umfassen müsste. Damit wäre der gesamte Arbeitsspeicher des ATmega328 belegt, so dass für andere Operationen kein Platz vorhanden ist. Das Problem könnte durch Reduzieren der FFT auf 128 Abtastwerte gelöst werden, doch das würde zu einer halbierten spektralen Auflösung führen, was natürlich nicht wünschenswert ist.

Wenn die Anzahl der Abtastwerte nicht reduziert werden soll, ist folgender Weg möglich: Die Real- und Imaginärteile der komplexen Zahlen werden getrennt als Integer-Werte gespeichert, zwei Bytes für jeden Abtastwert. Vor jeder mathematischen Operation werden die gespeicherten Werte in das Fließkomma-Format konvertiert, nach der Operation werden sie in das Integer-Format rückkonvertiert. Die für die Umwandlungen nötigen Rechenzeiten führen zwar zu einer niedrigeren Arbeitsgeschwindigkeit, die Verzögerung ist jedoch tolerierbar. Der Rundungsfehler $1/65536$, der durch die Umwandlungen verursacht wird, ist um den Faktor 64 kleiner als der Quantisierungsfehler $1/1024$ des ADC, so dass er vernachlässigt werden kann (siehe Funktionen `FFT_Int2Float()` und `FFT_Float2Int()` im Modul `fft_lib.c`).

liegt in diesem Bereich.

Der Filterausgang (Pin 7 des TS912) ist mit Kanal 5 des im Mikrocontroller ATmega328 integrierten A/D-Wandlers (ADC) verbunden. Damit auch die Netzfrequenz gemessen werden kann, wird das Ausgangssignal über Kondensator C7 zusätzlich dem Analog-Eingang A4 des Arduino-Boards zugeführt.

In **Bild 3** ist das Platinenlayout für das Filter-Shield wiedergegeben. Auf der Platine sind einige weitere Komponenten montierbar, die hier nicht benötigt werden. Das Platinenlayout steht im Internet auf der Elektor-Projektseite [3] zum kostenlosen Download bereit.

Ein Steckernetzteil (konventioneller Typ mit Trafo, ohne Gleichrichter) setzt die Netzspannung herab und sorgt für die galvanische Trennung. Die sekundäre Spannung wird an Filtereingang X1 gelegt. Da die Filterverstärkung 1 beträgt, genügt eine Spannung von etwa $2 V_{eff}$ ($5 V_{SS}$), um den ADC des ATmega328 auszusteuern. Der Einfluss des Trafos auf die Wellenform ist vernach-

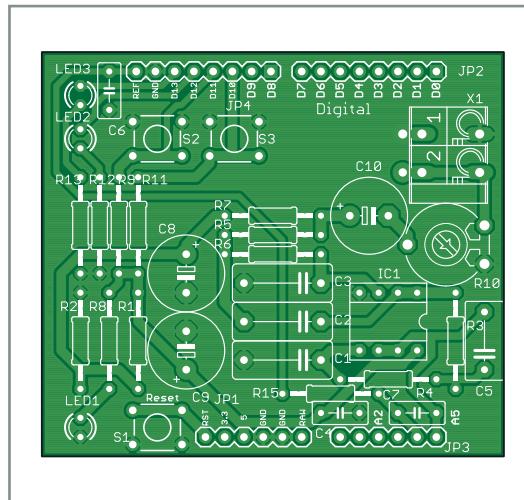
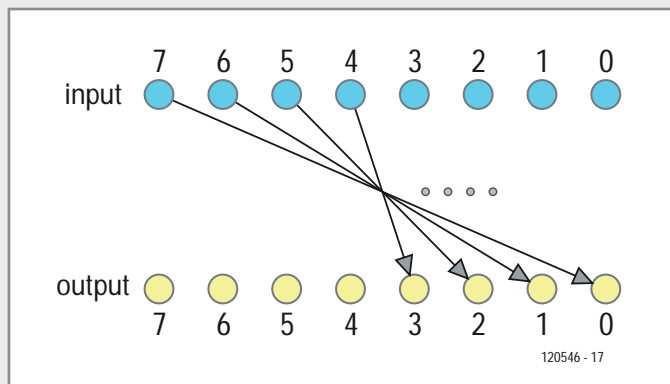


Bild 3.
Platinenlayout für das Filter-Shield.

lässigbar, weil am Trafo keine nennenswerte Last liegt (siehe **Bild 5**). Das Filter-Shield wird nicht vom Steckernetzteil, sondern vom Arduino-Board oder über USB mit Strom versorgt. Potentiometer R10 ist zunächst so einzustellen,

Um den Arbeitsspeicher möglichst effizient zu nutzen, werden die vom ADC kommenden Abtastwerte zuerst im Puffer der Imaginärteile gespeichert. Zu diesem Zeitpunkt greift die FFT auf diesen Puffer noch nicht zu.



Bitreversal-Algorithmus

Die Abtastwerte werden normiert und skaliert, anschließend werden sie in den Puffer der Realteile übertragen. Der Puffer der Imaginärteile ist wieder frei, und auf die Realteile kann die Fensterfunktion angewendet werden. Die Fensterfunktion ist notwendig, um die Randwertbedingungen der FFT zu erfüllen.

Zur FFT-Implementierung gehört der so genannte *Bitreversal*-Algorithmus, dieser Algorithmus wird zur Adressierung der in den Puffern stehenden Abtastwerte benutzt. Von außen betrachtet ist der Algorithmus leicht überschaubar: Die Reihenfolge der in einem Register stehenden Bits wird lediglich umgekehrt (siehe Skizze). Leider gibt es für diese Operation keinen spezifischen Befehl im Befehlssatz des Mikrocontrollers. Schnelle Implementierungen benutzen zusätzliche Hardware-Register, andere greifen auf Tabellen zurück, die jedoch kostbaren Speicherplatz belegen. Am langsamsten ist die Variante, bei der alle Bits aus einem Register geschoben werden, um sie in umgekehrter Reihenfolge in ein anderes Register einzuschieben. Die notwendigen 16 Schiebeoperationen müssen von einem Zähler gesteuert werden, der in einer Schleife zu aktualisieren ist. Einen gangbaren Kompromiss zwischen Speicherbedarf und Rechenzeit stellt die Kombination beider Methoden dar. Ein Byte wird in zwei Halbytes (4 bit) unterteilt, die Halbytes werden mithilfe einer Tabelle konvertiert. Die Ergebnisse werden in der korrekten Reihenfolge zusammengefügt (siehe Funktion `fft_BitRev()` im Modul `fft_lib.c`). Der Zeitgewinn beträgt $3 \mu s$ für jeden Funktionsaufruf ($2 \mu s$ statt $5 \mu s$), allerdings wird dies durch einen 75 Bytes längeren Programmcode erkauft.

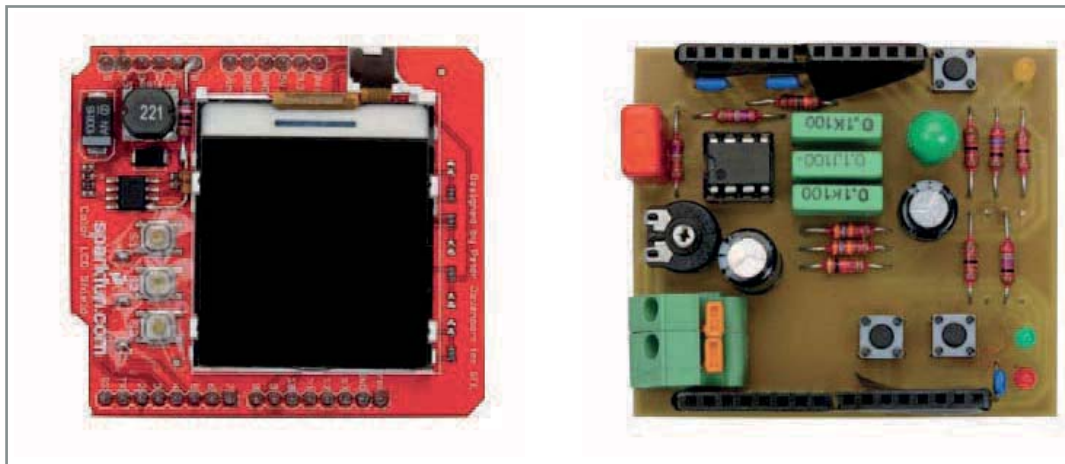


Bild 4.
Die drei Platinen in Szene
gesetzt.

dass bei 230 V Netzspannung etwa $1,7 V_{\text{eff}}$ an C10 liegen. Dann wird das Poti nachgestellt, bis der vom Display angezeigte Wert gleich dem tatsächlichen Wert der Netzspannung ist. Zum Vergleichen wird die Netzspannung mit einem möglichst genauen Multimeter gemessen. Die Spannung an Filtereingang X1 soll $20 V_{\text{eff}}$ nicht übersteigen.

Entwicklungsumgebung

Die Software wurde nicht unter der Arduino-Programmierungsumgebung entwickelt, Entwicklungsumgebung war der GCC Cross Compiler für AVR-Mikrocontroller. In der Linux-Welt kann der Compiler *cross-avr* mit den zugehörigen *bin-utils* aus dem

Paketarchiv der Distribution installiert werden, für Windows ist WinAVR [4] das richtige Werkzeug. Die Varianten beider Betriebssysteme bauen auf Version 4.3.3 des GCC-Compilers auf [5]. Zum Laden der Hex-Dateien wird das Programm „AVRdude“ genutzt, das in WinAVR für Windows bereits enthalten ist, eine Linux-Version existiert ebenfalls. Die genannten Entwicklungsumgebungen beider Plattformen schließen Skript-Dateien ein, die ausführbare Programme in den Arduino laden. In eine Skript-Datei muss nur noch der serielle Port des Arduino eingetragen werden, außerdem ist der Pfad anzugeben, in dem die zu ladende Datei auf dem PC gespeichert ist. Das Compilieren und Linken der Quellcodes aller

Tabelle 1. Software-Module

Source-Datei	Header-Datei	Funktion
avrfft.c	(keine)	Main-Source-Code-Datei
drw_fft.c	drw_fft.h	High-level-Draw-Funktionen für die Grafik-Darstellung auf dem LC-Display
fft_lib.c	fft_lib.h	Definitionen und Funktionen für die FFT
freq_cnt.c	freq_cnt.h	Definitionen und Funktionen für die Datenerfassung und Frequenzmessung
lcd_driver.c	lcd_driver.h	Low-level-Funktionen und Definitionen für die LC-Display-Steuerung
menu.c	menu.h	Funktionen für die Abfrage der Bedientaster auf dem LC-Display-Shield
plt_fft.c	plt_fft.h	High-level-Funktionen für die Grafik-Ausgabe über die serielle Schnittstelle
ser_avr.c	ser_avr.h	Low-level-Funktionen zum Senden und Empfangen von Daten
std_avr.c	std_avr.h	Allgemeine Standard-Definitionen und -Funktionen
timer_0.c	timer.h	Timer-Funktionen und Konfiguration des System-Timing

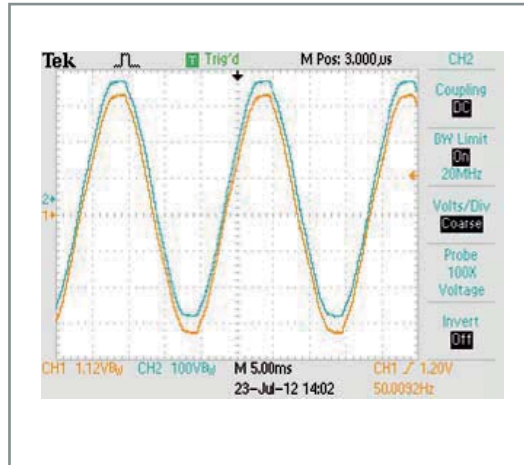
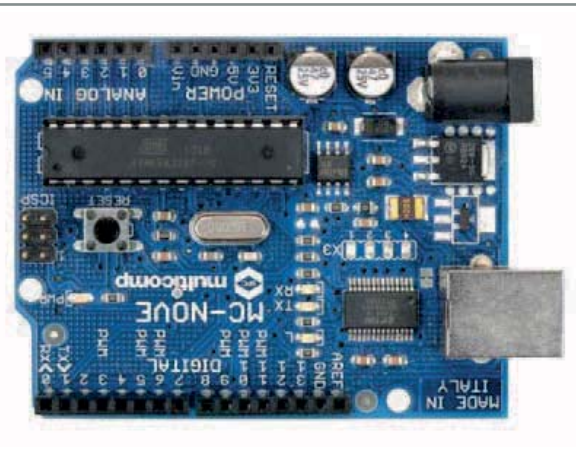


Bild 5.
Spannungsverlauf auf der Primärseite (blau) und Sekundärseite (gelb) des Netztrafos.

Module übernimmt das Tool *make*. Auch dieses Tool ist Bestandteil des Pakets WinAVR, auch hier existiert eine Linux-Version.

Die Quellcodes sind mit beliebigen Texteditoren modifizier- oder erweiterbar. Mit Eclipse und der CDT-Erweiterung ist komfortables Arbeiten möglich, auch Notepad++, Scite oder vergleichbare Editoren sind gut brauchbar.

Software

Die Software ist modular strukturiert, sie ist in die Source-Dateien gegliedert, die in **Tabelle 1** gelistet sind.

Alle C-Programme beginnen mit der Funktion *main()*, die hier in der Datei *avrfft.c* steht. Nach dem Programmstart werden diverse globale Systemvariablen sowie die Hardware durch die zugehörigen *init*-Funktionen initialisiert. Das LC-Display wird gelöscht und in den System-Timer werden die nötigen Parameter geschrieben. Die Initialisierung mündet in eine Endlosschleife ein, die vom System-Timer gesteuert zyklisch in 2,5-s-Intervallen durchlaufen wird. Innerhalb jedes Zyklus werden die Operationen in stets gleicher Reihenfolge ausgeführt. **Bild 6** zeigt ein vereinfachtes Flussdiagramm der Endlosschleife.

Der Zyklus beginnt mit dem Abtasten von 256 Werten des Eingangssignals mit der Abtastfrequenz 800 Hz. Anschließend wird der wahre Effektivwert (TRMS) der Spannung berechnet. Der nächste Schritt ist die Frequenzmessung des Signals, das an Eingang *Analog In* (A4 von JP3) liegt. Der Mikrocontroller führt die Frequenzmessung mit seinem internen Komparator und dem Timer 1 durch.

Die nächsten Schritte sind das Skalieren der Messwerte und das Anwenden der gewünschten Fensterfunktion. Diese Operationen schaffen die Voraussetzungen für die dann folgende FFT-Analyse. Zum Schluss werden die Daten des berechneten Spektrums zum LC-Display-Shield übertragen und dort als Grafik dargestellt.

Display und Bedienung

Das Farbdisplay des LC-Display-Shield hat die Auflösung 131 · 131 Pixel, für das Darstellen des Frequenzspektrums sind horizontal 96 Pixel und vertikal 80 Pixel reserviert. Die Frequenzachse verläuft von 0 Hz bis 300 Hz (96 Linien · 3,125 Hz = 300 Hz), während vertikal 10 Pixel gleichbedeutend mit 6 dB sind. Von der FFT wird zwar das Spektrum 0...400 Hz berechnet, die Signalanteile oberhalb 300 Hz werden jedoch nicht mehr angezeigt. Die Daten, die über die serielle Schnittstelle ausgegeben werden, sind von dieser Begrenzung nicht betroffen. Auf dem Display sind unter der Grafik die Netzspannung und die Netzfrequenz als Zahlenwerte ablesbar. Außerdem wird die Netzspannung als proportionaler Balken dargestellt.

Links neben dem Display befinden sich drei Taster für die Bedienung. Nach dem Einschalten stellt der Netzspannungsanalyser das Frequenzspektrum logarithmisch dar. Nach Drücken des oberen Tasters wechselt die Skalierung der vertikalen Achse in den linearen Modus. Der nächste Druck auf diesen Taster schaltet den Netzspannungsanalyser in die Darstellung des Zeitbereichs (Oszilloskop-Funktion). Auf dem Display werden skaliert die ersten 96 Abtastwerte dargestellt, die im Eingangspuffer gespeichert sind. Die **Bilder 7 und 8** geben einen Eindruck von dem, was auf dem Display zu sehen ist. Der nächste Druck auf den Taster hat zur Folge, dass die gewählte

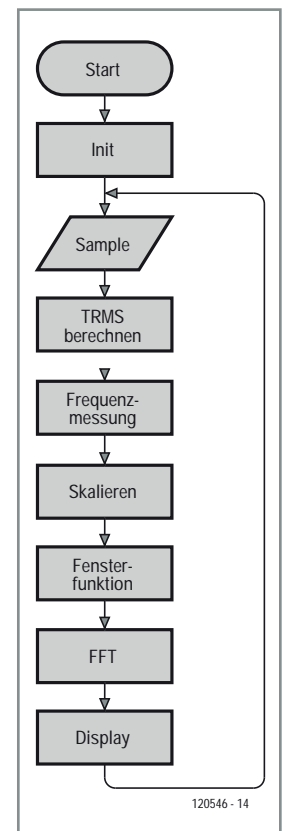


Bild 6.
Vereinfachtes Flussdiagramm der Software.

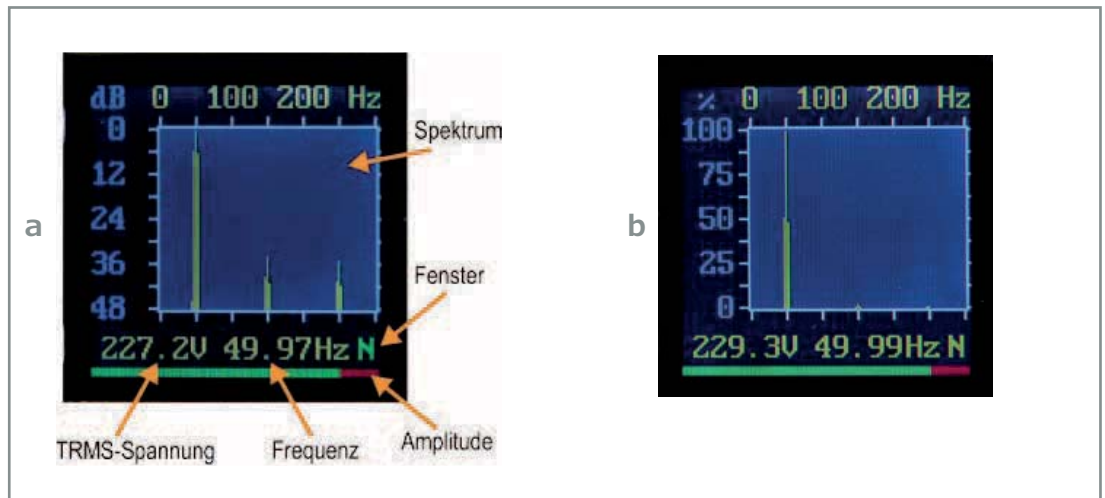


Bild 7.
Grafische Darstellung der
Frequenzebene auf dem
Display (a = logarithmisch,
b = linear).

Fensterfunktion auf den gesamten Inhalt des Eingangspuffers angewendet wird, das Ergebnis erscheint auf dem Display. Der Rücksprung zur logarithmischen Darstellung des Frequenzbereichs wird durch den vierten Druck auf den Taster ausgelöst.

Mit dem unteren Taster lässt sich die anzuwendende FFT-Fensterfunktion umschalten. Die Grundeinstellung ist das Hanning-Fenster, ein Tasterdruck schaltet auf das Blackman-Fenster um, dann folgt das Hamming-Fenster. Die aktive Fensterfunktion wird unten rechts auf dem Display angezeigt, wobei N für Hanning, B für Blackman und M für Hamming stehen.

Der mittlere Taster schaltet den seriellen Datenaustausch mit einem auf dem PC laufenden Terminal-Programm an oder aus. Wenn keine Daten übertragen werden, kehrt die Anzeige der Fensterfunktion nach Berechnen des Spektrums von gelber Schrift nach grüner Schrift zurück. Solange ein serieller Datenaustausch stattfindet, erscheint diese Anzeige in Rot.

Performance

Die Zeit, die das Durchführen einer FFT in Anspruch nimmt, hängt von den Einstellungen beim Compilieren ab. Im ungünstigsten Fall und ohne Code-Optimierung (Parameter -O0) beträgt die Dauer etwa 1 s, das compilierte Programm hat einen Umfang von ungefähr 26 kB. Die Code-Optimierung (-O2) verkürzt das Programm auf rund 18 kB, und eine FFT dauert nur noch 0,7 s. Ein vollständiger Zyklus (Frequenz messen, Daten aufbereiten, TRMS berechnen, Skalieren, Fensterfunktion anwenden, FFT durchführen, Skalieren, Ergebnis auf dem Display darstellen) wird

mit vorangegangener Code-Optimierung in zirka 2,1 s durchlaufen. Aufgrund dieser Werte wurde die Zykluszeit auf 2,5 s festgelegt, so dass eine vorhersehbare konstante Reaktionszeit des Systems gewährleistet ist. Über die Bedeutungen der Optimierungsparameter (-Ox) gibt die Dokumentation des GCC-Compilers Auskunft.

Von den 2048 Bytes des Arbeitsspeichers im Mikrocontroller werden 1372 Bytes genutzt (rund zwei Drittel), davon entfallen 1024 Byte auf den Eingangspuffer der FFT. Der 32 KB große Programmspeicher ist nur mit etwa 18 KB belegt. Da der Bootlader des Arduino mit rund 2 KB in die Programmspeicher-Bilanz eingeht, sind noch 10 KB für optionale Erweiterungen frei.

Resümee

Der Netzspannungsanalyser ist ein Echtzeit-FFT-Analyser mit voller Funktionalität. Die FFT arbeitet mit 256 Fließkommawerten, die berechneten Spektren werden in 2,5 s langen Intervallen grafisch auf einem Farbdisplay dargestellt. Das System wurde auf der Arduino-Plattform implementiert, die Software belegt nur etwa ein Drittel des verfügbaren Programmspeichers.

Das Projekt demonstriert, wie ein durchschnittlicher 8-bit-Mikrocontroller effizient eingesetzt werden kann, um komplexe mathematische Operationen auszuführen und die Ergebnisse grafisch darzustellen. Außerdem bedient der Mikrocontroller eine Schnittstelle zu einem PC. Grenzen werden insbesondere durch die Größe des verfügbaren Arbeitsspeichers gesetzt, die den Umfang der FFT auf 256 Werte beschränkt. Die Rechengeschwindigkeit des Mikrocontrollers stellt ebenfalls eine systembedingte Grenze dar. In dieser

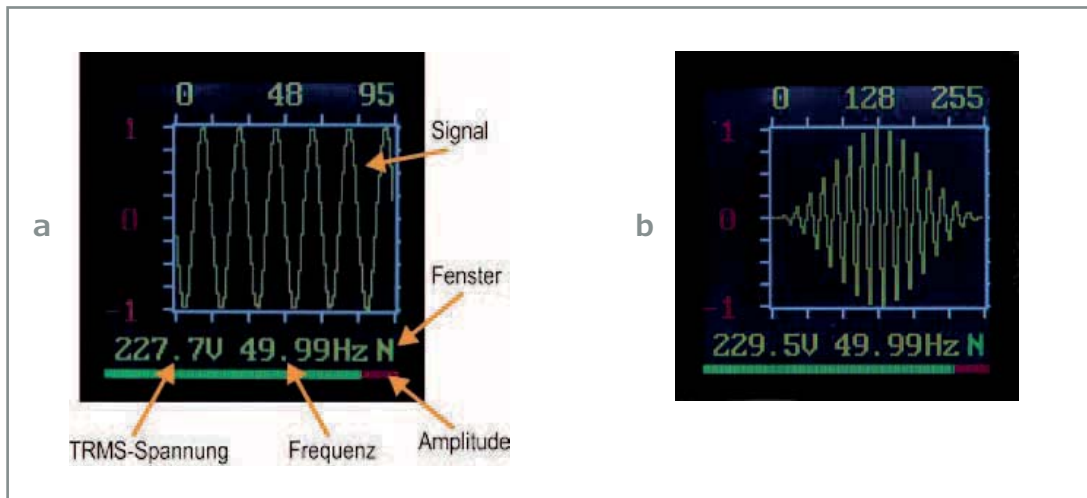


Bild 8.
Grafische Darstellung der Zeitebene auf dem Display (a = 96 Abtastwerte ohne Fensterfunktion, b = 256 Abtastwerte mit Hanning-Fenster).

Hinsicht besteht einiger Spielraum für weitere Optimierungen.

Das Projekt wurde ausschließlich mit Open-Source-Tools entwickelt, auch die Arduino-Plattform gehört vollständig der Open-Source-Domäne an.

Die modular strukturierte Software dieses Projekts wurde unter der GNU General Public License, Version 3 veröffentlicht. Die Bedingungen der GNU GPL V3 können auf der Website des GNU Operating System [6] nachgelesen werden. Die Projekt-Software und das Platinenlayout stehen auf der Elektor-Website [3] kostenfrei zur Verfügung.

Ausblick

Trotz der umfangreichen Funktionalität, die im Netzspannungsanalyser bereits implementiert ist, sind diverse Upgrades denkbar. Beispiele sind das Berechnen des THD-Faktors (*THD = Total Harmonic Distortion*) oder der Ausbau zum Analyser für Sprachsignale. Das Verarbeiten von Sprachsignalen setzt eine Anpassung des Tiefpasses am Eingang voraus. Der ADC des ATmega328 kann bei der Auflösung 10 bit mit der Abtastfrequenz 15 kHz arbeiten, was für die Bandbreite 5 kHz bei Sprachsignalen voll ausreicht.

Die Arbeitsgeschwindigkeit des Systems lässt sich steigern, wenn die Abtastwerte im Hintergrund von Interrupt-Routinen erfasst werden. Während dieser Zeiten muss die Ausgabe an das Display noch nicht beendet sein. Der Zeitgewinn liegt überschlägig bei 0,3 s (256 Werte / Abtastfrequenz 800 Hz). Ferner ist ein Zeitgewinn durch Steuern des Displays über die SPI-Schnittstelle des ATmega328 möglich.

Eine modern gestaltete grafische Benutzeroberfläche auf dem PC liegt ebenfalls im Bereich der Möglichkeiten.

(120546)gd

Weblinks

- [1] <http://arduino.cc/en/>
- [2] www.sparkfun.com/products/9363
- [3] www.elektor.de/120546
- [4] <http://sourceforge.net/projects/winavr/>
- [5] <http://gcc.gnu.org/onlinedocs/gccint/index.html>
- [6] www.gnu.org/copyleft/gpl.html

Der Autor

Fidelis Theinert war nach Absolvieren des Studiums der Elektrotechnik an der TU Berlin über zwei Jahrzehnte in deutschen und niederländischen Unternehmen als Hard- und Software-Entwickler tätig. Nach einem Lehrauftrag an einer Höheren Berufsfachschule in den Niederlanden ist er heute Dozent im Fachbereich Elektrotechnik an der Haager Hochschule in Delft. Die Tätigkeitsschwerpunkte liegen auf der Anwendung eingebetteter Systeme für die Signalverarbeitung in der Audio- und Videotechnik.



Geschenkideen für Elektroniker

Beschenken Sie sich und andere oder lassen Sie sich beschenken!

8-W-LED-Werkstattlupenleuchte mit 1,75facher Vergrößerung



- Helles, weißes LED-Arbeitslicht (6500 K), 80 LEDs
- Nur 8 W Leistungsaufnahme – ca. 60 % weniger Energieverbrauch gegenüber einer vergleichbaren Kaltlicht-Lupenleuchte mit 22-W-Leuchtstoffring
- Kein Flackern, sofort volles Licht, gleichmäßig verteilt
- Langlebige LED: bis 20.000 Betriebsstunden
- Stabiler Gelenkarm, 80 cm lang
- Robuste Tischplattenklemme bis 65 mm
- Wartungsfreies und stabiles Haltefedersystem – kein ungewolltes Absinken des Lupenkopfes
- Echtglaslupe, ø 127 mm, 1,75fache Vergrößerung (3 Dioptrien)
- Verzerrungsfreie Vergrößerung über die komplette Linienfläche
- Direktbetrieb an 230 V (kein Trafo)



Digital-Multimeter/RLC-Meter VC88C



Mit über großem Display mit 28 mm hohen Zeichen, beleuchtbar. Die RLC-Messbereiche ersetzen komplett ein separates RLC-Meter. Ein weiterer Frequenzmessbereich bis 10 MHz, die Transistortestfunktion und die externe Temperaturmessmöglichkeit bis 1000 °C machen die Ausstattung komplett. Abm. (B x H x T): 97 x 189 x 35 mm. Lieferung mit Messleitungen, 9-V-Blockbatterie, K-Type-Tempersensordr und Messadapter für Transistortest, LC-Messungen.

Scanner-Maus LSM-100



WF-10 17 86

€ 79,95

In Sekundenschnelle ist mit dieser Maus eine ganze DIN-A3-Seite eingescannt und kann so auf dem Rechner gespeichert oder weiterbearbeitet werden.

Lieferung inkl. Mauspad, Schutzbeutel, Reinigungstuch und Installations-CD

Abm. (B x H x T): 65 x 35 x 115 mm

- Technische Daten Maus: 2 Tasten und 1 Scrollrad, Laser-Sensor mit 1200 dpi
- Technische Daten Scanner: 2 Tasten (Scannen & Zurück) Einstellbar bis 320 dpi (320/200/100), 640 x 300 Pixel bei 30 Hz; jede Größe bis DIN A3 scanbar
- Speicherformate: JPEG, TIFF, PNG, BMP, XLS, DOC, PDF
- OCR-Texterkennung
- USB-2.0-Anschluss
- Für Windows XP/Vista/7



Durchschnittliche Bewertung der ELV-Kunden im Web-Shop



(7 Produktbewertungen)

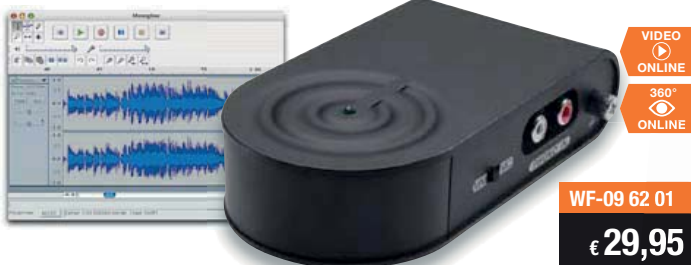
(Stand 10.2012)

TECHNISCHE DATEN

*Grundgenauigkeit

Gleichspannung	200 mV/2/20/200/1000 V; 0,5 %*
Wechselspannung	200 mV/2/20/200/750 V; 0,8 %*
Gleich-/Wechselstrom	2/20/200 mA/20 A; 0,8/1,0 %
Widerstand	200 Ω/2/20/200 kΩ/2/20/2000 MΩ; 0,8 %*
Kapazität – Induktivität	20/200 nF/2/20/200 µF; 2,5 % – 2/20/200 mH/2/20 H; 2,5 %
Temperatur	-40 bis +1000 °C; 1,0 %
Frequenz	2/20/200/2000 kHz/10 MHz; 0,5 %
Sonstiges	Dioden- und Transistortestfunktion, akustischer Durchgangsprüfer, Peak-Hold-Funktion, automatische Abschaltung, Batteriewarnung

Audio-Digitalisierer AD1209



VIDEO
ONLINE
360°
ONLINE

WF-09 62 01

€ 29,95

Der AD 1209 ist sowohl zur Digitalisierung von analogen Audioaufnahmen als auch als Vorverstärker für den Betrieb von Plattenspieler geeignet.

- Entzerrer-Vorverstärker mit RIAA-Kennlinie
- Geeignet für MC- (Moving Coil) und MM-Tonabnehmer (Moving Magnet)
- USB-Port für den Computeranschluss und die Spannungsversorgung
- Vorverstärkerausgang für den Anschluss eines Stereo-Verstärkers

Durchschnittliche Bewertung der ELV-Kunden im Web-Shop



(17 Produktbewertungen)

(Stand 10.2012)

kers für die Schallplatten-Wiedergabe

- Hochwertige Audio-Digitalisierung mit bis zu 48 kHz Samplingrate
- Plug & Play: nutzt Betriebssystemtreiber (MS Windows XP/ Vista/7 und Mac OS X)

Lieferung inkl. USB-Kabel und Cinch-Klinke-Adapterkabel (2x Cinch auf 1x Klinke)

Audio Tape Express



VIDEO
ONLINE

WF-08 95 89

€ 29,95

Der kompakte USB-Kassettenplayer kann sowohl zum normalen Abspielen von Kassetten als auch zum Übertragen der Musikdaten via USB auf einen PC genutzt werden.

- Für alle Kassettenarten, auch Chrom-/Metall-Kassetten
- Zweirichtungsbetrieb

- Plug-&-Play-Anschluss ohne Treiber – systemübergreifend anschließbar

Abm. (B x H x T): 112 x 29 x 78 mm

Lieferung inkl. USB-Kabel und Software zur Konvertierung von MP3-Dateien (Windows XP/ Vista/7/Mac ab Version 10.5)

Über 1100 Multimedia-Produkte finden Sie auf: www.multimedia.elv.de

LED-Taschenlampe M7R mit innovativem Ladesystem



- Einfaches Aufladen über innovative magnetische Halterung
- 3 verschiedene Lichtprogramme und 8 Lichtfunktionen
- Extrem hell: Lichtstrom*: bis zu 220 lm, Leuchtweite*: bis zu 255 m
- Fokussierbar
- Energiesparmodus für optimierte Laufzeit
- Ladezustandsüberwachung

Abm. (ø x H): 37 x 156 mm.

Lieferung inkl. Akku, Ladehalterung mit Floating Charge, System für Wandmontage, Intelligent Clip, Handschlaufe, Kunststoffkoffer



VIDEO
ONLINE

WF-10 28 21

€ 129,95

Weitere Infos: www.taschenlampen.elv.de

* Lichtstrom (Lumen/lm) bzw. maximale Leuchtweite (Meter) gemessen in der hellsten Einstellung beim Einschalten mit einem frischen Satz Lithium-Batterien. Es handelt sich um durchschnittliche Werte, welche im Einzelfall je nach Chip und Batterien um ±15 % abweichen können.

Ladegerät BC 700



VIDEO
ONLINE
360°
ONLINE

WF-06 90 75

€ 24,95

Der Weltweit-Lader mit dem individuell wählbaren Ladeprogramm für jeden Akku: lädt, entlädt, frischt ältere oder länger gelagerte Akkus wieder auf und testet die Kapazität Ihrer Akkus.

Durchschnittliche Bewertung der ELV-Kunden im Web-Shop



(65 Produktbewertungen)

(Stand 10.2012)

- Für 1–4 Mignon-/Micro-Akkus, NiMH/NiCd gemischt bestückbar, automatische Typ-Erkennung
 - Akku-Defekt-Erkennung mit Ladeunterbrechung
 - Ladeströme: 200/500/700 mA, individuell einstellbar
 - Einzelanzeige für jeden Ladeschacht: Lade-/Entladestatus, Kapazität, Ladestrom, Akkuspaltung, Ladezeit
 - Parameter für jeden Ladeschacht einzeln einstellbar
 - Temperatur- und -Δ-Ladeend-Überwachung für jeden Ladeschacht
 - Weltweit nutzbares Netzteil
- Lieferung ohne Akkus

Gleich mitbestellen: ELV Power-NiMH-Akku, 2er-Pack

Bis zu 2700 mA

Wiederaufladbare NiMH-Akkus für lang anhaltende Power

- Hohe Energiedichte, geeignet für stromintensive Anwendungen
- Hervorragender Ersatz für Batterien
- Bis zu 1000 Schnellladezyklen
- Geringe Selbstentladung
- TÜV-zertifiziert*



ab

€ 3,95



Typ	Kapazität	Best.-Nr.	Preis
Micro	1000 mAh	WF-10 72 93	€ 3,45
Mignon*	2700 mAh	WF-10 72 94	€ 6,95

Bestell-Hotline: + 49 (0)491/6008-88

www.elv.de

- ✓ Innovative Produkte, viele aus eigener Herstellung
- ✓ Ausführliche Produktbeschreibungen
- ✓ Kauf ohne Risiko

Geschenkideen für Elektronikprojekte

80-W-Komfort-Lötstation LF-1600



WF-10 76 72

€ 69,95

Leistungsfähige und kompakte Lötstation mit schneller Anheizzeit, weitem Temperatureinstellbereich und hoher Leistung.

- Löttemperaturbereich 200 bis 480 °C, Leistung 80 W
 - Einfache Temperatureinstellung mit Drucktasten, Anzeige von Soll- und Ist-Temperatur über großes LED-Display
 - Kurze Aufheizzeit, in ca. 45 Sekunden wird eine Lötspitzen-temperatur von 400 °C erreicht
 - 32-V-NiederspannungslötKolben, innenbeheizt
 - ESD-sicheres Arbeiten durch Nulldurchgangssteuerung und galvanische Netztrennung, Potentialausgleichsbuchse
 - Fehleranzeige bei Heizelement- und Sensordefekt
 - SMD-Tweezer-LötKolben direkt anschließbar (optional erhältlich, Angebot siehe Web-Shop)
 - Kompaktes und robustes Metallgehäuse: 105 x 126 x 90 mm
- Lieferung inkl. LötKolben 108ESD mit 1,0-mm-Lötspitze, bleistiftspitz, LötKolbenablage mit Metallschwamm und Bedienungsanleitung

Gleich mitbestellen: No-Clean-Lötdraht Sn99Cu1+ML



- Bleifrei
- 227 °C Schmelztemperatur
- Lieferung auf 100-g-Spule

Ø	Best.-Nr.	Preis
0,5 mm	WF-10 76 78	€ 6,95
1,0 mm	WF-10 76 79	€ 5,95
1,5 mm	WF-10 76 80	€ 5,50

Viele weitere Produkte und Informationen finden Sie unter: www.loettechnik.elv.de

Taschenoszilloskop HPS 140 I inkl. Tastkopf

velleman[®]



WF-09 61 29

€ 119,95

**Oszilloskop, Digitalvoltmeter und
Audioleitungsmessung in einem Gerät**

- Echtzeit-Abtastzeit 40 MS/s, Analogbandbreite 10 MHz, max. 100 Vss AC + DC
 - Einfach bedienbar – automatische Parametereinstellung/ Menüführung
 - 1 mV bis 20 V/DIV, 250 ns bis 1 h/DIV
 - Empfindlichkeit 0,1 mV
 - Hold- und Speicherfunktion
 - DVM-Anzeige
 - Audio-Leistungsmessung (2–32 Ω)
 - Kalibriersignal für Tastkopf
 - Bis zu 6 Stunden Akkubetrieb möglich
- Abm. (B x H x T): 74 x 114 x 29 mm, 200 g. Lieferung inkl. Ladekabel mit USB-Anschluss am PC und 1:10-Tastkopf

USB-Temperatur-Feuchte-Datenlogger TFD 128



WF-06 68 28

€ 49,95

Der batteriebetriebene Datenlogger sammelt an seinem Einsatzort über einen integrierten Sensor automatisch Temperatur-/Luftfeuchtedaten für bis zu 151 Tage (nur Temperatur: bis 227 Tage) und stellt diese über eine USB-Schnittstelle einem PC zur Auswertung zur Verfügung. Damit sind Klimadaten auch an weit abgelegenen Orten über lange Zeiträume erfassbar. Über ein Auswerteprogramm sind die Da-

ten grafisch und tabellarisch darstell- sowie für die weitere Verarbeitung speicherbar.

- Messbereich Temperatur: -20 °C bis + 80 °C, Auflösung/ Toleranz: 0,1 °C/±0,8 °C
- Messbereich Luftfeuchtigkeit: 0–99 %, Auflösung/Toleranz: 1 %/±5 %

Lieferung mit Software auf CD-ROM und USB-Kabel

Laser-Entfernungsmesser C2



- Messdistanz 0,3–60 m, Genauigkeit typ. $\pm 1,5$ mm
- Auch im Freien einsetzbar
- Heller Ziellaser
- Single- und Dauermessung (mit Min./Max.-Wert-Tracking)
- Rechenfunktionen: Addition/Subtraktion, Fläche, Volumen, indirekte Messung (Höhenberechnung)
- Speicher für 20 Messwerte
- Bezugspunkt einstellbar (Vorder-/Rückseite des Geräts)
- Schaltbare Displaybeleuchtung
- Maßeinheit wählbar (m/ft/in)
- Energiesparfunktion – automatische Abschaltung des Messlasers und des Geräts
- Sehr leichte, robuste, spritzwasserfeste Ausführung: IP 54

Abm. (B x H x T): 45 x 121 x 26 mm

Lieferung inkl. Schraubendreher für Batteriefachdeckel, Gürteltasche und 2x Micro-Batterien (AAA/LR03)

WF-10 76 77

€ 74,95

12-V-LED-Vorschaltgerät, 0–60 W



WF-10 75 31

€ 18,95

Kompakter LED-Treiber für 230-V-Betrieb, dimmbar mit Phasenanschnittdimmer, geeignet für Möbel- oder Zwischendecken-einbau. Der LED-Treiber kann im Leistungsbereich bis 60 W auch 12-V-Halogenlampen versorgen.

- Ideal für LED-Betrieb geeignet: keine Mindestlast erforderlich

- Dimmbar mit Phasenanschnittdimmer
- Auch für 230-Vdc-Notstrombetrieb geeignet
- Klein, kompakt, leicht
- IP-Schutzart IP 20

Abm. (B x H x T): 73 x 26 x 36 mm

Labornetzgerät HCS-3102



WF-10 67 65

€ 134,95

Das per USB fernprogrammierbare und per Spannungs-/Widerstandseingang fernsteuerbare Labornetzgerät liefert bis zu 36 Vdc bei bis zu 5 A Ausgangsstrom. Lieferung inkl. Software-CD, Netzkabel und USB-Kabel.

TECHNISCHE DATEN

Eingangsspannung	230 V/50 Hz
Ausgangsspannung	1–36 VDC
Ausgangsstrom	0–5 A
Ausgangsleistung	max. 180 VA
Restwelligkeit (RMS/Spitze-Spitze)	5/50 mV
Netzspannungsausregelung (Lastvariation 0–90 %)	50 mV
Netzspannungsvariation (90–264 VAC)	20 mV
Stromausregelung Last: 10–90 %	100 mA
Anschlüsse	USB-B
Abm. (B x H x T)	200 x 80 x 208 mm
Gewicht	2,4 kg

LED-Module LEDxALUSTAR



IP 65

Anschlussfertige LED-Module

ab

€ 8,95

Die Alustar-Serie überzeugt durch ihre kleine und kompakte Bauform und hohe Leuchtkraft. Setzen Sie gezielt Lichtakzente!

Die anschlussfertigen LED-Module, bestückt mit einer hochwertigen Edison Edixeon-LED, fügen sich nahtlos in jedes Ambiente ein. Eine präzise Linsenoptik ermöglicht eine besonders akzentuierte Beleuchtung.

- Hochwertiges Aluminiumgehäuse für gute Wärmeableitung, durch gute Kühlung hohe Lebensdauer
- Betrieb nur mit geeigneter Konstantstromquelle

Die komplette Alustar-Serie finden Sie unter **Webcode #5004**

Viele weitere Produkte und Informationen finden Sie unter: www.electronic-components.elv.de

Starterkit mit TFT EA EVALeDIPTFT70 800 x 480 Pixel

ELECTRONIC ASSEMBLY
making things easy



WF-10 72 58

€ 399,95

Die umfassenden Starterpakete mit jeweils einem Display inkl. Touchpanel ermöglichen eine schnelle Inbetriebnahme und eine einfache Einarbeitung. Das Evaluation-Board sorgt für kürzeste Entwicklungszeiten. Eine USB-Schnittstelle sorgt für eine direkte Verbindung zum PC. Es ist keine externe Spannungsversorgung notwendig. Das Starterkit ermöglicht einen schnellen Start des im Lieferumfang enthaltenen Displays.

- Typ: EA EVALeDIPTFT70
- Anzahl Zeichen: 800 x 480

- Farbe: 16-Bit-Color
 - Betriebsspannung: 5 V
- Abm. (B x H x T): 170 x 112 x 13,9 mm

Lieferumfang:

- Display inkl. Touchpanel
- USB-Programmierboard bzw. USB-Evaluation-Board
- DVD mit Treibern für Windows, einem Editor, Compiler sowie Demos
- USB-Kabel
- 5x Interfaceboards für RS232, RS485, I²C, SPI und Mikrocontroller RS232

Bestell-Hotline: + 49 (0)491/6008-88

www.elv.de

LED-Weihnachtsbaum

Mit glanzvollen Lichtshow-Effekten, als Single oder Duo!

Von Willem Tak (NL)



Zum stimmungsvollen weihnachtlichen Glanz trägt seit Jahren auch die Elektronik bei. Die Glühlampenketten früherer Zeiten, die höchstens monoton blinken konnten, räumen ihren Platz für

LED-Arrays und Lichtshow-Effekte. Die 56 LEDs dieses stilisierten Weihnachtsbaums werden von einem pfiffigen Programm gesteuert, das in einem unscheinbaren Mikrocontroller steckt. Die Lichtshow dauert länger als eine dreiviertel Stunde, ohne dass sich ein Show-Element wiederholt.

Wieder ist die Jahreszeit gekommen, in der die Menschen unserer Breiten ihre Umgebung vorweihnachtlich dekorieren. Es dauert nicht mehr lange, bis der Weihnachtsbaum ganz oben auf der Einkaufsliste steht. Dieses Symbol der dunklen Jahreszeit wird nur noch selten mit offen brennenden Wachskerzen geschmückt.

Elektrische Kerzen sind schon lange die Favoriten, denn sie lassen sich auf Knopfdruck ein- und ausschalten. Die Gefahr, dass der Baum unversehens Feuer fängt, ist mit hoher Wahrscheinlichkeit gebannt. Die Glühlämpchen elektrischer Kerzen und Lichterketten werden allmählich von

LEDs verdrängt, denn sie sparen Energie, und sie lassen sich vielfältig steuern.

Der Autor hat einen elektronischen LED-Weihnachtsbaum entworfen, der sich von den Massenprodukten des Handels auffällig abhebt. Die steuernde Hardware besteht aus kaum mehr als einer Mini-Platine und einem Mikrocontroller. Die Baum-Platine trägt 56 LEDs, sie ist in Form und Gestalt dem klassischen Weihnachtsbaum nachempfunden. Der Clou des LED-Weihnachtsbaums liegt jedoch weniger in der Hardware, sondern vielmehr in der Software. Der Mikrocontroller steuert die LEDs so virtuos, dass der LED-Weihnachtsbaum zu einem nicht alltäglichen Blickfang

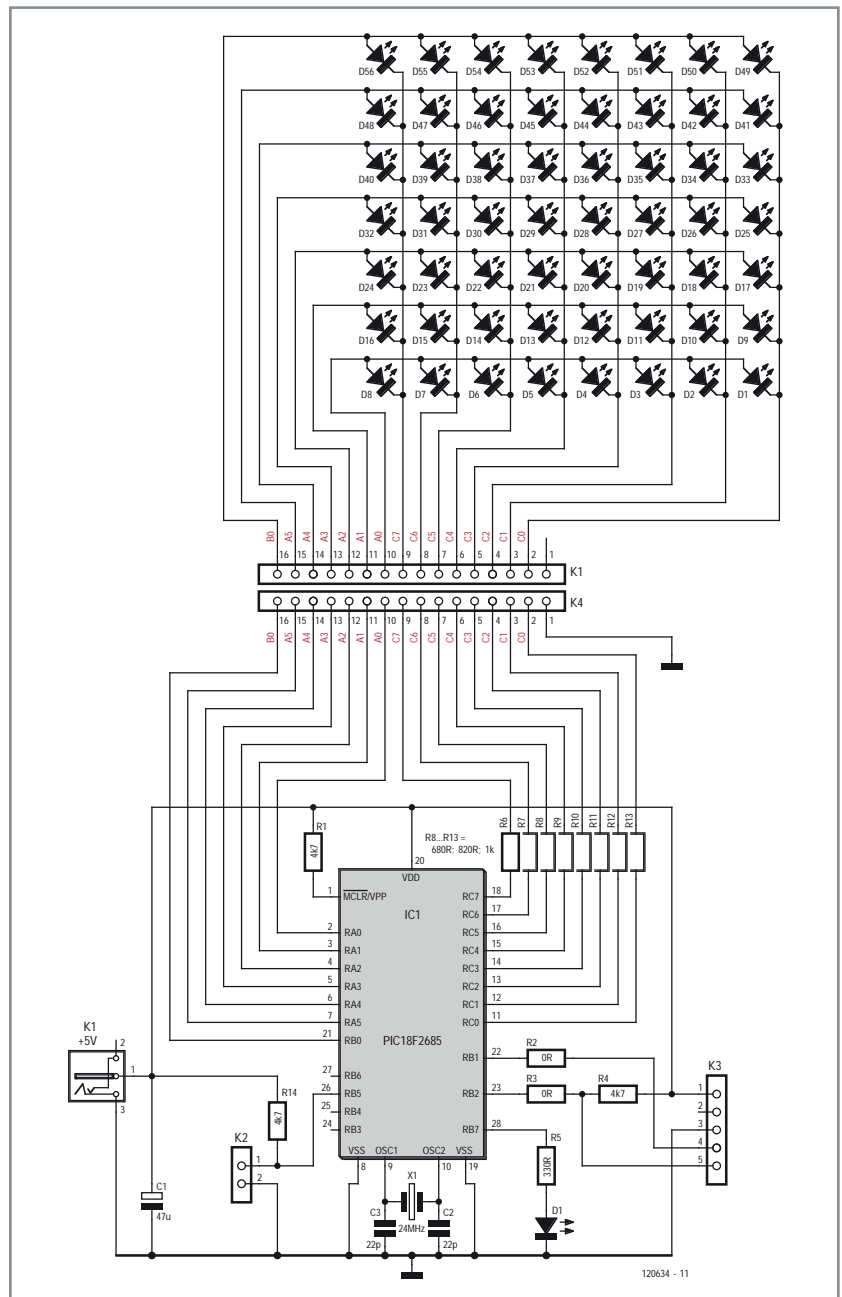
wird. Einen Vorgeschmack auf die Lichtshow-Effekte gibt der Video-Clip auf der LED-Weihnachtsbaum-Projektseite [1]. Werden zwei LED-Weihnachtsbäume miteinander gekoppelt, können die Lichtshow-Effekte zwischen den Bäumen überspringen: Der visuelle Eindruck wird um eine Dimension erweitert! Die Lichtshow besteht nicht einfach aus aneinander gefügten Folgen stets gleicher kurzer Muster, es vergehen mehr als 50 Minuten, bis die nächste Vorstellung beginnt.

Wenig Hardware, viel Speicher

Wenn sich Lichtshow-Effekte schon nach kurzer Zeit stets gleichförmig wiederholen, wirken sie schnell öde und fahl. Um möglichst viele abwechslungsreiche Effekte speichern zu können, hat der Autor für die Steuerung den Mikrocontroller PIC18F2685 von Microchip gewählt, der einen 96 KB großen Programmspeicher mitbringt. Von der mehr als 50 Minuten dauernden Lichtshow wird der Programmspeicher fast vollständig belegt. Die Frequenz des externen Quarzes beträgt 24 MHz, die interne Taktfrequenz 8 MHz hat sich für einige Show-Effekte als zu langsam erwiesen.

Die Schaltung in **Bild 1** ist unkompliziert: Die 56 LEDs sind in einer Matrix angeordnet (8 mal 7 Elemente), die Matrix ist mit dem Mikrocontroller unmittelbar verbunden. Der Mikrocontroller schaltet die Spalten sequentiell über die Anschlüsse A0...A5 und B0 nach +5 V. Die Reihen werden vom Mikrocontroller über Port C und R6...R13 nach Masse gezogen. Von dem Bitmuster an Port C hängt ab, welche LEDs einer Spalte aufleuchten, während die Signale an den Leitungen A0...A5 und B0 die Spalten multiplexen.

Wegen der Multiplex-Steuerung kann jede LED nur $1/7$ der Zeit leuchten, die ein Multiplex-Zyklus dauert. Die Helligkeit ist deutlich niedriger als bei statischem Betrieb über den gleichen Widerstand. Zum Zweck der Beleuchtung ist der LED-Weihnachtsbaum nicht gedacht, er entfaltet seinen Glanz eher in einer Umgebung, die stimmungsvoll gedämpft beleuchtet ist. Der Autor hatte den Prototyp des LED-Weihnachtsbaums mit rundstrahlenden LEDs des Herstellers SBL [2] bestückt. Natürlich sind auch LEDs aus anderen Quellen einsetzbar, sie sollen jedoch einen hohen Wirkungsgrad haben (so genannte „ultrahelle“ LEDs). Die Mikrocontroller-Leitungen sind bis 25 mA belastbar, was wegen des Multiplex-Betriebs einem Strom von nur ungefähr 3 mA für



die einzelne LED entspricht. Wenn die Umgebung völlig abgedunkelt ist, können die LEDs trotzdem zu hell wirken. In diesem Fall kann die Helligkeit durch höhere Werte für R6...R13 (680 Ω , 820 Ω oder 1 k) herabgesetzt werden.

Zwei LED-Weihnachtsbäume sind über den Anschluss K3 zu einem Weihnachtsbaum-Duo koppelbar, was den Effekt hat, dass Lichtkonfigurationen von Baum zu Baum springen. Dazu müssen die Steckverbinder K3 (Leitungen RB1

Bild 1.
Der Mikrocontroller steuert
56 LEDs im Multiplex-
Betrieb. Das Programm
belegt den 96 KB großen
Flash-Speicher des
PIC18F2685 fast vollständig.

Bild 2.
Platine der Steuerung,
sie ist ausschließlich
mit Komponenten für
die Durchsteckmontage
bestückt.

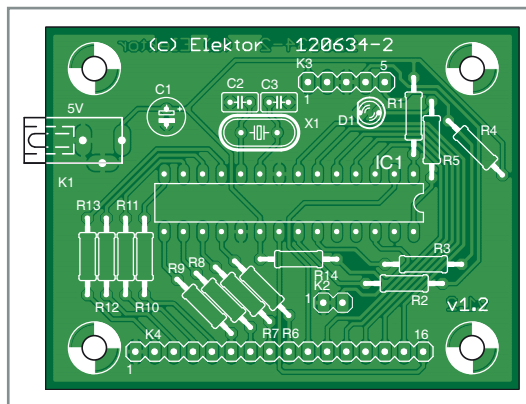
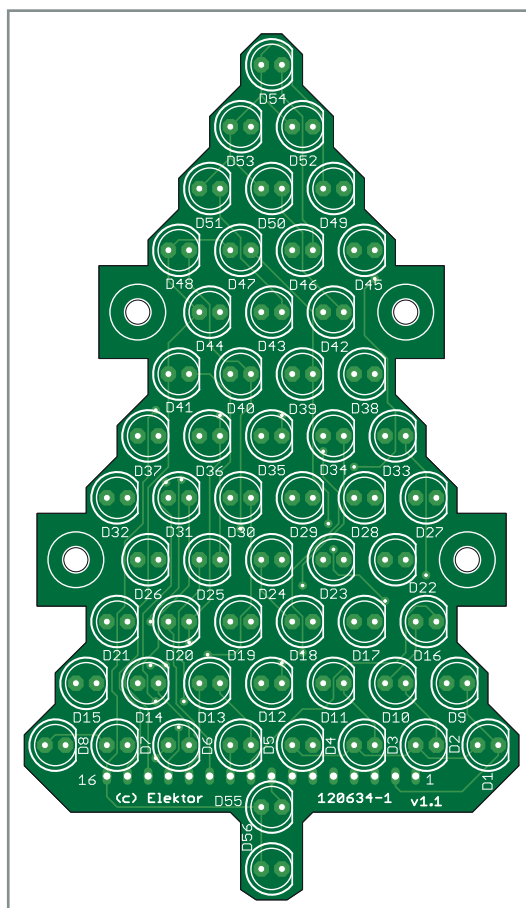


Bild 3.
Auf der doppelseitigen
Platine mit der Silhouette
eines Weihnachtsbaums
haben die 56 LEDs ihren
Platz.



und RB2 der Mikrocontroller) *über Kreuz* miteinander verbunden werden. Die Mikrocontroller-Software existiert in zwei Varianten: Die L-Version ist für den von vorn gesehenen linken Baum bestimmt, die R-Version für den rechten.

LED D1 an Leitung RB7 überwacht den Mikrocontroller, sie kann gegebenenfalls weggelassen werden. Nach Einschalten der Betriebsspannung oder nach einem Reset muss die LED D1 kurzzeitig blinken.

Die Platinenlayouts für den LED-Weihnachtsbaum und seine Steuerung sind in **Bild 2** und **3** abgedruckt, gebrauchsfertige Platinen hat der Elektor-Shop auf Lager (siehe [1]). Die Bestückung ist absolut unkritisch, es sind nur Bauelemente für Durchsteckmontage vorhanden. Die Verbindungen zwischen den Platinen kann ein 16-adriges Bandkabel herstellen, Kabel-Steckverbinder sind nicht unbedingt nötig.

Das Mikrocontroller-Programm steht auf der Elektor-Projektseite [1] zum Download in zwei Versionen bereit: Die L-Version ist für den linken LED-Weihnachtsbaum eines Duos bestimmt, die R-Version wird in den Mikrocontroller des rechten Baums geladen. Wenn sich die Bauaktivitäten auf nur einen Weihnachtsbaum beschränken, ist die Wahl der Version beliebig. Wie schon erwähnt, kann auf der Projektseite ein vom Autor gedrehter Video-Clip betrachtet werden. Auch die Platinenlayouts wurden in gewohntem Format in die Projektseite eingestellt. Wer sich darin versuchen möchte, neue Lichtshow-Effekte zu kreieren, kann den Quellcode des Mikrocontroller-Programms herunterladen. Doch die mehr als 50 Minuten lange LED-Weihnachtsbaum-Lichtshow dürfte fürs Erste kaum Langeweile aufkommen lassen...

(120634)gd

Weblinks

[1] www.elektor.de/120634

[2] www.superbrightleds.com

Stückliste

Mikrocontroller-Platine

Widerstände:

R1,R4,R14 = 4k7
R2,R3 = Drahtbrücke
R5 = 330 Ω (optional)
R6...R13 = 680 Ω (oder 820 Ω oder 1 k, siehe Text)

Kondensatoren:

C1 = 47 μ /16 V

C2,C3 = 22 p keramisch

Halbleiter:

D1 = LED rot, 3 mm (optional)
IC1 = PIC18F2685-E/SP (Microchip)

Außerdem:

X1 = Quarz 24 MHz
K1 = Stromversorgungsbuchse (z. B. CUI PJ-031DH)
K2 = Stiftkontaktleiste 2-polig (optional)

K3 = Stiftkontaktleiste 5-polig
K4 = Stiftkontaktleiste 16-polig (oder kurzes Bandkabel, 16-adrig)
Drucktaster für Betriebsart und Reset (optional)
Platine 120634-1 (siehe [1])

Weihnachtsbaum-Platine

D1...D56 = LED 5 mm ultrahell
K1 = Buchsenkontaktleiste 16-polig (passend zu K4 der Mikrocontroller-Platine)
Platine 120634-2 (siehe [1])



DAS ORIGINAL SEIT 1994
PCB-POOL
Beta LAYOUT

Leiterplatten-Prototypen und kleine Serien

FREE Stencil
bei jeder PCB Prototyp-Bestellung

Easy-going
17 akzeptierte Layoutformate



www.pcb-pool.com

Beta

LAYOUT
create:electronics

PCB-POOL® ist eine eingetragene Marke der Beta LAYOUT GmbH.
Alle eingetragenen Warenzeichen sind eingetragene Warenzeichen der jeweiligen Hersteller!



eSTORE
Beta LAYOUT

Entwickeln, Bestücken und Löten



Arduino Mega (ATMega 1280-16AU)
kompatibel

Reflow-Controller



€ 129,00*

**LED Wechselblinker
SMD-Bausatz**



€ 6,00*

Big Beta-Reflow-Kit



€ 129,00*

Tool-Kit Extended



€ 149,00*

* inkl. MwSt. und zzgl. Versandkosten

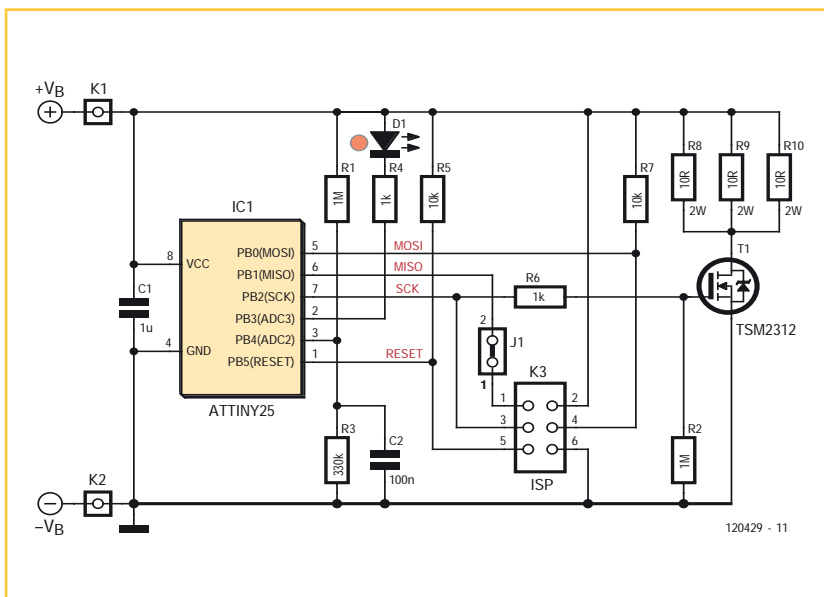
www.beta-eSTORE.com

Beta

LAYOUT
create:electronics

Preisfrage: Warum lässt sich in dieser Schaltung (verflucht!) mit einem AVRISP-MKII-Programmer keine Firmware per ISP flashen?

Darüber grübelte ich viel zu lange nach und probierte dies und das, bis mir endlich ein Licht aufging. Zum Hintergrund: Ich wollte einen möglichst einfachen, genauen und flexiblen Balancer für meine neuen LiFePo-Akkus bauen. Die gibt's zwar auch fertig oder richtig aufwändig mit Spezial-ICs, doch ich bin es schon gewohnt, den einfacheren elektronischen Herausforderungen mit kleinen AVR-Controllern und etwas Code zu begegnen. Gerade den hier verwendeten acht-poligen Typ ATtiny25 habe ich schon so vielfältig eingesetzt (siehe [1] und [2]), dass ich es kaum glauben konnte, dass mir das Teil nun solche Probleme machte. Aber erstens kommt es anders und zweitens als man denkt.



Nach x-fachem Durchmessen der Leiterbahnen und Kontrolle der Bestückung war meine erste Vermutung die: Vom Hersteller war möglicherweise bei den Fuses „SPIEN“ nicht gesetzt und daher kein Flashen via ISP/SPI möglich. Da es sich um die SMD-Variante handelte, konnte ich den Controller nicht einfach mit meinem STK500-Board per „High Voltage“ programmieren. Das direkte Anlöten von Käbelchen an die Platine und die Verbindung mit den HV-Pins des STK500 war erstaunlicherweise erfolglos, obwohl eine HV-Programmierung eigentlich immer klappen muss. Daraufhin lötete ich an einen SMD-Tiny25 kleine Drahtstücke, damit der Controller in eine der DIL-Fassungen auf dem STK500 passte. Das funktionierte und SPIEN war auch korrekt gesetzt. Also steckte der Fehler ärgerlicherweise doch in der Schaltung. Bloß wo?

Nach längerem Sinnieren kam mir die Idee, dass vielleicht die 1 k Ω von R4 in Serie mit der LED D1 beim Programmieren irgendwelche Signale zu stark belasteten. Aber wieder falsch gedacht, denn Pin 2 (PortB.3) ist überhaupt nicht mit K3 verbunden. An dieser Stelle krabbelte die Erkenntnis aus meinem Hinterkopf, dass T1 zwar physikalisch sehr klein ist, aber als recht kräftiger Schalt-MOSFET doch eine ordentliche Eingangskapazität haben könnte. Ein Blick ins Datenblatt bestätigte die Hypothese: $C_{iss} = 900$ pF. Aha!

Also den Gate-Widerstand R6 auf 10 kΩ vergrößert, den AVRISP an K3 angeschlossen, im AVR Studio auf „Program“ geklickt und schon war die Welt der kleinen Mikrocontroller wieder in Ordnung. Und der Preis für die Auflösung der Frage? Wenn Sie meinen Fehler rein durch das Betrachten des Schaltplans und somit schneller als ich gefunden haben, dann dürfen Sie nun (berechtigt) über mich Schmunzeln ;-).

(120429)

- [1] www.elektor.de/jahrgang/2011/januar/i555.1647770.lynx
- [2] www.elektor.de/jahrgang/2008/januar/anti-standby-switch.321531.lynx



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!

Topfgucken

Ist das nicht das neue Elektor-Schaltnetzteil?

Trafo geeignet für 115 und 230-V-Netz ü Trafo-Serie netterweise mit gleichen Abmessungen für unterschiedliche Ausgangsleistungen.

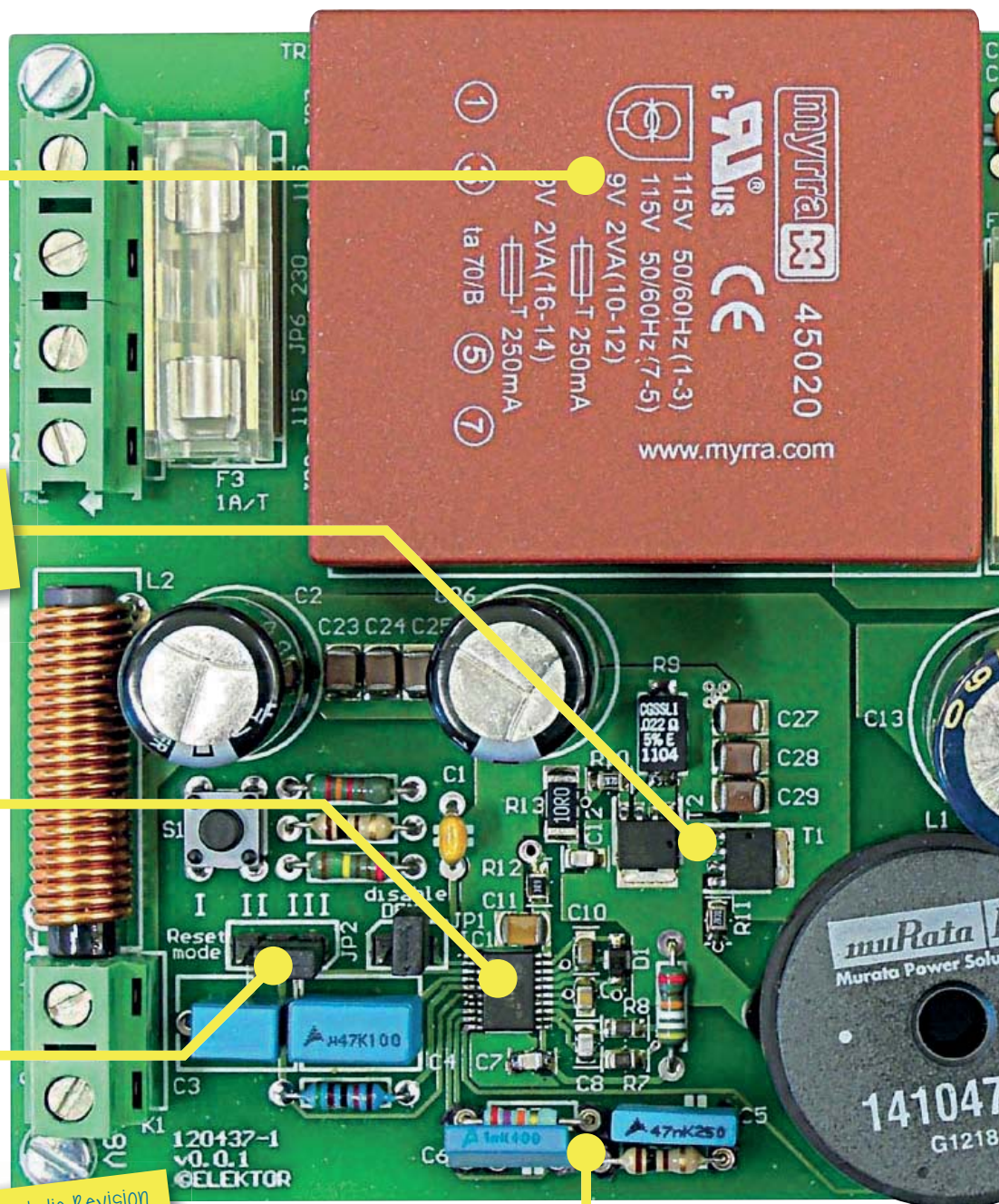
T1, T2 - spezielle Hochstrom-FETs von NXP mit extrem niedrigem $R_{DS(on)}$.

IC1 und IC3 sind von Hand nur schwer zu löten. Es geht schon, aber dann wäre eine Heißluft-Lötstation angebracht. Ein Reflow-Lötofen wäre ideal dafür.

Der Reset-Modus wird mit JP2 eingestellt (hiccup, latch-off, cycle-by-cycle)

Das Board hat die Revision „v0.0.1“.

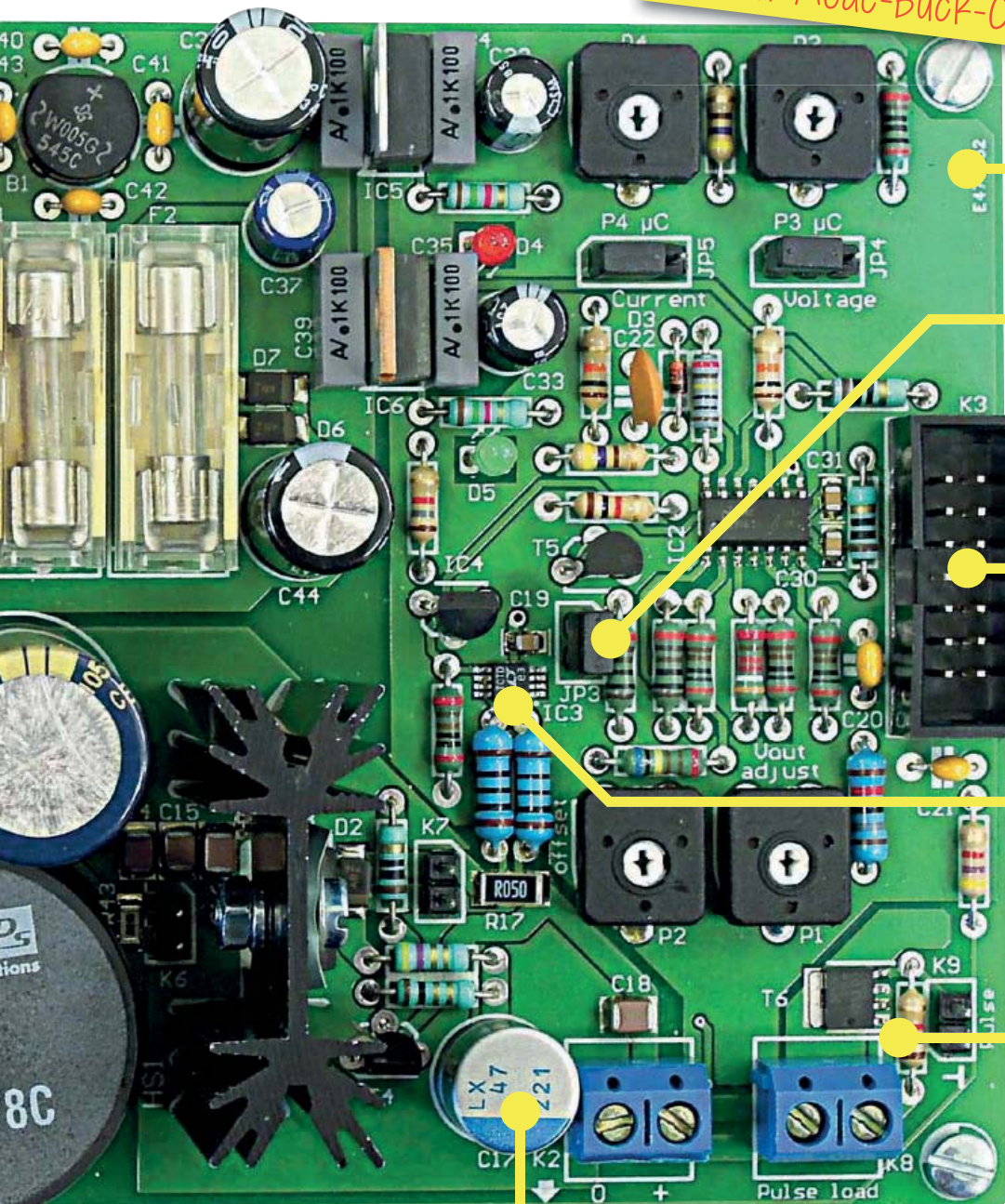
Die Werte für das Feedback-Filter (C5, C6, R5, R28) wurden experimentell ermittelt.



Ein neues Projekt im Elektor-Labor: In der kundigen Hand des Entwicklers Ton Giesberts beginnt ein Schaltnetzteil (SMPSU) langsam Gestalt anzunehmen. Im Moment befindet es sich noch im Konzeptstadium. Ein erster Vorgeschmack:

Von Thijs Beckers (Elektor-Redaktion & Labor)

Im Prinzip ist das Netzteil ein Current-Mode-Buck-Converter.



Doppelseitige Platine.

Mit JP3 kann man zwischen Hardware und/oder Software-Strombegrenzung wählen.

Interface zum Mikrocontroller - das Design und die Software müssen noch erstellt werden.

Sind bei IC3 die NC-Pins verbunden? Muss noch überprüft werden.

T6 - nur übergangsweise zum Test mit gepulster Last implementiert.

Die Anforderungen an C17 werden nur von einem Kondensator von Nichicon eingehalten. Wie steht es um die Erhältlichkeit? ESR-Messungen an C17 waren unbefriedigend ($0,4 \times \Omega$). Das gewöhnliche Messverfahren brachte zunächst unklare Resultate. Dann zeigte sich, dass da wohl eine Masseschleife zwischen den Messgeräten existierte, die für die Verwirrung sorgte.

Massefläche weg?

Von Thijs Beckers (Elektor-Labor)

Nicht alles lief reibungslos mit dem Flugfunkempfänger, den unser altgedienter Entwickler Luc in der Mache hatte. Verschiedene Bauteile, die der Autor in der Schaltung verwendete, waren zwar

Platine mit PCB-Visualizer testen!

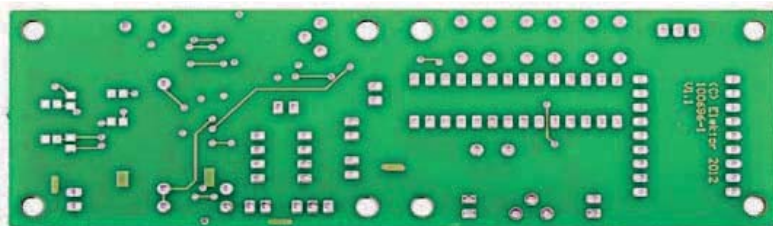
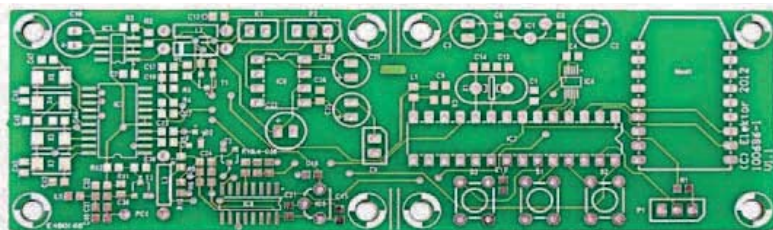
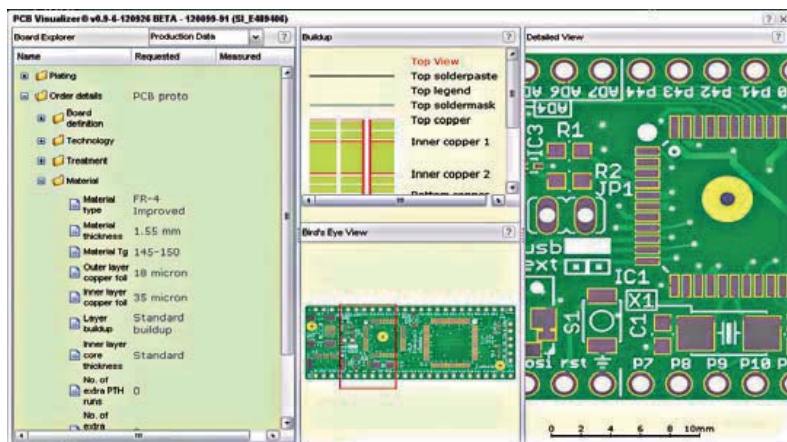
bei einigen Händlern im Angebot, wurden aber längst nicht mehr produziert. Um die Nachbaueignung auch in fernerer Zukunft sicherzustellen, mussten diese obsoleten Bauteile durch aktuelle Typen ersetzt werden. Die Beschaffung der

neuen Bauteile, die Anpassung der Platine mit den neuen Komponenten für den Prototyp, all das verlief ohne die geringsten Probleme. Bis das Päckchen mit der Proto-Platine im Labor eintraf... Irgendwann im Laufe des Designvorgangs mit unserer CAD-Software-Suite Altium Designer hatte Luc die oberen und unteren Masseflächen „shelved“ (versteckt), um einen besseren Überblick der gerouteten Signale zu gewinnen. Nach der Fertigstellung des Boards am Computer generierte Luc die Gerber-Dateien für das Proto-Board ohne die geringste Fehlermeldung wie gewöhnlich als *.outjob*-File und schickte sie zu unserem langjährigen Platinenlieferanten Eurocircuits [1]. Das böse Erwachen kam dann mit der Post. Die versteckten Masseflächen waren tatsächlich auch nicht auf der Platine vorhanden, das gute, nein, schlechte Stück war völlig nutzlos, da viele, wenn nicht alle Masseanschlüsse der Bauteile über die Masseflächen verliefen. Erst im Nachhinein erinnerte sich Luc an das Verstecken der Masseflächen. Wie es scheint, ist alles, was versteckt ist, auch nicht im Gerber-File vorhanden. Man hätte die fehlerhafte Produktion aber leicht ausschließen können, wenn man vorher mit einem Gerber-Betrachter einen virtuellen Blick auf die Platine hätte werfen können. Hätte und könnte, Zeitdruck und Ablenkung durch andere „Tasks“ ließen Luc diesen finalen Schritt vergessen!

So etwas scheint bei Entwicklern öfter vorzukommen, denn Eurocircuits bietet einen neuen Service namens „PCB Visualizer“ (siehe Screenshots), der online alle hochgeladenen Layouts im Extended-Gerber- oder CadSoft-EagleV6-Format akzeptiert und das nach der Produktion zu erwartende Layout im Detail in einem Fenster darstellt, begleitet von einer Vielzahl von Parametern. So können alle Produktionsfehler von vornherein ausgeschlossen werden.

Nach der Bestätigung, dass alles so aussieht wie gewünscht, wird die Platine bestellt. Dann sitzt man da und wartet auf das Päckchen, mit ruhigem Gewissen und in der Sicherheit, dass alles so eintrifft wie es sein soll, ohne fehlende Leiterbahnen, Kupfer-Gitter oder -Polygone.

(120568)



[1] www.eurocircuits.com



DESIGNSPARK

NEW. FREE. **DESIGNSPARK PCB** **VERSION 4**

COMPREHENSIVE NEW LIBRARIES.
EASIER COMPONENT SELECTION & QUOTATION.
NEW PCB PROTOTYPING SUPPORT.

Discover today at
www.designspark.com

UNIQUE
RESOURCES BY 

Bauelemente-Tipps

Von Raymond Vermeulen (Elektor-Labor)

Temperaturabhängige Ventilatorsteuerung

Diesen Monat stehen zwei ICs im Fokus des Interesses, mit denen man eine kompakte Lüfterregelung realisieren kann. Sie eignen sich für Anwendungen, bei denen in einem Gehäuse so viel Wärme abgegeben wird, dass eine aktive Kühlung per Ventilator unumgänglich ist. Wenn man den Ventilator aber temperaturabhängig steuern möchte und kein Mikrocontroller zur Verfügung steht, der diese Aufgabe noch mit übernehmen könnte, dann ist hierfür ein extra IC notwendig. Im Idealfall muss man an dieses nur einen Temperatursensor und den Lüfter anschließen. Die hier vorgestellten ICs umgehen den bei modernen PC-Boards üblichen vierpoligen Stecker mit 25-kHz-PWM-Steuerung. Sie schalten direkt die Betriebsspannung mit deutlich niedrigeren Frequenzen.

(120569)

MIC502

Beim IC MIC502 von Micrel dient ein NTC oder PTC als Temperatursensor. Es kann sogar noch ein zweiter Sensor angeschlossen werden. Die Sensoren kommen an die Pins VT1 und VT2. An diesen Eingängen hat eine Spannung zwischen 30 % bis 70 % von V_{dd} eine Impulsbreite von 0 % bis 100 % am Ausgang zur Folge. Der Eingang mit der höheren Spannung hat Priorität. Das eignet sich sehr gut, um mit einem Sensor die Umgebungstemperatur zu erfassen und mit dem anderen ein Bauteil mit Verlustleistung. Durch eine Spannung an Pin V_{slp} wird das IC in den Ruhezustand versetzt, wenn die Eingänge VT1 und VT2 unter diesen Wert fallen. Erst wenn VT1 oder VT2 eine höhere Spannung haben, wird das IC wieder reaktiviert. Mit diesem Feature kann man verhindern, dass der Lüfter mit einer zu niedrigen PWM läuft. An Pin CF kommt ein zeitbestimmender Kondensator, für den ein Wert von 100 nF für eine Frequenz von 30 Hz empfohlen wird. Mit niedrigeren Werten kann man aber auch höhere Frequenzen erreichen. Laut Datenblatt sind Frequenzen von 15 bis 90 Hz zulässig.

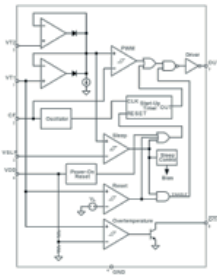


Bild 1. Blockschaltung des MIC502.

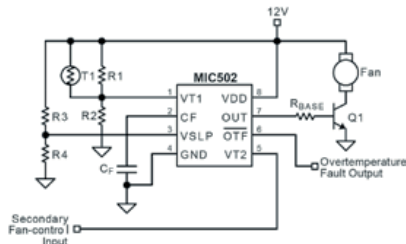


Bild 2. Prinzipschaltung mit dem MIC502.

Datenblatt MIC502: www.micrel.com/_PDF/mic502.pdf

MAX6643

Bei MAX6643 dient als Sensor ein als Diode geschalteter Transistor. Es gibt drei verschiedene Einstellmöglichkeiten für Temperaturen. Mit „Overtemperature Threshold“ (Pins OT1 und OT2) wird eine Temperatur zwischen 60 und 100 °C eingestellt, bei deren Überschreitung der Ausgang OT auf Masse gezogen wird. Eine „High Temperature Threshold“ (Pins TH1 und TH2) bestimmt, bei welcher Temperatur der Duty-Cycle des PWM-Signals höher eingestellt wird. Die „Low Temperature Threshold“ (Pins TL1 und TL2) bestimmt umgekehrt, bei welcher Temperatur dieser Duty-Cycle reduziert wird.

Mit den je zwei Eingängen kann man über die Pegel „low“, „high“ oder „open“ jeweils $3^2 = 9$ Temperaturen einstellen (siehe Datenblatt). Mit dem Pin „FULLSPD“ kann man einen Duty-Cycle von 100 % erzwingen. Das ist nützlich, wenn eine zu hohe Temperatur bemerkt wird. Außerdem gibt es noch Eingänge für das Tachosignal des Lüfters. Dessen Ausfall wird am Ausgang „FANFAIL“ signalisiert.

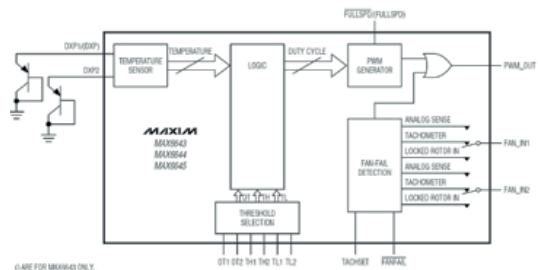


Bild 3. Blockschaltung des MAX6643.

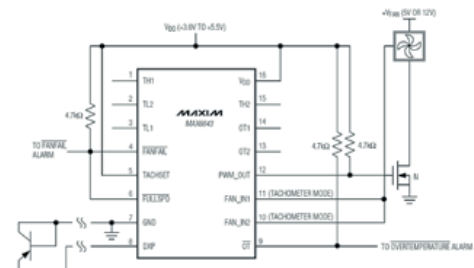




Bild 4. Prinzipschaltung mit dem MAX6643.

Datenblatt MAX6643: <http://datasheets.maximintegrated.com/en/ds/MAX6643-MAX6645.pdf>

PCBs Muuuuch Cheaper...

PCBs starting from **17.22 EURO***

* incl. VAT (23%), add delivery costs
(Example Germany) of 10.89 EURO,
min. number of ordered pcs:
5 @ 100 mm x 100 mm

No-frills policy

www.jackaltac.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



powered by Eurocircuits

25% Rabatt auf alle neuen Elektor-Platinen!

Der Elektor-PCB-Service gewährt auf jede neue Elektor-Platine (ab Erscheinungsdatum) 90 Tage lang 25% Rabatt!

Elektor-Platinen jetzt unter www.elektor.de/pcb zum Einführungspreis bestellen und 25% sparen!

Arduino auf Kurs (3b)

Teil 3b: Android-Smartphones & externe Sensoren

Von
David Cuartielles
(Spanien) und
Andreas Goransson
(Schweden)

Letzten Monat stellten wir eine Reihe von Tools für die Arbeit mit Android und dem Arduino Mega ADK vor. Diesen Monat gehen wir noch einen Schritt weiter und zeigen, wie man mit einem Smartphone Sensoren auslesen kann, die an Arduino angeschlossen sind. Wenn alle Entwicklungs-Tools am richtigen Platz sind, dann sind die Ausführung von Code im Smartphone und die Kommunikation mit Arduino sehr einfach. Man kann direkt mit den Beispielen loslegen, die bei den ADK-Libraries dabei sind!



Android-Geräte findet man mittlerweile fast überall, und es gibt viele Wege, um Apps dafür zu erstellen. In unserem Fall beschäftigen wir uns weiter mit Processing [1] als einem Weg, um Code zu schreiben, zu compilieren und auf Android-Smartphones zu laden.

Materialien

Um die in der letzten Folge begonnenen Experimente fortzusetzen, werden folgende Materialien benötigt:

- Entweder ein Arduino Mega ADK oder Arduino Uno mit USB-Host-Shield.
- ein Smartphone mit Android 2.3.4 oder neuer (je neuer desto besser),
- ein Standard-USB-Kabel und ein Micro-USB-Kabel,
- ein TinkerKit-Shield (optional, eine Experimentierplatine mit Steckbrücken reicht) und
- ein TinkerKit-Modul für Potentiometer und Taster (ebenfalls optional).

Hinweis: Um die Inhalte dieses Artikels nachvollziehen zu können, benötigen Sie zwingend ein Android-Smartphone. Wir haben uns viel Mühe gegeben, dass der Code auch mit der neuesten Android-Version (im Moment 4.1 „Jelly Bean“) kompatibel ist.

Tipp: Laden Sie sich den kompletten Code von der Elektor-Webseite zu diesem Artikel herunter, da hier nur Ausschnitte davon abgedruckt sind.

Der status quo

Nach der letzten Folge sollte ein gewisses Verständnis von Googles Accessory-Mode und auch davon vorhanden sein, wie Accessories den Android-Geräten mitteilen, beim Anschluss eine bestimmte App zu starten. Außerdem sollte man Processing (Java Learning Tool) kennen und damit kleine Apps erstellt und auf ein Smartphone oder Tablet übertragen haben.

Jetzt geht es um die Erstellung neuer Apps, welche die Verbindung zwischen Arduino (als Accessory) und Android nutzen.

Die USB-Host-Library

Das Android Open Accessory Protokoll (AOAP) bestimmt, wie Android und Arduino miteinander kommunizieren. Ein Arduino-Board ist ein USB-Host, während das Smartphone USB-Client spielt. Android-Geräte beherrschen von Haus aus das so genannte „USB On The Go“. Von daher kann der gleiche USB-Anschluss (hier eine Micro-USB-Buchse) als USB-Client (wie eine Maus oder

Tastatur) oder USB-Host (wie ein PC) fungieren. Das AOAP sorgt dafür, dass Accessories Host spielen und Android-Geräte Client. Da der Host den Client auch mit Strom versorgt, wird das Smartphone vom Arduino-Board geladen – und anderes mehr.

Um eine Kommunikationsmöglichkeit einzurichten, muss man wie in der vorherigen Folge die USB-Host-Library als Teil des Arduino-Codes einbinden. Man kann eines der Beispiele zur Library ausprobieren, um zu überprüfen, ob diese installiert ist.

Hinweis: Per Menü navigiert man zu: „File/Examples/USB Host“. Wenn man hier keine Beispiele findet, dann hat man bei seiner Arduino-Umgebung auch keine USB-Host-Library installiert. Dann führt der Weg über den Download [3] und die Installation, denn die Library wird noch gebraucht. **Bild 1** zeigt die passende Hardware.

Grundlegendes: Processing

Zunächst die Software, die zur Programmierung von Android-Apps benötigt wird:

- Android SDK mit allen Updates bis zur letzten Version der APIs,
- Processing 2.0b3 oder neuer (in der letzten Folge stand, dass wir die neueste Version verwenden – Sie sollten das auch tun),
- das Arduino ADK Tool für Processing und
- die Arduino ADK USB Library für Processing.

Der letzte Punkt der Liste fehlte in der vorhergehenden Folge dieser Artikelserie - er wurde dort absichtlich ausgelassen.

Auf der Seite von Processing benötigt man die zum USB-Host komplementäre Library, damit das Board mit dem Smartphone kommunizieren kann. Diese Library nennt sich „Arduino ADK USB“. Der folgende Abschnitt erklärt, wie man sie installiert. **Bild 2** zeigt die entsprechende Hardware-Seite.

Installation der Arduino ADK USB Library für Android

Im heruntergeladenen Archiv zu diesem Artikel steckt ein Verzeichnis namens „Processing“, das ein Unterverzeichnis namens „libraries“ enthält. Sein Inhalt sollte in den Libraries-Ordner von Processing kopiert werden. Letzterer findet sich im Sketchbook-Ordner von Processing, das üblicherweise über „.../Eigene Dokumente/Processing“ bei Windows oder via „.../Dokumente/Processing“



Bild 1.
Arduino Mega ADK mit
TinkerKit-Shield und
Smartphone.



Bild 2.
Arduino Mega ADK,
angeschlossen an einen PC
und ein Smartphone.

in OS X oder Linux zu finden ist.

Die Library bringt vier Beispiele mit. Wir empfehlen das Starten eines solchen Beispiels mit Ihrem Accessory, da dies das Leben sehr viel einfacher macht. Zu Anfang probiert man am besten einmal das Beispiel „digitalWrite“ aus. In diesem Fall sollte beim Anschluss eines Smartphones an das Arduino-Board eine LED aufleuchten, wenn man den Bildschirm berührt. Die LED erlischt, wenn man den Finger wieder wegnimmt.

DigitalWrite

Jedes Beispiel hat zwei Teile. Eines läuft unter Arduino und übernimmt die Kommunikation mit dem Smartphone, während Aktionen bei den Pins des Boards stattfinden. Der zweite Teil besteht aus mit Processing compiliertem Code, der im Android-Gerät läuft. Diese App kann Daten per Kabelverbindung hin und her schicken sowie einige Aktionen auf dem Bildschirm ausführen oder aber Sensoren wie Kamera, Beschleunigungssensoren etc. auslesen.

Zunächst zum Arduino-Code: Man öffne die Arduino-IDE und tippe den Code von **Listing 1** ein. Nachdem man fertig ist, sollte man nicht ver-

Listing 1. Beispiel DigitalWrite von der Arduino USB Host Library.

```
#include <AndroidAccessory.h>

// accessory descriptor. It's how Arduino identifies itself to Android
char accessoryName[] = "DigitalWrite"; // your Arduino board
char companyName[] = "Arduino SA";

// led variables
int ledPin = 10;

// counters
long timer = millis();

// initialize the accessory:
AndroidAccessory usb(companyName, accessoryName);

void setup() {
    // start the connection to the device over the USB host:
    usb.begin();
    // configure the LED pin as output
```

gessen, den Code auf das Board zu übertragen. Der Code dieses Beispiels lauscht am USB-Port. Wenn das Smartphone via USB mit Arduino verbunden wird, gibt die Funktion „usb.isConnected()“ eine positive Antwort. Wenn das Smartphone Daten sendet, dann gibt „usb.availa-

ble()“ einen von Null verschiedenen Wert zurück, der in der Variable „val“ abgelegt wird. Bei Ihnen sollte Ähnliches wie in **Bild 3** passieren.

Auf der Seite von Processing benötigt man dazu ein Programm, das den Event auswertet, wenn man den Bildschirm berührt. Wenn es via USB

Listing 2. Processing-Sketch zur Erkennung von Bildschirm-Berührungen.

```
import cc.arduino.*;

// create the ADK object
ArduinoAdkUsb arduino;

void setup() {
    // Lock PORTRAIT view
    orientation( PORTRAIT );
    // initialize the ADK object
    arduino = new ArduinoAdkUsb( this );
    if ( arduino.list() != null )
        arduino.connect( arduino.list()[0] );
}

void draw() {
    // Draws a filled rect based on arduino connection state
    connected( arduino.isConnected() );
}
```

```
pinMode(ledPin, OUTPUT);
}

void loop() {
  // print to USB 10 times per second
  if(millis()-timer>100) {
    if (usb.isConnected()) { // is the USB connection open?
      if (usb.available() > 0) { // is there data?
        char val = usb.read();
        // 'a' turns the LED on, 'b' off
        if( val == 'a' )
          digitalWrite( ledPin, HIGH );
        else if( val == 'b' )
          digitalWrite( ledPin, LOW );
      }
      timer = millis();
    }
  }
}
```

angeschlossen ist, sollte das Smartphone ein „a“ als Hinweis senden, dass der Bildschirm berührt wurde. Ein „b“ signalisiert, dass der Finger vom Bildschirm genommen wurde. Es sollte also lediglich auf die Events „start“ bzw. „stop touching“ reagiert werden, damit man die Datenverbindung

nicht mit Meldungen überflutet.

Man öffnet die Processing-IDE und sucht das Beispiel „adk_digitalWrite“. Man findet es per Menü „Examples/Contributed Libraries/Arduino-ADKUsb“ – wenn die Library korrekt installiert wurde (siehe **Listing 2**).

```
public boolean surfaceTouchEvent(MotionEvent event) {
  if ( arduino.isConnected() ) {
    if ( event.getAction() == MotionEvent.ACTION_DOWN )
      arduino.write('a');
    else if ( event.getAction() == MotionEvent.ACTION_UP )
      arduino.write('b');
  }

  // if you want the variables for motionX/motionY,
  // mouseX/mouseY etc.
  // to work properly, you'll need to call
  // super.surfaceTouchEvent().
  return super.surfaceTouchEvent(event);
}

[...]
```

Der Code von Listing 2 sollte selbsterklärend sein. Lediglich die Methoden-Definition von `surfaceTouchEvent(MotionEvent event)` dürfte Ihnen merkwürdig vorkommen. Java ist eine sehr leistungsfähige Programmiersprache und erlaubt hier das Überschreiben existierender Callback-Methoden. Das System ist so konfiguriert, dass bei jeder Berührung des Bildschirms die Funktion „`surfaceTouchEvent(MotionEvent event)`“ aufgerufen wird. Ändert man dies im Programm (indem man z.B. die Funktion überschreibt oder ändert), dann kann man hier Aktionen nach Wunsch ausführen lassen.

Hier geht es darum, als Folge des Events „`MotionEvent.ACTION_DOWN`“ (Äquivalent zu Anfang der Bildschirmberührung) ein „a“ und auf den Event „`MotionEvent.ACTION_UP`“ (Äquivalent zu Ende der Bildschirmberührung) ein „b“ an Arduino zu senden.

Spaß muss sein

Wenn das erste Beispiel läuft, ist es an der Zeit etwas zu bauen, das mehr Spaß macht. Das Internet ist bekanntlich voller Katzen. Das ist der Grund, warum in der letzten Folge die Elektor_Miau-Applikation vorgestellt wurde. Das sollte ein kleiner Spaß sein, der zugleich vermittelt, dass man etwas Netteres als das öde „Hello World“ als erstes Beispiel verwenden kann.

In dieser Folge setzen wir noch einen drauf und zeigen, wie man mit Processing kleine Animationen erstellen kann. Man kann sich hierzu ein animiertes GIF aus dem Internet saugen und damit nachvollziehen, was in diesem Beispiel zu sehen ist. Die einzige Bedingung ist, dass das animierte GIF in Form einzelner Frames exportiert wird, die nach der Konvention `s00.gif`, `s01.gif`, `s02.gif` etc. benannt werden sollten.

Selbstverständlich können Sie aber auch einfach das von uns verwendete Bildmaterial einsetzen. Für diesen Zweck haben wir extra das aus einem Hut erscheinende Kaninchen zeichnen lassen. Unser zweites Beispiel ist ein kleines interaktives Spiel, das einen Taster und ein Poti als externe Sensoren benutzt, um mit dem Kaninchen zu interagieren (siehe **Bild 4**).

Nun folgt der Code, der auf dem Smartphone laufen soll. Zunächst wird mit Processing eine Animation erstellt. Wer sich mit Android schon etwas auskennt, der kann ja einmal versuchen, die in der letzten Folge verwendete GIFdecoder-Library zu portieren, um animierte GIFs mit einer Android-App abzuspielen. Besser ist aber eine

einfache und verständliche Programmierung. Die empfohlene Methode zur Animation des Kaninchens ist wirklich simpel: Man kreiert ein Array aus Bildern, welche den Frames der Animation entsprechen. Dieses Konzept fußt auf dem Processing-Beispiel „Animated Sprite“ von James Paterson [2], siehe auch **Listing 3** [3].

Hinweis: Das Listing 3 zeigt nicht die Animations-Klasse, welche die Bilder mit den Bezeichnungen `s00.gif`, `s01.gif` etc. öffnet. Sie können aber gerne den kompletten Code im Beispiel „Elektor_Simple_Animation“ inspizieren. Im Processing-Sketch-Verzeichnis steckt der Ordner „data“, der alle 54 Bilder der Animation enthält.

Das ultimative Setup

Als nächstes wird das Beispiel so modifiziert, dass es die Library zur USB-Kommunikation nutzt. Am besten verknüpft man den bisherigen Code mit dem Beispiel „`adk_analogRead`“. Den resultierenden Code kann man in **Listing 4** [3] anschauen.

Es gibt noch einige erwähnenswerte Dinge in diesem Beispiel: Zuerst wäre da das Array „`delays[i]`“ zu nennen. Es steuert die Zeit, die jeder Frame der Animation zu sehen ist. Bei jedem neu geladenen Frame sollte man den Timer-Wert so anpassen: `millis() - timer > delays[index]`. Sie können gerne mit den im Array enthaltenen Zeiten experimentieren und schauen, wie sich das auf die Animation auswirkt. Das Array muss so viele Zeitintervalle wie Frames der Animation enthalten.

Weiter gibt es noch das Array „`keyFrames[i]`“, das eine Liste der Frames enthält, bei denen eine Aktion des Benutzer erwartet wird (damit die Animation weiter läuft). Unter Aktionen werden hier Betätigungen von Tastern verstanden, die wie in **Bild 5** an das TinkerKit-Shield angeschlossen sind.

Wenn man wissen will, wie viele Bytes vom Arduino-Board abgeschickt wurden, dann kann man dies mit einer Funktion überprüfen. Mit „`arduino.available() >= 3`“ lässt sich testen, ob drei Bytes angekommen sind. Die Information ist wie folgt codiert:

- Byte 0 hat als Marker des Block-Anfangs den Wert 255,
- Byte 1 ist die Stellung des Potentiometers

- mit Werten von 0 bis 254 und
- Byte 2 enthält den Zustand des Tasters mit Werten von 0 oder 1.

Der Wert des Potis wird dazu verwendet, die Hintergrundfarbe des Bildschirms zu verändern. Der Taster startet die Animationssequenz.

Hinweis: Mit dem Arduino ADK USB-Tool im Menü „Tools/Arduino ADK“ der Processing-IDE kann man das Programm auf das Android-Gerät laden. Hierzu poppt ein Dialog auf. Im Textfeld „model“ gibt man den Namen der App an. Der letzte Parameter dient zur Auswahl der Version des SDK für den Accessory-Typ. Wenn auf Ihrem Gerät Android 3.0 oder neuer läuft, sollten Sie SDK v12 einsetzen, denn SDK v10 ist nur für Android 2.3.4 passend. Falls Ihr Android-Gerät tatsächlich noch mit Version 2.3.3 oder älter bestückt sein sollte, kann es den Accessory-Mode leider nicht nutzen. Der Arduino-Code ist praktisch eine Wiederholung des Beispiels „`adk_analogRead`“ mit ein paar kleinen Änderungen (Marker-Byte, Begrenzung der Werte auf 254, Taster-Byte). Die Programmschleife ist in **Listing 5** [3] zu finden. Beachten Sie bitte auch das komplette Programm in der Dokumentation im Ordner „Arduino/code“ des Downloads von der Elektor-Webseite [3].

Schlusswort

Mit dieser Folge ist die Einführung in Android und seiner Verbindung mit Arduino abgeschlossen. Sie sollten nun in der Lage sein, neue Projekte zu erstellen, zum Beispiel zur Steuerung eines Roboters mit dem Smartphone. Oder aber Sie rüsten externe Sensoren nach, über die Ihr Android-Gerät leider nicht verfügt. Wenn Sie coole Projekte mit unserem Code auf die Beine stellen, wären wir erfreut, davon via „blog@arduino.cc“ zu erfahren. Wir würden dann gerne den Rest der Community informieren.

(120573)

Weblinks

- [1] Processing-Projekt: <http://processing.org>
- [2] Animationen aus mehreren Bildern mit Processing:
<http://processing.org/learning/topics/sequential.html>
- [3] www.elektor.de/120573



Bild 3.
Die LED leuchtet bei Berührung des Bildschirms.

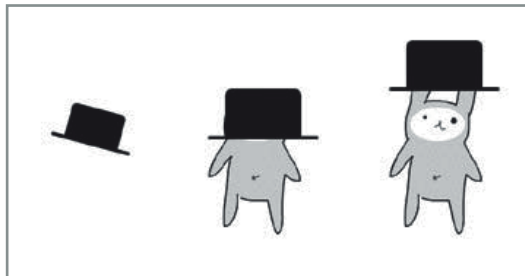


Bild 4.
Einige Frames der Kaninchen-Animation.

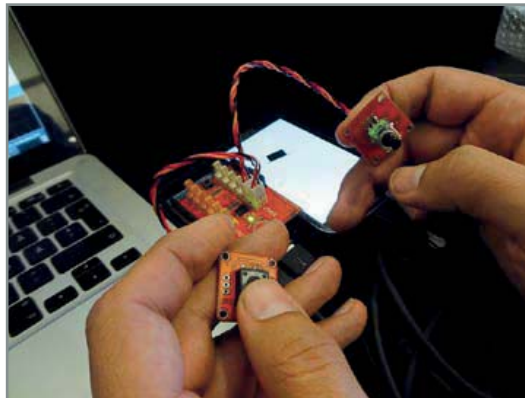


Bild 5.
Taster und Poti am TinkerKit-Shield.

Dank

Das Arduino-Design-Team dankt dem Team von Circuits@Home für die ersten Arbeiten an der USB-Host-Library für Arduino und speziell Philip Lindsay für sein Debugging der initialen Sammlung der offiziellen Arduino-Libraries für Android. Ein Dankeschön geht auch an Rodrigo Calvo für seine Hilfe bei der Verbesserung der USB-Host-Library bei Android-4.1-Geräten und an Laura Balboa für ihre Illustrationen.

Embedded Linux leicht gemacht (6) Netzwerk und Server-Dienste

Von Benedikt Sauter
[1]

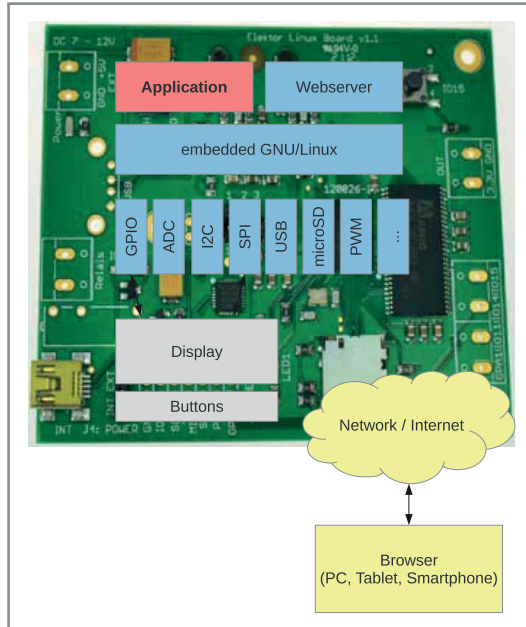


Bild 1.
Übersicht der Anwendung.

Vom Bootloader über den Kernel bis hin zur Installation von Gerätetreibern sind in dieser Artikelserie die wesentlichen Linux-Themen behandelt worden. Bis jetzt fehlt aber noch ein Grundkonzept, mit dem man eine größere Anwendung realisieren könnte; beispielsweise mit einer „Logik“ im Hintergrund, einer Weboberfläche für die Konfiguration und Bedienung und einer einfachen Bedieneinheit mit Display und Tastern.

Nachrichten zwischen Anwendungen

In **Bild 1** ist ein typisches Szenario dargestellt. In der Mitte befindet sich das Linux-System mit der eigenen Anwendung und dem Webserver, um die Oberfläche auf den verschiedensten Geräten im Browser darstellen zu können. Optional sind noch ein Display und ein paar Taster vorhanden. Für die Ansteuerung des Displays könnte eine eigene Anwendung zuständig sein. Die eigentliche Hauptanwendung ist jetzt nur noch ein Prozess, der neben vielen weiteren im Hintergrund läuft. Unter Linux nennt man eine solche Anwendung einen *Daemon*

Das Baukasten-System Linux bietet sich perfekt als Plattform für Komplettlösungen im Bereich Messen, Steuern und Regeln an. Eine maßgeschneiderte Steuerung kann man in der Programmiersprache der Wahl schreiben, wobei sich die Benutzer-Oberfläche per Webserver zu einem PC, Tablet oder Smartphone schicken lässt. In dieser Folge binden wir das Elektor-Linux-Board in ein typisches Heim-Netzwerk ein. Das Schreiben einer eigenen Server-Anwendung und die Skriptsprache Lua sind weitere Themen.

oder Dienst (bzw. Serverdienst). Einmal gestartet, arbeitet diese ohne Eingriff durch den Benutzer. Die Schnittstelle zu weiteren Anwendungen (wie z.B. der Weboberfläche) kann auf verschiedenen Wegen verwirklicht werden (in der Informatik fasst man diesen Austausch zwischen einzelnen Programmen bzw. Prozessen und Threads unter dem Thema Inter-Process-Communication/IPC zusammen). Zur Auswahl stehen Pipes, Fifos, Shared Memory, Sockets, die Standard-Aus- und Eingabe, Dateien, Datenbanken und weitere Hilfsprogramme. Was tatsächlich zum Einsatz kommt, hängt von verschiedenen Faktoren ab. Sind die Anwendungen in der gleichen Programmiersprache geschrieben? Muss die Software auch auf anderen Betriebssystemen laufen können? Welche Datenrate wird bei der Kommunikation zwischen den Anwendungen benötigt?

Eine sehr verbreitete Methode sind klassische Netzwerk-Sockets. Solche Sockets haben den großen Vorteil, dass die Anwendungen sogar über das Netzwerk hinweg Daten und Kommandos

Embedded Linux leicht gemacht

austauschen können. Mit Hilfe eines Sockets kann man sehr einfach Nachrichten senden und empfangen, in den verschiedensten Programmiersprachen stehen hierfür Funktionen und Klassen zur Verfügung.

Wir entscheiden uns daher ebenfalls für Sockets. Da das Surfen im Internet auch auf Standard-Sockets basiert, lernen wir zudem alle Bordmittel kennen, um direkt auf Daten im Internet zugreifen zu können. Aus diesem Grund werden wir im nächsten Schritt eine Internetverbindung für unser Linux-Board einrichten.

Verbindung in das Internet

Voraussetzung ist ein Router im Netzwerk, der eine Verbindung zum Internet hat und diese als Gateway anderen Teilnehmern zur Verfügung stellt. Natürlich müssen auch ein USB/LAN-Adapter an die USB-Schnittstelle des Boards angeschlossen und die Treiber für den Netzwerk-Adapter geladen sein [2].

Um einen Teilnehmer in ein Netzwerk zu integrieren, muss man eine IP-Adresse und weitere Parameter wie Gateway und DNS-Server einstellen. Entweder macht man dies manuell oder per DHCP („Dynamic Host Configuration Protocol“). Verwendet man das DHCP-Protokoll, so kann der zuständige Router im lokalen Netzwerk das Elektor-Linux-Board automatisch konfigurieren (siehe **Bild 2**). Ist die IP-Adresse per DHCP zugewiesen worden, kann man zum Testen z.B. den Webserver unter www.elektor.de mit dem Befehl `ping` ansprechen (siehe **Bild 3**).

Ist kein DHCP-Server verfügbar, muss man die Parameter für die Netzwerkverbindung manuell eingeben.

IP-Adresse

Hierfür gibt man ein:

```
ifconfig eth0 <ipadresse>
```

Der Platzhalter `<ipadresse>` muss durch eine freie IP-Adresse im Netzwerk ersetzt werden. Um zu testen, ob die IP-Adresse erfolgreich übernommen wurde, kann man den Router mit dem Befehl `ping` ansprechen.

```
ping <routeradresse>
```

Der Platzhalter `<routeradresse>` muss entsprechend durch die IP-Adresse des Routers ersetzt werden.

```
root@gnublin:~# udhcpd
udhcpd (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.190...
Lease of 192.168.0.190 obtained, lease time 864000
Setting IP address 192.168.0.190 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.0.1
root@gnublin:~#
```

Bild 2.
DHCP-Konfiguration der IP-Adresse.

```
root@gnublin:~# ping www.elektor.de
PING www.elektor.de (94.236.12.177): 56 data bytes
64 bytes from 94.236.12.177: seq=0 ttl=108 time=89.798 ms
64 bytes from 94.236.12.177: seq=1 ttl=108 time=76.505 ms
```

Bild 3.
Ansprechen eines Servers mit dem `ping`-Befehl.

Adresse des Routers

Um Netzwerkpakete über den Router senden zu können, muss die Standard-Route für den Netzwerkverkehr eingerichtet werden. Dafür gibt es unter GNU/Linux einen Befehl auf der Kommandozeile:

```
route add default gw <routeradresse>
```

Der Platzhalter `<routeradresse>` muss wiederum durch die IP-Adresse des Routers ersetzt werden. Möchte man diesen Schritt einzeln überprüfen, so kann man einen beliebigen Computer im Internet mit dem Befehl `ping` ansprechen.

```
ping 94.236.12.177
```

Hier sprechen wir wieder den Elektor-Webserver an. Erhält man ein positives Ergebnis (keine verlorenen Pakete), kann man sich an den letzten Schritt machen.

Adresse des DNS-Servers

Um im Internet per Domain-Namen navigieren zu können, benötigt jeder Netzwerkteilnehmer einen DNS-Server. Der DNS-Server kann zu Domain-Namen wie z.B. „www.elektor.de“ die passende IP-Adresse ermitteln.

Um einen DNS-Server für das Elektor-Linux-Board einzustellen, editiert man die Datei „`/etc/resolv.conf`“ (**Bild 4**). Nach Eingabe von ...

```
nano /etc/resolv.conf
```

... öffnet sich der Editor. In der Datei sieht den Eintrag „`nameserver`“. Dort kann man seinen lokalen Router eingeben, der meist auch einen DNS-Dienst anbietet. Ist kein lokaler DNS-Server



Bild 4.
Konfiguration des DNS-Servers.



Bild 5.
Download der Datei „server.c“.

vorhanden, kann man einen aus dem Internet nutzen. Einfach zu merken und für Testzwecke ideal ist der DNS-Server von Google mit der IP-Adresse 8.8.8.8. Die Datei speichert man mit „Strg-o“. „Strg-x“ schließt den Editor wieder. Jetzt sollte jeder Leser erfolgreich beliebige Server im Internet per URL und dem ping-Befehl ansprechen können.

Eigenen Server-Dienst schreiben

Nachdem jetzt die Verbindung vom Board in das Internet steht, können wir uns die Beispieldatei für einen eigenen Serverdienst direkt aus dem Internet herunterladen.

Mit dem Programm „wget“, das wir bereits vom PC-Linux her kennen, lassen sich vom Board aus beliebige Dateien aus dem Netz ziehen. Der Download des Beispiel-Serverdienstes wird in **Bild 5** gezeigt. Der Serverdienst ist in der Programmiersprache C geschrieben. Für erste Tests verwenden wir den Compiler auf dem Board. Möchte man mit der Entwicklung selbstständig weitermachen, sollte man unbedingt mit der Toolchain auf dem Entwicklungs-PC arbeiten. In **Listing 1** sieht man den Quelltext der Anwendung. Im Wesentlichen wird ein einfacher Server gestartet, der auf Port 5000 lauscht. Meldet sich dort ein Teilnehmer, dann erhält er automatisch die aktuelle Uhrzeit auf dem System als Antwort. Später kann man hier einen Kommandointerpreter einbauen, der abhängig von dem vom Client versendeten Befehl verschiedene Antworten geben bzw. Aktionen durchführen kann.

Um den Server zu testen muss man die C-Datei übersetzen:

```
gcc server.c
```

Ist der Compiler fertig, kann man den Server starten mit:

```
./a.out
```

Jetzt blockiert die Anwendung die aktuelle Konsole und es scheint, als ob man nichts mehr machen kann (außer die Anwendung mit „Strg-c“ abzu-brechen). Doch das stimmt nicht. Man kann die Server-Anwendung mit „Strg-z“ schlafen legen, so dass man auf der Konsole wieder weitere Programme oder Befehle aufrufen kann.

Auf der Konsole erscheint dann:

```
[1]+  Stopped ./a.out
```

Gibt man jetzt...

```
fg
```

...ein (fg bedeutet „Foreground“), so ist die Konsole wieder blockiert. Man kann diese jedoch abermals mit „Strg-z“ freigeben. Danach kann man...

```
bg
```

...eingeben, der Server-Prozess läuft jetzt aktiv im Hintergrund weiter.

Wie schon erwähnt, schickt unser Test-Socketserver einem Client nach einer Anfrage die aktuelle Server-Uhrzeit. Testen kann man dies auf der Konsole mit dem Programm „telnet“, als Parameter werden die IP-Adresse des Servers und der Port angegeben:

```
root@gnublin:~/c# telnet 127.0.0.1 5000
```

```
Tue Sep 27 21:34:49 2011
```

```
Connection closed by foreign host
```

Möchte man den Server wieder in den Vordergrund holen (um ihn beispielsweise zu beenden), gibt man einfach wieder...

```
fg
```

...auf der Konsole ein.

Natürlich kann man nun auch von einer Weboberfläche aus eine Anfrage an den Server stellen. Und statt die Uhrzeit zurückzugeben, könnte der Server auch eine andere Anwendung steuern (die zum Beispiel eine LED an- und ausschaltet). Im letzten Teil wurde ja schon gezeigt, wie das geht [2].

Listing 1: Eine einfache Server-Anwendung.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

int main(int argc, char *argv[])
{
    int listenfd = 0, connfd = 0;
    struct sockaddr_in serv_addr;

    char sendBuff[1025];
    time_t ticks;

    listenfd = socket(AF_INET, SOCK_STREAM, 0);
    memset(&serv_addr, '0', sizeof(serv_addr));
    memset(sendBuff, '0', sizeof(sendBuff));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(5000);

    bind(listenfd, (struct sockaddr*)&serv_addr,
        sizeof(serv_addr));

    listen(listenfd, 10);

    while(1)
    {
        connfd = accept(listenfd, (struct sockaddr*)
            NULL, NULL);

        ticks = time(NULL);
        snprintf(sendBuff, sizeof(sendBuff),
            "%.24s\r\n", ctime(&ticks));
        write(connfd, sendBuff, strlen(sendBuff));

        close(connfd);
        sleep(1);
    }
}
```

Fernwartung per Netzwerk

Aktuell verbinden wir uns immer mit „picocom“ oder einem anderen Terminalprogramm zu der Konsole auf dem Board. Da wir jetzt aber über eine Netzwerkverbindung verfügen, können wir beliebig viele weitere Konsolen einfach über das Netzwerk öffnen. Dies hat vor allem während der Entwicklung von Anwendungen einen enormen Vorteil. In einer der Konsolen kann man beispielsweise den Editor mit dem Programm öffnen, während sich in einer weiteren Konsole das Programm immer wieder aufrufen lässt, um es im Live-Betrieb zu testen. Um eine Netzwerkverbindung zum Elektor-Linux-Board aufbauen zu können, verwenden wir SSH („Secure Shell“). Diese Konsole bietet einen automatisch verschlüsselten Kanal zum Board an. Dies hat den großen Vorteil, dass kein anderer Teilnehmer im Netzwerk mitlesen kann, was man zu dem Board sendet.

SSH von Linux aus

Unter Linux gehört „ssh“ zu den Standard-Tools. Auf der Konsole am Host-Rechner gibt man folgende Zeile ein:

```
ssh root@192.168.0.190
```

Die IP-Adresse muss hier durch diejenige ersetzt werden, die für das Board vergeben wurde. Wenn die Verbindung zum Board steht, wird man aufgefordert, das Wort „yes“ einzugeben

(Bild 6). Und schon sieht man die Standard-konsole, auf der man sich mittlerweile schon heimisch fühlen sollte.

SSH von Windows aus

Unter Windows kann man das Programm PuTTY

Listing 2: Start-Skript.

```
#!/bin/sh

if [ ! -d /var/log/lighttpd ]
then
    mkdir /var/log/lighttpd
    chown -R lighttpd:lighttpd /var/log/lighttpd
    chmod 777 /var/log/lighttpd/
    touch /var/log/lighttpd/error.log
fi

modprobe asix
udhcpc

echo 3 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio3/direction
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value

chown lighttpd:lighttpd /dev/lpc313x_adc
echo 1 > /dev/lpc313x_adc

chmod 666 /var/log/lighttpd/error.log
/etc/init.d/lighttpd start
```

Bild 6.
Zugang per SSH unter Linux.

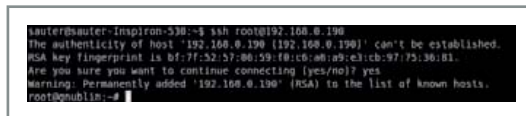
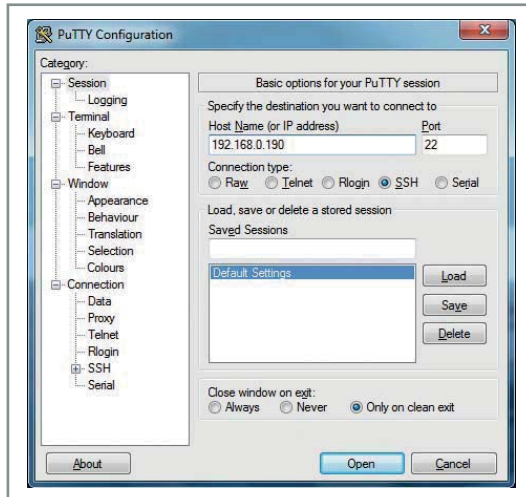


Bild 7.
Zugang per SSH unter Windows.



[3] verwenden. In **Bild 7** sieht man das Hauptfenster dieser Software; hier muss ebenfalls die IP-Adresse für das Board eingegeben werden. Nach einem Klick auf „Open“ wird man gefragt, ob der *Fingerprint* des Teilnehmers (ein Teil des SSH-Schlüssels zur Identitäts-Prüfung des Teilnehmers) korrekt ist. Hier kann man auf „Yes“ klicken. Bei dem Prompt „login as“ gibt man wie gewohnt „root“ ein. Nach einer Bestätigung mit der Enter-Taste ist man auf dem System angemeldet.

Skriptsprache Lua für einfache Steuerungen

Im Beispiel zuvor haben wir C als Programmiersprache verwendet. C ist sehr verbreitet, schnell und beliebt. Neben C können wir in einem

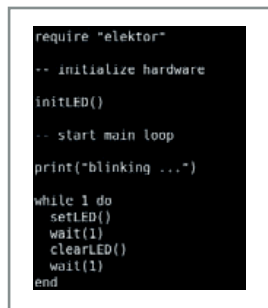
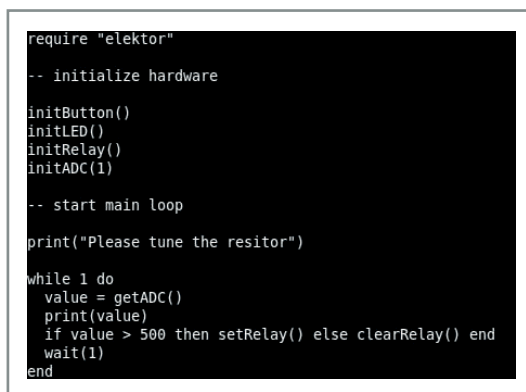


Bild 8.
Beispielprogramm „blink.lua“.

Bild 9.
Beispielprogramm „adc_relay.lua“.



Linux-System aber viele weitere Programmiersprachen nutzen. Jede Sprache hat ihre Vor- und Nachteile und eignet sich für verschiedene Anwendungszwecke.

Eine sehr flexible, schnelle und einfach zu verwendende Sprache ist Lua. Es handelt sich um eine interpretierte Sprache, die vor allem als Hilfssprache für Spiele bekannt geworden ist. Der Lua-Interpreter ist gerade mal 120 KB groß und bietet dennoch modernste Programmierkonstrukte an. Zugriff auf Dateien – und das können unter Linux ja auch Geräte oder Hardware-Funktionen sein – hat man mit den Befehlen `io.open`, `io.read`, `io.write` und `io.close`. Für das Elektor-Board hat der Autor eine Lua-Bibliothek geschrieben, um die wichtigsten Hardware-Funktionen (GPIO und A/D-Konverter) einfach nutzen zu können.

Möchte man schnell eine kleine einfache Steuerung realisieren, ist der Vorteil solch einer Interpreter-Sprache gegenüber C enorm. Denn auf dem Board kann man mit dem integrierten Editor (nano oder vi) direkt Programme schreiben und sofort ausführen lassen.

Über die neu eingerichtete Internetverbindung übertragen wir zuerst die Lua-Beispiele auf die SD-Karte:

```
mkdir lua
cd lua
wget http://www.gnublin.org/downloads/elektor.lua
wget http://www.gnublin.org/downloads/blink.lua
wget http://www.gnublin.org/downloads/button.lua
wget http://www.gnublin.org/downloads/adc_relay.lua
```

Das Lua-Listing eines einfachen Blinklichts sieht man in **Bild 8**. Zu Beginn muss die Elektor-Bibliothek geladen werden. Die LED wird mit einem Aufruf von `initLED()` initialisiert und anschließend in einer Endlosschleife ein- und ausgeschaltet. Auf der Konsole startet man das Programm so:

```
lua blink.lua
```

Ein einfaches Ansteuern des Relais mit dem IO15-Taster auf dem Board zeigt die Anwendung „button.lua“. Und in **Bild 9** ist der Code des Beispiels „adc_relay.lua“ zu sehen. In der Endlosschleife wird über den A/D-Wandler zuerst ein Spannungswert eingelesen. Liegt das Ergebnis über dem Grenzwert von 500, dann wird das Relais geschlossen, ansonsten wird es geöffnet. Zum Testen schließt man ein Poti an den Analog-Eingang an, so wie in der letzten Ausgabe

beschrieben [2].

Literatur für Lua findet man ausreichend im Internet [4] und es gibt auch gute Bücher [5].

Webserver und Anwendung automatisch starten

Es ist ein wenig umständlich, wenn man jedes Mal nach dem Hochfahren des Boards eine Server-Anwendung von Hand starten (und eventuell noch die Hardware konfigurieren) muss. Deshalb fassen wir die entsprechenden Befehle in einem Start-Skript zusammen. **Listing 2** zeigt ein solches Skript für unsere Webserver-Anwendung aus der letzten Ausgabe [2]. Die Datei „start.sh“ laden wir uns ebenfalls aus dem Internet herunter (siehe **Bild 10**).

Soll das Skript automatisch beim Start des Linux-Systems aufgerufen werden, kann man hierfür beispielsweise in die Skript-Datei „/etc/rcS.d/S55bootmisc.sh“ einen Eintrag aufnehmen (im Ordner „/etc/rcS.d“ stehen die Start-Skripts). Zuerst öffnen wir die Datei mit dem Editor:

```
nano /etc/rcS.d/S55bootmisc.sh
```

Am Ende der Datei findet man bereits die Einträge:

```
/bin/mkdir /var/run/sshd  
/usr/sbin/sshd
```

Nun ergänzt man dort:

```
/home/root/start.sh
```

Jetzt muss man die Datei wieder mit „Strg-o“ abspeichern und den Editor mit „Strg-x“ beenden. Damit die Datei „start.sh“ gefunden wird, muss diese im Verzeichnis „/root“ liegen. Ebenso muss noch das Executable-Flag „x“ für die Datei gesetzt werden:

```
chmod +x /home/root/start.sh
```

Ab jetzt sollte sich das Board automatisch nach dem Hochfahren mit dem Netzwerk verbinden, außerdem wird der Webserver gestartet.

Zugriff per Smartphone und Tablet

In der letzten Ausgabe haben wir gezeigt, wie man per PC-Browser auf eine Demo-Webanwendung zugreift. Natürlich geht das mit einem Smartphone oder Tablet genauso. Im Netzwerk, in dem das Elektor-Linux-Board hängt, muss man nur einen WLAN-Zugang für mobile Geräte eingerichtet haben. **Bild 11** zeigt die HTML-Oberfläche auf dem iPhone. Mit dem HTML-Button

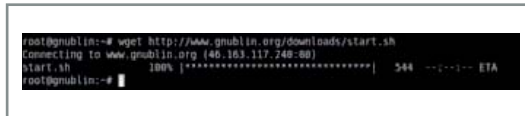


Bild 10.
Download des Start-Skripts
für den Webserver.



Bild 11.
Zugriff vom iPhone aus.

kann die LED auf dem Board ein- und ausgeschaltet werden.

An dieser Stelle noch ein Tipp für interessierte Leser: Im Community-Wiki [6] des GnuBlin-Projekts werden die verschiedensten Hard- und Softwarethemen behandelt. Unter anderem findet man dort Anleitungen, wie man über die I2C-, SPI- und UART-Schnittstelle Daten empfangen und senden (und in den User-Space übertragen) kann.

Für die nächste Folge könnten sich unsere Leser Themen wünschen (zum Redaktionsschluss dieses Artikels lief die Umfrage noch). Wir sind schon gespannt! Und schon jetzt können wir ankündigen, dass wir an einem Erweiterungsboard arbeiten, das weitere Schnittstellen bieten wird. Mehr darüber in der nächsten Ausgabe!

(120578)

Weblinks

- [1] sauter@embedded-projects.net
- [2] www.elektor.de/120182
- [3] www.putty.org
- [4] www.lua.org
- [5] www.lua.org/pil
- [6] <http://wiki.gnublin.org>

Zurück

zu den Wurzeln (10)

Hochfrequenz

Von
Burkhard Kainka
(D)

Mit der Hochfrequenz fing alles an, die Funktechnik war der Motor der Elektronik. Am Anfang standen die Elektronenröhre und der AM-Rundfunk, später kamen dann UKW und FM hinzu. Inzwischen arbeitet man mit Halbleitern, und die Frequenzen werden immer höher. Dennoch bleiben HF-Experimente auf Mittel- und Kurzwelle gerade für Einsteiger äußerst spannend.

Einmal selbst ein Radio bauen, das kann auch heute noch der Einstieg in die Elektronik sein. Aber ganz einfach sollte es sein, ohne Spezialbauteile und möglichst funktionssicher. Da bietet sich das Breitband-Audion nach **Bild 1** an.

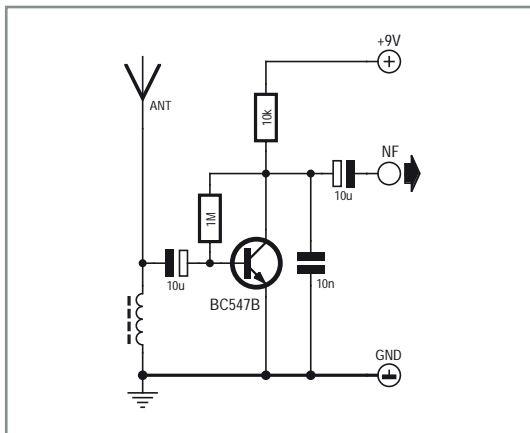


Bild 1.
Ein Breitband-Audion.

Als Audion bezeichnet man eine Schaltung, die HF-Signale gleichzeitig demoduliert und verstärkt. Dieser einfachste Empfänger bringt alles zu Gehör, was gerade in der Luft liegt, das kann ein naher Mittelwellensender sein, aber auch ein weit entfernter Kurzwellensender, und außerdem natürlich alle möglichen HF-Störungen aus dem näheren Umfeld. Voraussetzung ist nur eine genügend lange Drahtantenne. Etwa zehn Meter frei hängender Draht sind optimal, aber auch kürzere Antennen können schon zum Erfolg führen. Außerdem sollte es eine Erdverbindung geben. An den NF-Ausgang kann man einen empfindlichen Kopfhörer anschließen, noch besser aber einen Lautsprecherverstärker.

Das Breitband-Audion empfängt alles gleichzeitig. Normalerweise hört man mehrere Sender, wobei meist gerade einer besonders stark ist. Mit der Wahl der Spule hat man einen gewissen Einfluss auf den Empfangsbereich. Es kann auch eine Festinduktivität verwendet werden. Für Mittelwellenempfang sollte man etwa 300 μH nehmen, für die Kurzwelle ca. 10 μH oder auch nur 3 μH , wenn es um die höheren Bänder bei 15 bis 17 MHz geht. Eine selbst gewickelte Spule für Mittelwelle hat z.B. 100 Windungen auf einem Ferritstab; für Kurzwelle genügen etwa 10 bis 30 Windungen mit 10 mm Durchmesser ohne Ferritkern. Das Ergebnis hängt aber auch von der Antenne ab.

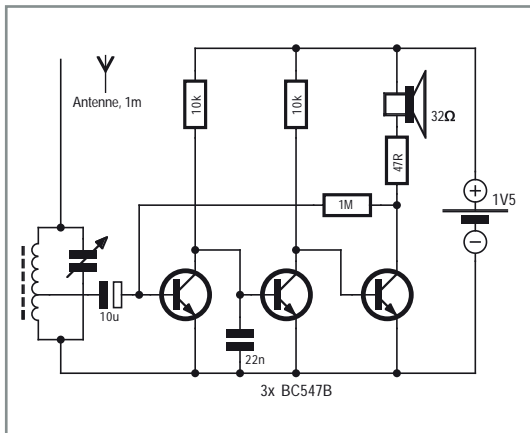


Bild 2.
Mehr Selektion und
Verstärkung für das Audion.

Versuchen Sie es mal mit ganz unterschiedlichen Spulen. Je nach Empfangsort, Tageszeit und verwendeter Spule sind ganz unterschiedliche Stati-

onen zu hören. So kann es z.B. passieren, dass Sie mit diesem einfachen Empfänger Radio China hören, denn besonders auf den höheren Kurzwellenbändern hat man eine sehr große Reichweite. Wenn man die Schaltung genauer ansieht, entdeckt man eigentlich nur einen Verstärker. Das Hochfrequenzsignal von der Antenne wird also verstärkt. Und wenn es ein wirklich perfekter Verstärker wäre, dürfte man absolut nichts hören, denn die verwendeten Frequenzen liegen weit oberhalb des hörbaren Bereichs. Dass es aber doch funktioniert, liegt an der „krummen“ Eingangskennlinie des Transistors. Die oberen Halbwellen des HF-Signals vergrößern den Kollektorstrom stärker als die unteren Halbwellen ihn verkleinern. Damit verändert sich der mittlere Strom in Abhängigkeit von der HF-Eingangsspannung. Da aber auf Seiten des Senders die HF-Amplitude im Takt der Audio-Signale moduliert wird, erscheint das NF-Signal am Ausgang.

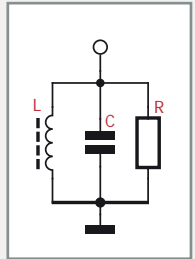
Wichtig sind auch die Kondensatoren in der Schaltung. Der relativ große Kondensator am Eingang schließt NF-Signale an der Basis über die Eingangsspule kurz. An der Basis liegen also nur HF-Spannungen. Im Kollektorkreis fließen verstärkte HF-Ströme und gleichzeitig demodulierte NF-Ströme. Ein 10-nF-Kondensator am Ausgang schließt aber die HF-Signale kurz. Übrig bleiben die hörbaren NF-Wechselspannungen. So bekommt man mit geringem Aufwand einen Empfänger mit recht guter Empfindlichkeit!

Bild 2 zeigt ein einfaches Radio mit dem gleichen Empfangsprinzip. Aber diesmal gibt es mehr Verstärkung und eine bessere Selektivität. Am Eingang liegt ein abstimmbarer Schwingkreis. Je nach den Spulendaten kann das Radio von Langwelle bis in den oberen Kurzwellenbereich reichen. Die Empfangsspule erhält eine Anzapfung bei etwa 10 % der Windungen. Damit erreicht man, dass der Schwingkreis nur wenig gedämpft wird und eine hohe Güte besitzt (siehe Infokasten Schwingkreise). So lassen sich einzelne Sender herausfiltern.

Mit insgesamt drei Transistoren und bei einer Versorgungsspannung von nur 1,5 V hat man damit bereits ein brauchbares Radio für den Kopfhörer-Empfang. Die Schaltung hat übrigens auch eine gewisse Ähnlichkeit mit dem HF-Schnüffler Tapir aus Elektor 7/8 2012 [1], der ebenfalls drei Transistoren hat, aber als breitbandiges Audion arbeitet.

Schwingkreise

Eine Spule L und ein Kondensator C bilden zusammen einen Schwingkreis. Elektrische Energie kann ähnlich wie die mechanische Energie in einem Pendel zwischen Spule und Kondensator hin und her schwingen, wobei eine definierte Resonanzfrequenz f auftritt. Der Schwingkreis führt eine freie Schwingung aus, nachdem er durch einen kurzen Stromstoß angeregt wurde. Der ideale Schwingkreis hat keine Verluste, eine einmal angeregte Schwingung würde ewig weiter schwingen. In der Praxis tritt jedoch durch Energieverluste am ohmschen Widerstand des Spulendrahtes, durch magnetische Verluste des Spulenkerns und durch elektromagnetische Abstrahlung eine Dämpfung der Schwingung auf. Man kann vereinfachend alle Verluste einem parallelen Verlustwiderstand R zuordnen.



Für die Resonanzfrequenz f gilt:

$$f = 1 / (2 \pi \sqrt{LC})$$

Für jeden Schwingkreis kann man einen Gütefaktor Q angeben. Q lässt sich ermitteln, wenn der parallele Dämpfungswiderstand R ins Verhältnis zum induktiven Widerstand $R_L = 2 \sqrt{f L}$ oder zum kapazitiven Widerstand $R_C = 1/(2 \sqrt{f C})$ bei der Resonanzfrequenz gesetzt wird.

$$Q = R/R_L \text{ bzw. } Q = R/R_C$$

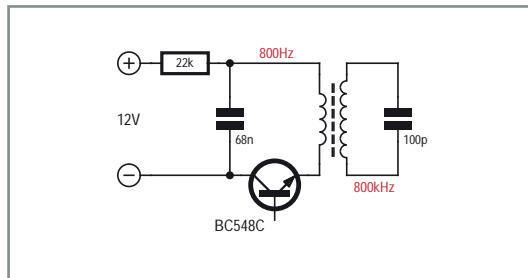
Erregt man einen Schwingkreis mit variabler Frequenz über eine Wechselstromquelle (mit hohem Innenwiderstand), dann ist die Schwingkreisspannung im Resonanzfall am höchsten. Je kleiner die Dämpfung der Schwingung durch Energieverluste jeglicher Art, bzw. je größer die Güte des Schwingkreises, desto höher steigt die Resonanzspannung. Zu beiden Seiten der Resonanzfrequenz lassen sich Punkte auf der Resonanzkurve bestimmen, bei denen die Spannung auf den Faktor $1/\sqrt{2} = 0,707 = -3 \text{ dB}$ abgefallen ist. Der Frequenzabstand dieser Punkte wird als die Bandbreite b des Schwingkreises bezeichnet. Zwischen der Resonanzfrequenz f, der Bandbreite b und der Güte Q des Kreises besteht der Zusammenhang $b = f/Q$.

Meist ist bei sorgfältigem Aufbau der Spule eine Güte im Bereich bis etwa $Q = 100$ erreichbar. Ein Schwingkreis wird jedoch auch durch die angeschlossene Schaltung oder durch eine Antenne bedämpft. Dieser Dämpfung wirkt man durch eine lose Kopplung des Schwingkreises über eine kleine Hilfswicklung, eine Spulenzapfung oder durch einen geeigneten Kondensator entgegen. Beim direkten Anschluss an einen Verstärker sollte dessen Innenwiderstand hoch sein, um die Dämpfung klein zu halten.

HF-Signale erzeugen

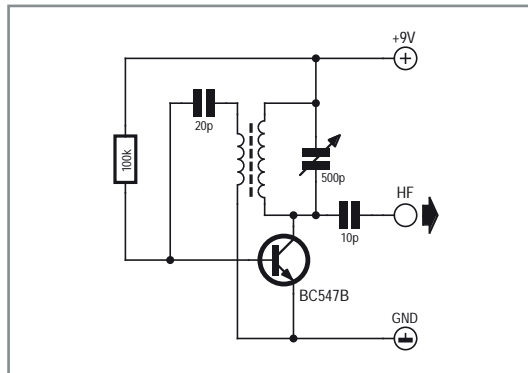
Die ersten funktionierenden Sender überhaupt waren Funkensender. Da reicht schon ein ganz normaler Lichtschalter, um das Prinzip zu verstehen. Man betätigt den Schalter, und aus dem

Bild 3.
Ein abgestimmter
Impulssender.



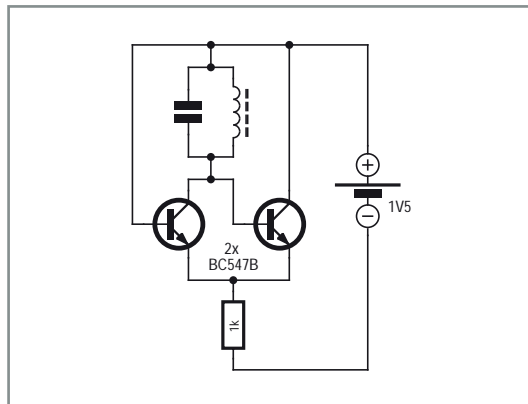
Radio hört man ein Knacken, jedenfalls auf Mittelwelle, wenn gerade kein starker AM-Sender zu hören ist. Ein Schaltsignal mit einer scharfen Flanke enthält grundsätzlich Signalanteile im Hochfrequenzbereich. Tatsächlich ist vor allem der Mittelwellenbereich vielerorts durch Schaltanteile und andere Geräte stark gestört. Das Problem ist also nicht, HF zu erzeugen, sondern eher umgekehrt, die Anstrahlung unerwünschter HF-Signale zu vermeiden.

Bild 4.
Ein Sinus-Oszillator mit
Transformatorkopplung.



Es muss ja nicht gleich ein ausgewachsener Schwarzsender sein, aber so ein ganz klein wenig HF-Power möchte man vielleicht auch mal selbst zur Verfügung haben, und wenn es nur zum Test der eigenen Empfänger dient. **Bild 3** zeigt einen ganz kleinen „Sender“, der nach dem Prinzip der alten Funkensender arbeitet. Die regelmäßigen „Funken“ entstehen hier mit einer Frequenz von ca. 800 Hz durch Kippschwingungen eines NPN-Transistors (siehe Folge 7, Elektor 9/2012 [2]). Jede dieser scharfen Schaltflanken regt den Schwingkreis zu Schwingungen an, die dann schnell wieder abklingen (gedämpfte Schwingungen). Die Sendefrequenz wird durch den Schwingkreis bestimmt. Verwendet man z.B. einen bewickelten Ferritstab aus einem alten Radio, liegt das Signal irgendwo im Mittelwellenbereich, z.B. bei 800 kHz. Wenn man ein Radio nahe an den Ferritstab hält, hört man ein Summen. Da fehlt eigentlich nur noch eine Morsetaste, um die Sache komplett zu machen. So ähnlich funktionierte das damals auch bei den Schiffs-Funkensendern, nur mit wesentlich mehr Leistung und auf viel tieferen Frequenzen.

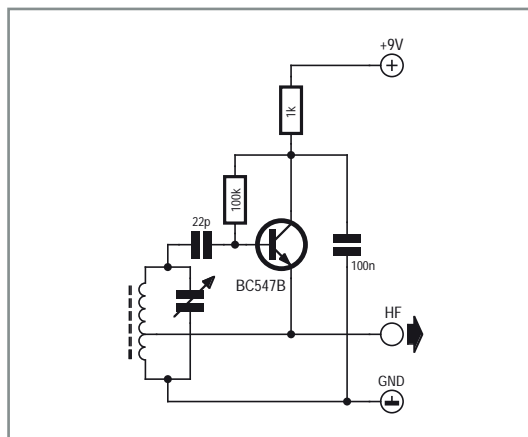
Bild 5.
Ein Oszillator für kleine
Betriebsspannungen.



LC-Oszillatoren

Während der Funkensender nur gedämpfte Schwingungen erzeugen konnte, kam der wahre Fortschritt erst mit der Elektronenröhre, die ungedämpfte Schwingungen generierte. Die letzte Folge des Kurses mit dem Thema Sinus-Oszillatoren [3] hat uns ja gezeigt, was man dazu braucht: Genügend Verstärkung und die passende Rückkopplung.

Bild 6.
Oszillator mit
Spulenzapfung.



Die Urform des HF-Oszillators ist der Meißner-Oszillator. Dabei sorgt eine Koppelwicklung für die richtige Phasenlage (**Bild 4**). So wurden auch die ersten Rundfunksender gebaut: Eine Röhre, Schwingkreis und Rückkopplung, fertig war der HF-Leistungsoszillator. Dazu wurde in die Antennenleitung ein Kohlemikrofon eingebaut, womit

auch die Aufgabe der Modulation gelöst war. Die Schaltung in Bild 4 zeigt einen Meißner-Oszillator mit einem Transistor. Es ist nicht schwierig, diese Schaltung zum Schwingen zu bewegen. Wenn es nicht klappt, ist eventuell die Phasenlage falsch, dann reicht es, die Koppelspule umzupolen.

Bei diesem einfachen Oszillator kann die Frequenz mit dem Drehkondensator eingestellt werden. Die Amplitude ist von vielen Dingen abhängig, z.B. von der Spulendämpfung, von der Rückkopplungswicklung und vom Koppelkondensator. Manchmal muss man etwas experimentieren, bis der Oszillator stabil arbeitet. Es gibt da nämlich noch ein besonderes Problem dieser Schaltung: Bei zu großer Amplitude können Pendelschwingungen entstehen. Die Basis-Emitter-Diode richtet die HF-Spannung gleich und lädt die Basis negativ auf, bis der Transistor sperrt und die Schwingungen abreißen lässt. Dann dauert es einige Mikrosekunden, bis der Basiswiderstand wieder für den richtigen Arbeitspunkt sorgt und das Spiel von vorn beginnt.

Diese Probleme kennt der Oszillator nach **Bild 5** nicht, denn hier wird die Amplitude automatisch stabilisiert. Außerdem wird nur eine einfache Spule ohne Koppelwicklung oder Anzapfung gebraucht, und nicht einmal eine besonders hohe Spulengüte ist erforderlich, weil die Verstärkung hoch ist. Die Schaltung ist also sehr bastelfreundlich und schwingt immer problemlos an. Die Rückkopplung wird hier durch zwei Transistoren erreicht. Und beide Transistoren arbeiten mit sehr kleiner Spannung (ca. 0,6 V). Die gesamte Schaltung arbeitet bereits ab einer Betriebsspannung von 1 V und kann mit sehr geringen Strömen betrieben werden. Das kann für einige Anwendungen Vorteile bringen.

Prinzipiell arbeitet diese Schaltung je nach Schwingkreis in einem großen Frequenzbereich von NF bis UKW, sodass sich mit geringem Aufwand z.B. kleine Test-Generatoren aufbauen lassen. Einen Nachteil gibt es aber auch: Die internen Kapazitäten der Transistoren wirken sich auf die Frequenz aus und sind außerdem noch von der Betriebsspannung abhängig. Die Schaltung ist daher zwar sehr einfach, dafür aber nicht gerade ein Wunder an Frequenzstabilität. Aber vielleicht fällt Ihnen ja dazu etwas ein, vielleicht eine drahtlose Batterieüberwachung?

AM-Generator mit Tiny13

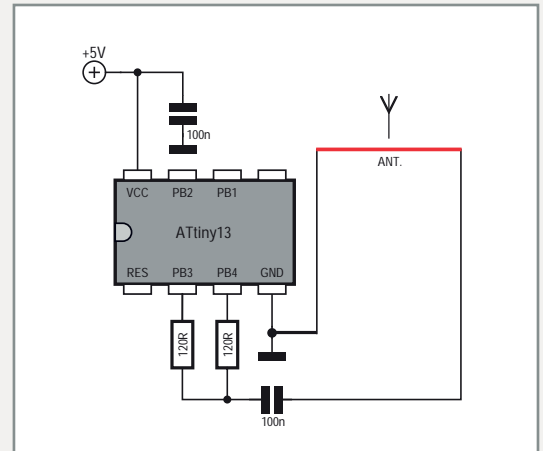
Wer Radioempfänger baut und entwickelt kann einen kleinen AM-Testgenerator gut gebrauchen. Bei geeigneter Signalform kommt man sogar ganz ohne eine Abstimmung der Frequenz aus. Das funktioniert, wenn der Generator auf einer ausreichend tiefen Frequenz arbeitet und genügend Oberschwingungen

erzeugt. Unser ATtiny13-AM-Generator erzeugt kurze Impulse mit einer Frequenz von 70 kHz. Es entstehen starke Oberwellen im gesamten Lang- und Mittelwellenbereich. Außerdem wird die Impulsfolge regelmäßig kurz unterbrochen. Damit kommt man zu einer Amplitudenmodulation mit einem 750-Hz-Ton. Ein Radio empfängt nun bei 70 kHz, 140 kHz, 210 kHz und so weiter einen summenden AM-Sender.

Als Antenne eignet sich eine Drahtschleife von z.B. 10 cm Durchmesser.

Sie erzeugt ein magnetisches Wechselfeld, das direkt an den Ferritstab des Empfängers koppelt. Der Generator eignet sich auch für vergleichende Messungen der Empfindlichkeit. Dazu testet man einfach, bis zu welcher Entfernung das Signal noch hörbar ist. Ein guter Empfänger empfängt noch in einem Meter Abstand ein klares Signal.

Den Quellcode kann man auf der Website zu diesem Artikel herunterladen [4].



```
'Attiny13 AM Generator
$regfile = "attiny13.dat"
$crystal = 1200000
$hstack = 8
$stack = 4
$framesize = 4
```

```
Config Portb = Output
Dim N As Byte
```

```
Do
  For N = 1 To 50 '70 kHz
    Portb = 255
    Portb = 0
  Next N
```

```
  For N = 1 To 50 'AM 750 Hz
    nop
    nop
  Next N
Loop
```

```
End
```

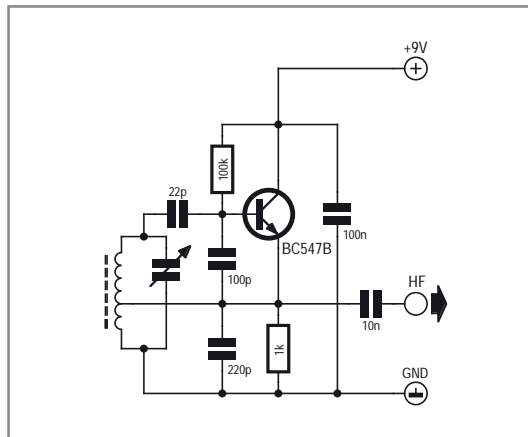


Bild 7.
Oszillator mit kapazitivem
Spannungsteiler.

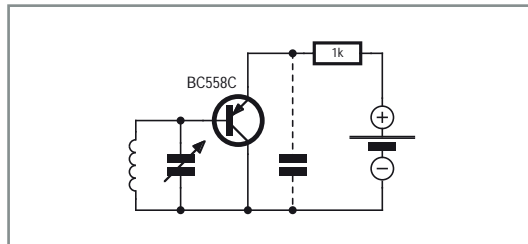


Bild 8.
DC-gekoppelter Oszillator in
Kollektorschaltung.

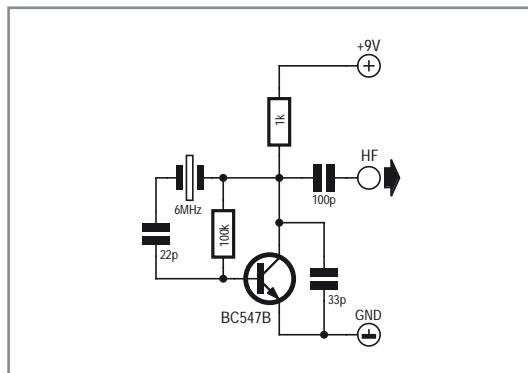


Bild 9.
Ein Quarz-Oszillator.

Die korrekte Phasenbeziehung in einem Oszillator kann man auch mit einer angezapften Spule erreichen (**Bild 6**). Oder man nimmt stattdessen einen kapazitiven Spannungsteiler (**Bild 7**). Verwendet man einen PNP-Transistor als Emitterfolger (Kollektorschaltung) mit direkter Gleichstromkopplung (**Bild 8**), dann kann die Schaltung noch einmal erheblich einfacher werden. Der kapazitive Spannungsteiler wie in Bild 7 kann bei ausreichend geringer Spulendämpfung sogar ganz entfallen, weil die inneren Transistorkapazitäten an seine Stelle treten. Außerdem kommt man wieder mit einer kleinen Betriebsspannung von z.B. 1,5 V aus. Allerdings führt die direkte Kopplung an den Schwingkreis auch hier wieder zu einer Beeinflus-

sung der Frequenz, weil die Sperrschichtkapazitäten sich mit der Betriebsspannung ändern. Dieser einfache Oszillator wirft auch ein Schlaglicht auf mögliche Probleme mit der Stabilität mancher Verstärkerstufen in Kollektorschaltung. Da wollte man vielleicht einfach nur einen Emitterfolger als Impedanzwandler bauen, aber versehentlich ist ein Oszillator daraus geworden. Ob es nur ein Verstärker wird oder eben ein Oszillator hängt z.B. vom Arbeitspunkt und von der Dämpfung des Schwingkreises ab.

Quarzoszillatoren

Wenn man eine richtig stabile Frequenz braucht, verwendet man am besten einen Quarzoszillator. Ein Quarz entspricht im Prinzip einem Schwingkreis extrem hoher Güte. Daher lassen sich Oszillatoren guter Stabilität mit geringem Aufwand ohne Induktivitäten realisieren. **Bild 9** zeigt eine typische Schaltung.

Wer gerne mit selbst gebauten Mittelwellenradios experimentiert, möchte vielleicht auch mal einen modulierten Oszillator für Testzwecke betreiben. Ein solcher Prüfsender wird nicht an eine Antenne gelegt und stört deshalb auch die Nachbarn nicht, denn die HF-Energie wird hier direkt von Spule zu Spule übertragen.

Aber stabil sollte er schon sein, sonst macht der Mittelwellen-Modulator keinen Spaß. Quarze für so kleine Frequenzen sind allerdings sehr teuer. Deshalb verwendet man hier gern Keramikresonatoren. Leicht erhältlich sind Frequenzen von 500 kHz oder 2 MHz, beide gerade außerhalb des Mittelwellenbereichs. Es gibt aber auch andere Frequenzen, die besser passen. Ein Keramikresonator von 976 kHz wurde z.B. in einer alten Fernseh-Fernbedienung entdeckt. **Bild 10** zeigt einen kompletten AM-Prüfoszillator. Mit dem Trimmer ist eine Feinjustierung möglich. Falls im Hintergrund noch eine (wahrscheinlich schwache) Station zu hören ist, wird einfach auf Schwebungsnul abgestimmt, z.B. auf 981 kHz. Die kleine Sendeferritsspule koppelt direkt auf den Ferritstab im Radio. Man könnte die Schaltung allerdings auch ohne große Änderungen mit einem Quarz im Kurzwellenbereich betreiben, z.B. auf der für experimentelle Anwendungen legalen Frequenz 13,56 MHz.

Die eigentliche Modulatorstufe ist als Emitterfolger ausgelegt und moduliert die Betriebsspan-

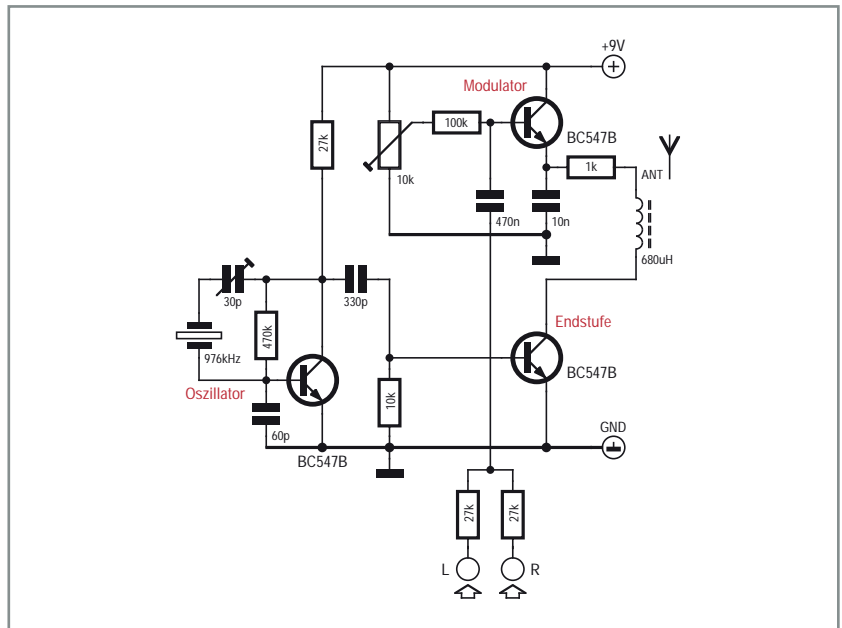
nung des Endverstärkers. Da man auf Mittelwelle und Kurzwelle ja noch mit Mono arbeitet, werden beide Eingangskanäle eines Stereo-Signals zusammengefasst. Mit dem Poti wird auf geringste Verzerrungen und besten Klang justiert. Die HF-Verstärkerstufe wurde bewusst bescheiden dimensioniert, denn hier kommt es nicht auf hohe Leistung an.

Kurzwellen-Audion mit Rückkopplung

Der dreistufige Kurzwellen-Empfänger nach **Bild 11** besitzt einen zusätzlichen Einstellregler für die Rückkopplung. Im Prinzip ist die erste Stufe ein Oszillator wie in Bild 8. Man kann aber den Arbeitspunkt beliebig einstellen und damit die Verstärkung reduzieren. Der Trick ist nun, dass man die Verstärkung so einstellt, dass alle Verluste des Schwingkreises gerade ausgeglichen werden, also eben noch keine Eigenschwingungen einsetzen. An diesem Punkt hat der Empfänger die größte Verstärkung und die beste Trennschärfe. Die PNP-Oszillatorstufe in Kollektorschaltung ist zugleich auch die Audionstufe und demoduliert das HF-Signal. Zwei nachfolgende NF-Stufen sorgen für eine ausreichende Leistung, sodass sogar ein kleiner Lautsprecher eingesetzt werden kann.

Ob das Audion bis zum Einsatz der Schwingungen eingestellt werden kann, hängt auch von der Dämpfung des Eingangskreises durch die angeschlossene Antenne ab. Daher gibt es zwei unterschiedliche Antennenanschlüsse. Beim Anschluss über einen kleinen Koppelkondensator ist die Kopplung lose und die Dämpfung gering. Der direkte Anschluss eignet sich dagegen auch für sehr kurze Antennen (eine längere Antenne strahlt nämlich zugleich auch HF-Energie ab und dämpft auf diese Weise den Schwingkreis). Bei geschickter Einstellung kann man mit einem Audion extreme Empfindlichkeiten erreichen. In der Anfangszeit der Funktechnik gehörten solche Audio-Empfänger zur Standard-Ausrüstung. Sogar mit schwachen Sendern wurden bereits Entfernungen von einigen tausend Kilometern überbrückt. Die Faszination dieser Schaltung kann man heute noch erleben, denn mit geringstem Aufwand lassen sich hervorragende Empfänger bauen.

Damit sind wir am Ende unserer Serie angelangt. Wir wollen aber auch in den nächsten Heften in lockerer Folge zu den Wurzeln der Elektronik



zurückkehren. Dann sollen zum Beispiel Operationsverstärker und wichtige Digital-ICs auf dem Stundenplan stehen. Bleiben Sie uns treu!

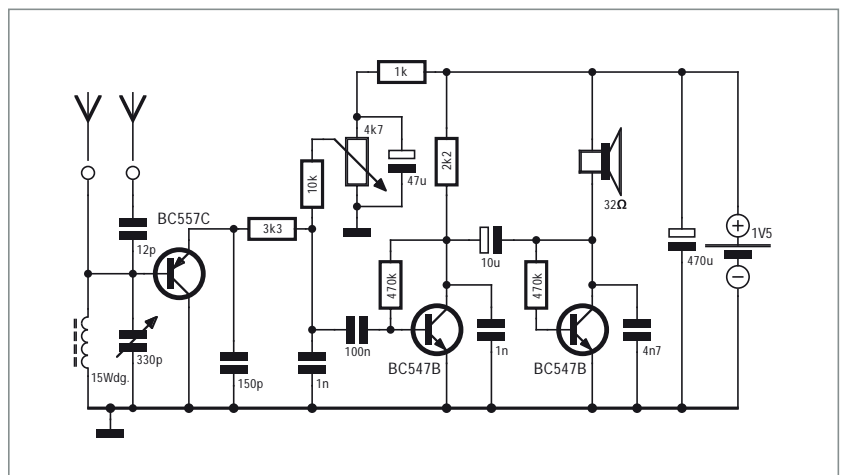
Bild 10.
Der Mittelwellenmodulator.

120010

Weblinks

- [1] www.elektor.de/120354
- [2] www.elektor.de/120007
- [3] www.elektor.de/120009
- [4] www.elektor.de/120010

Bild 11.
Ein Rückkopplungs-Audion.





Entwicklung
industrietauglicher
Software und Hardware
sowie Elektronik
**03303/212166 oder
www.jasys.de**



Gravuren / Schilder /
Frontblenden
www.webgrav.de

LOETRONIC

Embedded MP3 Module
www.loetronic.com

Alles Spule!

Wir **liefern und fertigen**:
Drähte, HF-Litzen, Ferrit- und
Eisenpulverkerne, Spulenkörper,
Isoliermaterial, Klebebänder,
Tränklacke, Übertrager, RFID-Spulen,
Sensor- und Aktorspulen, Prototypen,
Kleinserien, Serien, Ersatzteile
und vieles mehr.

MM Menting
Mikroelektronik
Spezial für Elektronik
www.spulen.com

Auch Ihr Unternehmen ist
eine Anzeige wert!

Bausätze zu ELEKTOR 1986 bis heute!

Teilesätze, Platinen, programmierte
Kontrollen sowie Cds zu fast allen
Elektor-Projekten vom Spezialist.
Alle Elektor-Artikel zum Verlagspreis.

Ihr zuverlässiger Partner für aktive
und passive elektronische Bauteile
und Komponenten:



**Geist Electronic-
Versand GmbH**
Tel.: 07720/36673
Fax: 07720/36905

Mail: info@geist-electronic.de
Shop: **www.geist-electronic.de**

**580,- €
SPAREN!**

HAMEG
Instruments
A Rohde & Schwarz Company



hameg.com/xmas2012

USB-ISOLATOR



Galvanische
Trennung
Störfreie
Messungen
Sicherheit
und Schutz

CESYS **www.cesys.com**

Elektronik · Messtechnik · FPGA

W2micro **www.w2micro.de**

- diverse SMD-Adapter
- RS232-TTL-Pegelwandler
- Sonderelektronik (z.B. Audioverteiler)
- Leiterplattenbestückung und Kabelkonfektionierung ab 1 Stück

Günstige Platinen
clever bestellen!



www.pcb-joker.com

DEIWO®

Dienstleistungen für die
Elektronik-Industrie

- Bestückung SMD und THT
- Verdrahtung
- Kabelkonfektionierung
- Baugruppenreinigung
- Verguss, Blech Biegeteile
- mechanische Baugruppen
- CNC Gehäuse-Bearbeitung
- Montage und Endprüfung

www.deiwo.de

Fertigung in Tschechien, Deutsche Leitung,
ISO 9001 zertifiziert, FAIRE PREISE!

Ansprechpartner: Herr Jorntitz
Telefon: 00420 475 205 071, Telefax: 00420 475 226 141
eMail: info@deiwo.cz | Anschrift:
Krásný Les 341, 40337 Petrovice, Tschechien

www.anttronic.de

ab 1 Stck. **ANTTRONIC**
Leiterplatten zu TOP-Preisen!!

Kleinrechner mit FPGA

www.bomerenzprojekt.de

Leiterplatten für
Pragmatiker!



www.pcb-joker.com

Schaeffer
AG

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: 34,93 €
zzgl. USt./Versand

Schaeffer AG
Nahmitzer Damm 32
D-12277 Berlin
Tel +49 (0)30 8 05 86 95-0
Fax +49 (0)30 8 05 86 95-33
Web info@schaeffer-ag.de
www.schaeffer-ag.de

Elektor im Netz



www.elektor.de

Das Internet der Dinge

Von Tessel Renzenbrink
(Elektor TTF-Redakteurin)

Das IoT (Internet of Things) ist die Verknüpfung von physischer und digitaler Welt. Gewöhnliche mit einem Controller bestückte Alltagsgegenstände verfügen über eine IPv6-Adresse und sind so im Internet präsent. Man kann Informationen über die Dinge des täglichen Lebens per Internet abrufen und einige davon sogar fernsteuern.



Bei vielen Aspekten des IoT sind Menschen nicht mehr beteiligt. Beim IoT als Netzwerk von Maschinen kommunizieren Sensoren und Aktuatoren miteinander und nehmen so Einfluss auf ihre Umgebung. Im September fand der Workshop „ROBIOTS: Robots Meet Internet of Things“ als Teil der zweitägigen Veranstaltung „PICNIC“ statt, dem Festival für Medien und Innovation in Amsterdam. Ich hatte die Gelegenheit, mit Experten über die Zukunft des IoT, die technischen Herausforderungen und auch eine mögliche Bedrohung zu diskutieren. Teilnehmer waren Florian Michahelles, Heico Sandee und Stefan Gessler.

Die IoT-Experten

Dr. Florian Michahelles ist der stellvertretende Direktor der Auto-ID Labs in Zürich/St. Gallen (CH). Dabei handelt es sich um ein globales Netzwerk von sieben wissenschaftlichen Forschungslabors im Bereich vernetzte RFIDs.

Dr. Heico Sandee ist der Programm-Manager der Abteilung Mechanik der Technischen Universität Eindhoven (NL). Er ist Spezialist für Automatisierung und Mitglied im RoboEarth-Projekt, einer multidisziplinären Kooperation von Forschung und Industrie zum Thema Roboter, die eine Art WWW für Roboter etablieren will.

Stefan Gessler ist Projektleiter in der „Software and Service Research Division“ der NEC-Labs in Europa. Er ist Mitglied beim Florence-Projekt, eine Initiative zur Interaktivität von preiswerten

Robotern mit „Smart Home“-Technik zur Unterstützung älterer Menschen im Alltag. Außerdem beteiligt er sich im IoT-Forum.

Was taugt IoT?

Tessel: Warum braucht die Welt das IoT?

Florian: Eine Antwort wäre: Es passiert einfach. Es ist einfach die Weiter-Entwicklung der vorhandenen Technik.

Heico: Es geht um Steuerungsmöglichkeiten. Ich bin es gewohnt, jederzeit mit jedem kommunizieren zu können – ich finde es dumm, dass ich das nicht mit den Dingen des Alltags tun kann. Ich möchte mit meinem Smartphone nachschauen können, ob ich die Haustüre abgeschlossen habe und dies aus der Ferne nachholen, falls es vergessen wurde.

Stefan: Beim IoT geht es um Kooperation und Kommunikation. Letzteres ist ja die Vorbedingung für Ersteres. Man kann sich tausende Szenarien vorstellen, wo es von Vorteil ist, wenn Dinge zusammenarbeiten. Sollte ein Sensor feststellen, dass der Boden im Badezimmer nass wird, dann könnte dieser dem Wasserhahn signalisieren, dass er sich schließen soll. Oder wenn ein Auto eine gefährliche Situation vor sich entdeckt, könnte es den nachfolgenden Autos signalisieren, dass sie besser langsamer fahren.

Heico: Und es eröffnet völlig neue Möglichkeiten. Im RoboEarth-Projekt arbeiten wir an einem System, das Informationen zwischen Objekten, Robotern und Menschen austauscht. Wenn man

www.techthefuture.com

Florian Michahelles.



Heico Sandee.



Stefan Gessler.



z.B. die Kaffeemaschine mit dem Wecker koppelt, dann wird die Maschine gestartet, sobald das Wecksignal kommt. Das ist mit einer „smarten“ Kaffeemaschine schon heute möglich. Doch wenn der Maschine der Kaffee ausgeht? Das ist der Punkt, wo ein Roboter ins Spiel kommt. Er kann sich nämlich bewegen und den Behälter für Kaffeebohnen wieder auffüllen. Roboter sind die Arme und Beine des Internets. Und wenn Roboter Teil eines Netzwerks sind, dann können sie voneinander lernen. RoboEarth arbeitet an einer cloud-basierten Datenbank, auf die von unterschiedlichen Robotern zugegriffen werden kann. Wenn ein Roboter ein Problem gelöst hat, dann steckt er diese Information in die Cloud, sodass der nächste mit diesem Problem befasste Roboter die Instruktionen lesen kann. Auf diese Weise muss man bei der Roboterentwicklung nicht immer wieder die gleichen Probleme lösen. Und wenn man den Robotern Lernfähigkeiten mitgibt, muss man sie auch nicht mit sehr vielen Sensoren ausstatten.

Herausforderungen

Tessel: Was sind die technischen Herausforderungen bei der Realisierung des IoT?

Stefan: Am schwierigsten ist sicher der Entwurf von Informationsmodellen, mit denen die Geräte untereinander kommunizieren. Im Moment dreht sich viel um den so genannten Silo-Ansatz. Man hat Anwendungsfelder und Geräte, die für eine bestimmte Aufgabe entwickelt wurden. Das ist dann ein geschlossenes System mit Sensoren, Aktoren und Software. Bei einer zweiten Anwendung muss man alles noch einmal kaufen, aber diesmal arbeiten die Sensoren und Aktoren mit einer anderen Software und anderen Protokollen. Der nächste Schritt wäre also die Wiederverwendung und Verteilung der gewonnenen Daten an andere Geräte. Dazu braucht es aber ein interoperables Interface. Genau daran arbeiten wir zurzeit.

Florian: Bei den Auto-ID-Labs fokussieren wir die Produktseite: Wie macht man den Güterstrom flüssiger und die

Logistik-Kette effizienter? Die Idee ist, einen Barcode oder ein RFID mit Daten zu verknüpfen, die das Produkt beschreiben. Der 40 Jahre alte Barcode ist nun plötzlich auch für Konsumenten mit Smartphones interessant. Doch der Konsument interessiert sich für andere Daten. Größe und Gewicht des Produkt interessiert ihn weniger als die Inhalte oder die Anwendungen. Als nächstes müssen wir eine Semantik aus den Daten gewinnen, damit diese von Robotern interpretiert werden können.

Doch Standardisierung ist weniger eine technische denn eine organisatorische Herausforderung, da nicht eine Firma die komplette Infrastruktur stellt, sondern viele verschiedene Akteure beteiligt sind. Wie bringt man diese Teilnehmer zusammen? Hat z.B. eine Firma Daten generiert, was ist dann der Anreiz für sie, diese zu teilen? Die entscheidende Frage wäre also die nach dem Ökosystem. Wie muss dieses aussehen, damit unterschiedliche Firmen Daten weitergeben, weil sie davon gegenseitig profitieren?

Heico: Ich stimme zu, dass Interoperabilität die größte Herausforderung ist, doch ist das wirklich so ein großes Problem? Ist es nicht eher ein Problem, sich erst einmal auf mögliche Anwendungen zu einigen?

Stefan: Ja, doch dabei müssen alle potentiellen Anwendungsfälle berücksichtigt werden.

Florian: Genau das ist der Punkt. Es ist technisch sehr schwierig, Lösungen für alle denkbaren Anwendungsfälle zu entwickeln. Es ist einfacher, sich z.B. auf fünf Szenarien zu einigen und dann die Standardisierung anzugehen.

Stefan: Das IoT entwickelt sich bottom-up, also von unten nach oben. Wir können nicht einfach ein IoT-System installieren und es dann anschalten, wie einst beim Fernsehen.

Internet-Infrastruktur

Tessel: Das Internet ist das Rückgrat des IoT. Reicht der gegenwärtige Ausbau des Internets dafür aus?

Stefan: Wir brauchen immer mehr Bandbreite. Zwar funktioniert das Internet bislang ganz gut, doch sollte man auf große Datenmengen vorbereitet sein. Das Datenaufkommen wächst exponentiell. Wie einer der Workshop-Sprecher sagte: Seit Beginn der Zeitrechnung fielen etwa fünf Billionen Gigabyte an. Diese Datenmenge wurde 2011 schon in zwei Tagen erreicht. Und 2013 wird das in etwa 10 Minuten der Fall sein.

Außerdem ist die Kapazität der Funkfrequenzen

begrenzt. Man kann nicht einfach eine unbegrenzte Zahl von Geräten an ein Funknetzwerk anschließen.

Florian: Man muss also den Mangel verwalten. Wenn das Internet heute klemmt, dann muss man eben etwas länger auf einen Download warten. Der meiste Datenverkehr ist ja nicht dringend. Beim IoT ändert sich das. Es könnten medizinische Anwendungen betroffen sein, die permanenten Zugang benötigen. Bei der Ressourcenverteilung muss also medizinischen Applikationen Priorität eingeräumt werden. Meines Erachtens werden wir immer an der Grenze der Bandbreite arbeiten, aber das ist keine Bremse für das IoT.

Stefan: Im Gegenteil: Das IoT könnte zu einer effizienteren Verteilung der Datenströme führen. Es geht ja darum, Entscheidungen früh zu treffen. Daten, die nur ein lokales Netzwerk betreffen, müssen ja nicht zu einem entfernten Rechenzentrum geschickt werden. Momentan wird zwischen „lokal“ und „in der Cloud“ unterschieden. Zukünftig wird es unterschiedliche Layer geben, also unterschiedliche Cloud-Ebenen. Daten werden hierarchisch organisiert sein: entweder nur lokal, regional oder gar global verfügbar.

Die dunkle Seite

Tessel: Was sind die möglichen Gefahren dieser Technologie?

Heico: Um die Sicherheit müssen wir uns schon Sorgen machen. Wir können bereits jetzt sehen, welche Schäden bösartige Hacker im Internet anrichten. Wenn alle möglichen Dinge an das Internet angeschlossen sind, wird das Ausmaß der Gefahr deutlich größer sein.

Stefan: Bei jeder Neuentwicklung bestehen negative Verwendungsmöglichkeiten! Ein anderer Aspekt ist die Beeinflussung unseres Lebens durch neue Technologien. Viele sehen das als Bedrohung, weshalb wir uns zukünftig auch um die Akzeptanz des IoT kümmern müssen. Sicherheit und Datenschutz sind da sicherlich zentral.

Florian: Bedenklich könnte auch eine mangelnde Transparenz sein. Menschen sind es gewohnt zu verstehen, warum etwas passiert. Mit der neuen digitalen Komponente unserer Welt wird man mit unseren Sinnen oft nicht mehr die Ursache von Wirkungen wahrnehmen können. Deshalb ist es erforderlich, dass man immer auch ein mentales Modell des Geschehens entwirft. Wenn ich nicht begreifen kann, was passiert, entsteht Angst. Es könnte bedeuten, dass ich von jemand anderem gesteuert werde. Das ist ein übles Szenario. Doch in meinen Augen wäre es das Schlimmste, wenn das IoT gar nicht erst Realität wird.

(120632)

Weblinks

- [1] www.autoidlabs.org
- [2] www.roboearth.org
- [3] <http://iot-forum.eu>

GROßE SPEICHERTIEFE MIXED-SIGNAL OSZILLOSKOPE NEU von Pico Technology



PORTABILITÄT & HÖCHSTLEISTUNG

pico[®]
Technology

PicoScope	3204 MSO	3205 MSO	3206 MSO
Kanäle	2 analog 16 digital		
Bandbreite	60 MHz	100 MHz	200 MHz
Speichertiefe	8 MS	32 MS	128 MS
Auflösung (erweitert)	8 bits (12 bit)		
Signalgenerator	Funktionsgenerator + AWG		
Preis	€785	€1028	€1270

ALLE MODELLE MIT OPTIMIERTEM DIGITALEM TRIGGER, SERIELLEM DECODER (I2C, SPI, RS232, CAN, LIN UND FLEXRAY), MASKENBEGRENZTE TESTS, SEGMENTIERTER SPEICHER, DIGITALE FILTER, KONSTELOSE SOFTWARE-UPDATES UND FÜNF JAHRE GARANTIE

www.USBmso.com/PS178

Hexadoku Sudoku für Elektroniker

Vielleicht haben Sie nach einem richtig anstrengenden Weihnachtseinkauf einmal Lust auf etwas Ruhe? Bei unserem monatlichen Hexadezimal-Rätsel finden Sie sicher Entspannung! Wer uns die richtigen Zahlen in den grauen Kästchen zuschickt, hat wie immer die Chance, einen von vier schönen Gutscheinen zu 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 **ELEKTOR-Gutschein im Wert von 100 €**
und drei **ELEKTOR-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. Dezember 2012!

Die Gewinner des Hexadokus aus dem Oktoberheft stehen fest!

Die richtige Lösung ist: **75E2B**.

Der Elektor-Gutschein über 100 € geht an: Steve Hasko aus Brampton (UK).

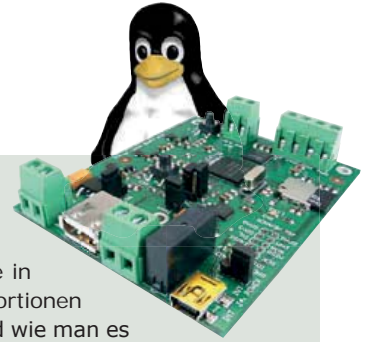
Einen Elektor-Gutschein über je 50 € haben gewonnen: Michael Franke, Jean-Paul Lagaisse und Franz Maurer.

Herzlichen Glückwunsch!

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

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



Embedded Linux in Theorie und Praxis (3-tägiges Seminar)

Sie haben schon mal, so eher schlecht als recht, mit Embedded Linux herumgespielt und waren ganz einfach überwältigt? Sie haben bis jetzt den Schritt noch nicht gewagt und wollen sich nur mal umschaun, was man damit eigentlich so machen kann? Dann sind Sie hier genau richtig! Was Sie in Eigenregie so ca. ein Jahr beanspruchen würde, bekommen Sie in wenigen, leicht verdaulichen Portionen serviert. In drei Tagen bekommen Sie einen groben Überblick, wie Embedded Linux aufgebaut ist und wie man es auf einer Embedded-Hardware-Plattform effektiv einsetzt.

Referent: R. Berger – Teilnahmegebühr: 1.898,00 € (inkl. MwSt.)

PIC-Mikrocontroller-Programmierung in C (3-tägiges Seminar)

In diesem 3-tägigen Kurs werden Sie die Programmierung von „Eingebetteten Systemen“ in der Programmiersprache C kennenlernen. Dazu wird auf die PIC-Mikrocontroller-Familie der Firma Microchip anhand des C18-/C30-Compilers eingegangen. Das Hauptaugenmerk liegt auf der C-Sprache, um sich besser auf die verschiedenen Elemente von C selbst zu konzentrieren. Die Darstellung wird von einer Reihe praktischer Übungen, die Sie selbst an Ihrem Notebook durchführen werden, begleitet. Dabei werden Übungen innerhalb des MPLAB-Simulators ausgeführt und ermöglichen Ihnen, die Vertiefung des Stoffes. Ferner ist es möglich, das Erlernete mit jedem ANSI C-Compiler anzuwenden. Abgeschlossen werden die Betrachtungen mit praktischen Übungen, die Sie selber mit einem PICKit 3 Debug Express und zugehöriger Platine durchführen werden.

Das Hardware-Tool im Wert von ca. 69,00 € nehmen Sie im Anschluss mit nach Hause.

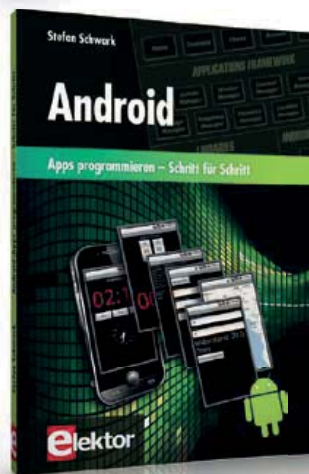
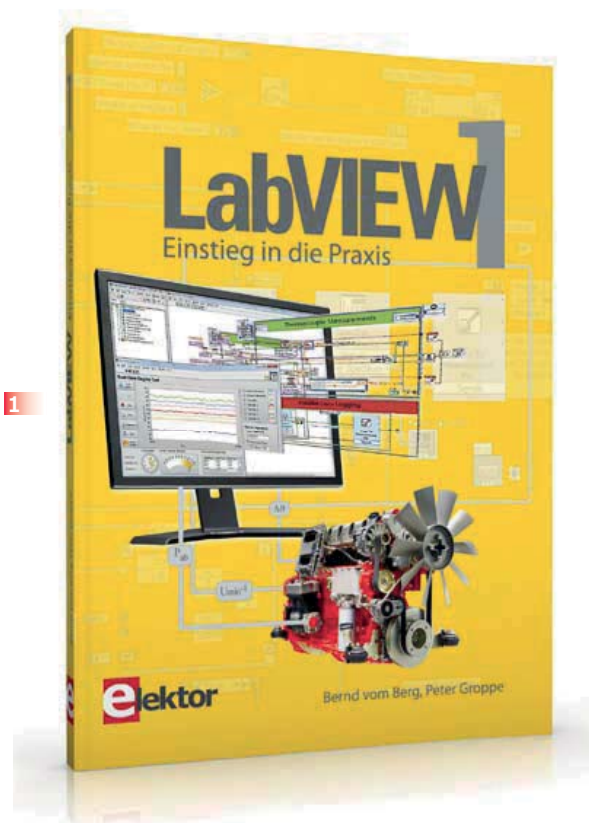
Referent: Prof. Dr.-Ing. Francesco P. Volpe – Teilnahmegebühr: 1.299,00 € (inkl. MwSt.)

Messen, Steuern, Regeln mit VB.NET' (2-tägiges Seminar)

In diesem 2-tägigen Seminar lernen Sie ganz konkret und praktisch, wie man mit einem Standard-PC und einer über die USB-Schnittstelle angeschlossenen Hardware verschiedenste Mess-, Steuer- und Regelaufgaben erledigt. Dabei wird im Seminar die Programmiersprache Visual Basic verwendet. Diese weit verbreitete Programmiersprache bietet die Vorteile der schnellen Erlernbarkeit und der leichten Lesbarkeit des erstellten Codes.

Referent: Dipl.-Ing. Franz Peter Zantis – Teilnahmegebühr: 999,00 € (inkl. MwSt.)





Einstieg in die Praxis

1 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

Bestückte und getestete Platine inkl. Gehäuse

2 USB-Isolator

Masseschleifen zwischen PCs und Geräten, die über USB miteinander kommunizieren, sind oft Ursache hoher Störpegel auf den Signalen. Der USB-Isolator

ist der Retter in der Not, wenn die Systeme in störintensiver Umgebung arbeiten müssen. Er trennt die Systeme galvanisch voneinander und schlägt gleichzeitig eine Brücke, sowohl für die Daten- als auch die Betriebsspannungsleitungen.

Art.-Nr. 120291-91 • € 69,95 • CHF 86,80

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

Band 1: 35 Einsteiger-Projekte in C

5 ARM-Mikrocontroller

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,

Die ganze Welt der Elektronik in einem Shop!



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

Von 0 und 1 zum FPGA **Digitale Logik** **6 selbst entwickeln**

Dieses Buch nimmt Sie mit auf eine Entdeckungsreise in die Welt der digitalen Elektronik. Nach dem Aufbau einer soliden Wissensbasis hierüber verlagert sich der Schwerpunkt zur programmierbaren Logik. Wie lassen sich mit vorhandenen Bausteinen logische Systeme aufbauen und wie koppelt man sie sicher und störungsfrei an die analoge Außenwelt? Das sind Fragen, die das Buch beantwortet.

603 Seiten (kart.) • ISBN 978-3-89576-254-3
€ 49,00 • CHF 60,80

Bausatz mit allen Bauteilen & Platine(n) **TAPIR –** **7 E-Smog-Detektor**

Der TAPIR (Totally Archaic but Practical Interceptor of Radiation) spürt "strahlendes Missverhalten" elektronischer Geräte in Ihrer Umgebung auf. Trotz der einfachen Schaltung handelt es sich hierbei um einen ultrasensitiven E-

Smog-Detektor, der jede Quelle eines elektrischen oder – mit einer entsprechenden Antenne versehen – magnetischen Feldes aufspürt und dies akustisch signalisiert.
Art.-Nr. 120354-71 • € 14,95 • CHF 18,60

Theorie und Praxis mit WinFACT und Multisim **8 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

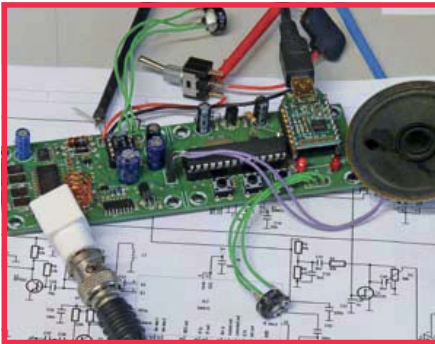
Weitere Informationen zu unseren Produkten sowie das gesamte Verlagssortiment finden Sie auf der Elektor-Website:

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

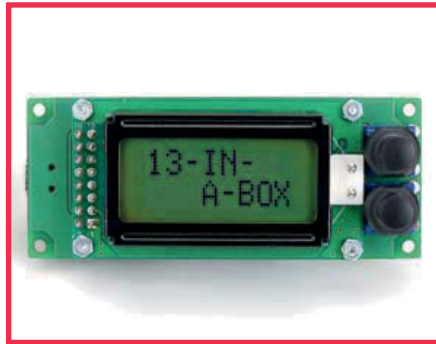
e Nächsten Monat in Elektor

Die nächste Ausgabe ist extra-stark! Mit vielen Projekten, Ideen und Tipps!



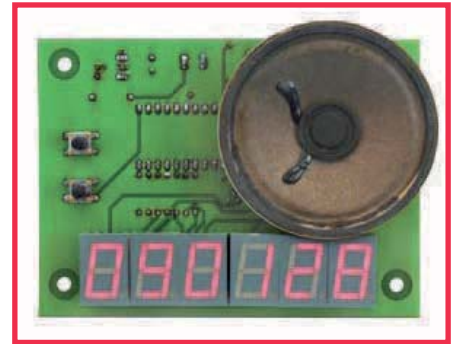
Flugfunk-Scanner

Für die Freunde der Luftfahrt können die Funkgespräche zwischen den Cockpits und dem Tower durchaus spannender als die Tagesnachrichten sein. Der Flugfunk-Scanner macht nicht nur das Mithören des Funkverkehrs im Flugfunkband möglich, er ist sogar vom PC aus über USB steuerbar. Ein FM-Empfängerbaustein, auf dem ein Einfachsuper fast vollständig integriert ist, empfängt die AM-Signale des Flugfunks. Als NF-Ausgangssignal wird das Signal genutzt, das normalerweise die Feldstärke-Anzeige steuert.



Mini-Multitester

Das kennen Sie bestimmt: Beim Entwickeln eines Mikrocontrollerprojekts müssen Sie Spannungen messen, logische Signalzustände erkennen, Impulse zählen und ähnliches mehr. Der Mini-Multitester ist ein handliches Allround-Talent, das viele nützliche Funktionen vereint: Voltmeter, Logik-Tester, Impulszähler, TTL/UART-Sender und -Empfänger und noch einiges mehr. Lassen Sie sich überraschen!



Kirchturm- oder Kuckucksuhr?

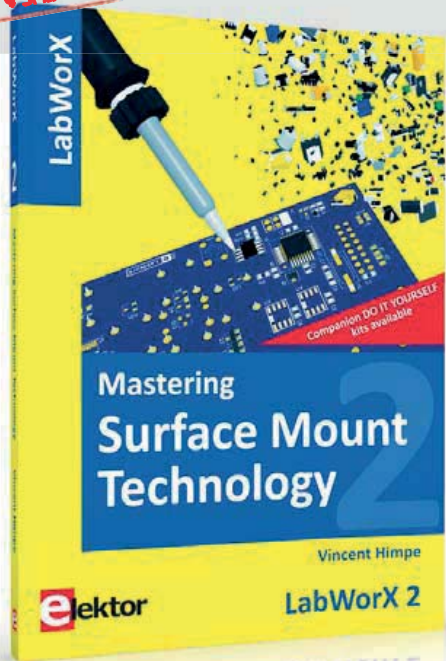
Selbstgebaute elektronische Uhren können heute eigentlich nur noch durch kreative visuelle Designs oder originelle akustische Begleiteffekte punkten. Die Uhr, die wir in der nächsten Ausgabe vorstellen, erfüllt diesen Anspruch: Abhängig vom Programm, das den Mikrocontroller steuert, klingt dieser Chronometer entweder wie die berühmte Kuckucksuhr aus dem Schwarzwald oder wie die Uhr im Turm eines Gotteshauses. Zu jeder Viertelstunde erinnert das Schlagwerk an die rasche Vergänglichkeit der Zeit...

Elektor Januar/Februar 2013 erscheint am 16. Januar 2013. Änderungen vorbehalten!

Rund um die Uhr und
sieben Tage die Woche

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



NEU

LabWorx 2

Mastering Surface Mount Technology

Neues Fachbuch in englischer Originalsprache

Der 2. Band aus der neuen LabWorX-Buchreihe ist ein Crash-Kurs mit Techniken, Tipps und viel Know-how, mit Hilfe dessen man SMD-Bauteile erfolgreich in die eigene Praxis integrieren kann. Selbst wenn man sparsam ist, kann man doch eigene Entwicklungen mit anspruchsvollen Fine-Pitch-Bauteilen realisieren.

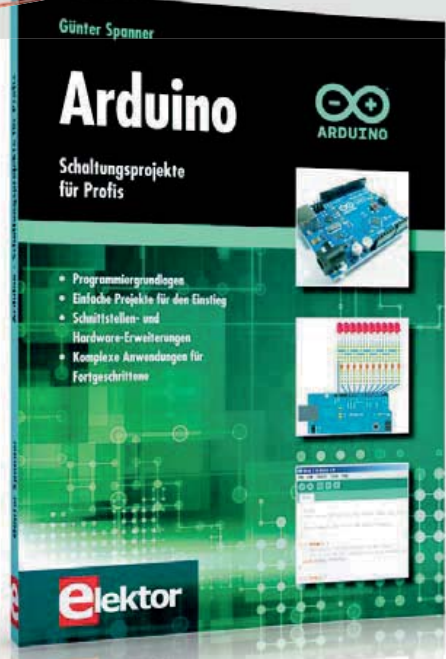
Neben der Erläuterung von Methoden und Ausrüstung wird besonderes Gewicht auf die Technologien hinter den SMD-Bauteilen und die Lötverfahren gelegt. Schritt für Schritt wird mit mehreren Projekten der Umgang mit SMDs erlernt und eingeübt. Mit der so gewonnenen Erfahrung kann man dann SMDs ohne große Kosten auch in eigenen Projekten sicher einsetzen.

LabWorX ist eine (englischsprachige) Praxis-Buchreihe, bei der jedes Buch ein ganz spezielles Thema aus der Welt der Elektronik aufgreift. Jede Ausgabe komprimiert Informationen, Anwendungen und nützliche Hinweise für die Praxis aus dem jeweiligen Themenfeld. Außer den Grundlagen der jeweiligen Technologie wird vertieft auf Beispiele und praktische Anwendungen eingegangen, die mit Tipps zur Fehlervermeidung ergänzt werden.

282 Seiten (kart.) · Format 17 x 23,5 cm · ISBN 978-1-907920-12-7
€ 34,50 · CHF 42,80



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

NEU

Arduino

Schaltungsprojekte für Profis

Für den großen Erfolg der Arduino-Plattform lassen sich zwei Ursachen finden. Zum einen wird durch das fertige Prozessor-Board der Einstieg in die Hardware enorm erleichtert; der zweite Erfolgsfaktor ist die kostenlos verfügbare Programmieroberfläche, die ohne Installationsprozeduren sofort einsetzbar ist.

Einfache Einstiegsbeispiele sorgen für den schnellen Erfolg. Eine komplizierte Auswahl von Parametern, wie etwa Prozessorversion oder Schnittstelleneinstellungen sind nicht erforderlich. Erste Beispielprogramme können innerhalb weniger Minuten auf das Arduino-Board geladen und getestet werden.

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. Hier fehlen oft detaillierte Beschreibungen und die mehr oder weniger gut beschriebenen Projekte führen eher zu Verwirrung. Ein klar erkennbarer roter Faden fehlt, da die Anwendungen natürlich von einer Vielzahl verschiedener Personen erstellt wurden, die alle jeweils ein spezielles Ziel vor Augen hatten. 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.) · Format 17 x 23,5 cm · ISBN 978-3-89576-257-4
€ 39,80 · CHF 49,40



Weitere Infos & Bestellung unter www.elektor.de/arduino-buch



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

+49 (0)4422 955-333

+49 (0)4422 955-360

**Professional quality
@ discount prices!**

Bauelemente
Werkstattbedarf
Netzwerk- & PC-Technik

Sicherheitstechnik

Sat- und TV-Technik
Stromversorgung

Fingerprint- und RFID-Zugangskontroll-System

Wer darf rein? Wer bleibt draußen? – Speichern Sie bis zu 1.000 Fingerabdruck-Einträge und bis zu 10.000 RFID-Karten.

- Stand-alone-System mit internem Speicher für bis zu 30.000 Logins oder Netzwerkbetrieb
- direkt in Ethernet-Computernetzwerk einbindbar
- Wetterfest, IP54



LAN

USB

IP 65
OUTDOOR

149,00

GRAND MA 300



ID-CARD-READER



FINGERPRINT-IDENTIFICATION



GRANDING

Ihren Schlüssel können Sie vergessen!

Zutritts-Kontrollsystem mit Gesichtserkennung und Fingerprint- und Kartenleseeinheit

Mit 3" (7,6cm) TFT-Touchscreen für intuitive Bedienung

- Stand-alone- oder Netzwerkbetrieb
- Infrarot-System für die Erkennung in dunkler Umgebung
- eingebautes Relais
- integrierte Leseinheit für Karten

Kapazität:

3000 Gesichtsprofile
5000 Fingerabdrücke
100.000 Log-Einträge

Schnittstellen:

TCP/IP, RS232/485,
USB Host, Wiegand-in,
Wiegand-out



ID-CARD-READER



GESICHTS-ERKENNUNG



FINGERPRINT-IDENTIFICATION



Bestellnummer:

GRAND MB 700 ID

399,00

Kartenleser

ID-Card 275 x 100 x 185 mm

Maße



ID Chipkarten & Transponder

- weiße PVC-Karte mit gekapselter Chip
- Schlüsselanhänger mit Transponderscheibe

Bestellnummer:

GRAND EM THIN
RFID-TP 35

Typ

ID-Card
Anhänger

Frequenz

125 KHz
125 KHz

Maße

85 x 54 x 0,8 mm
Ø 35 x 43 x 5 mm