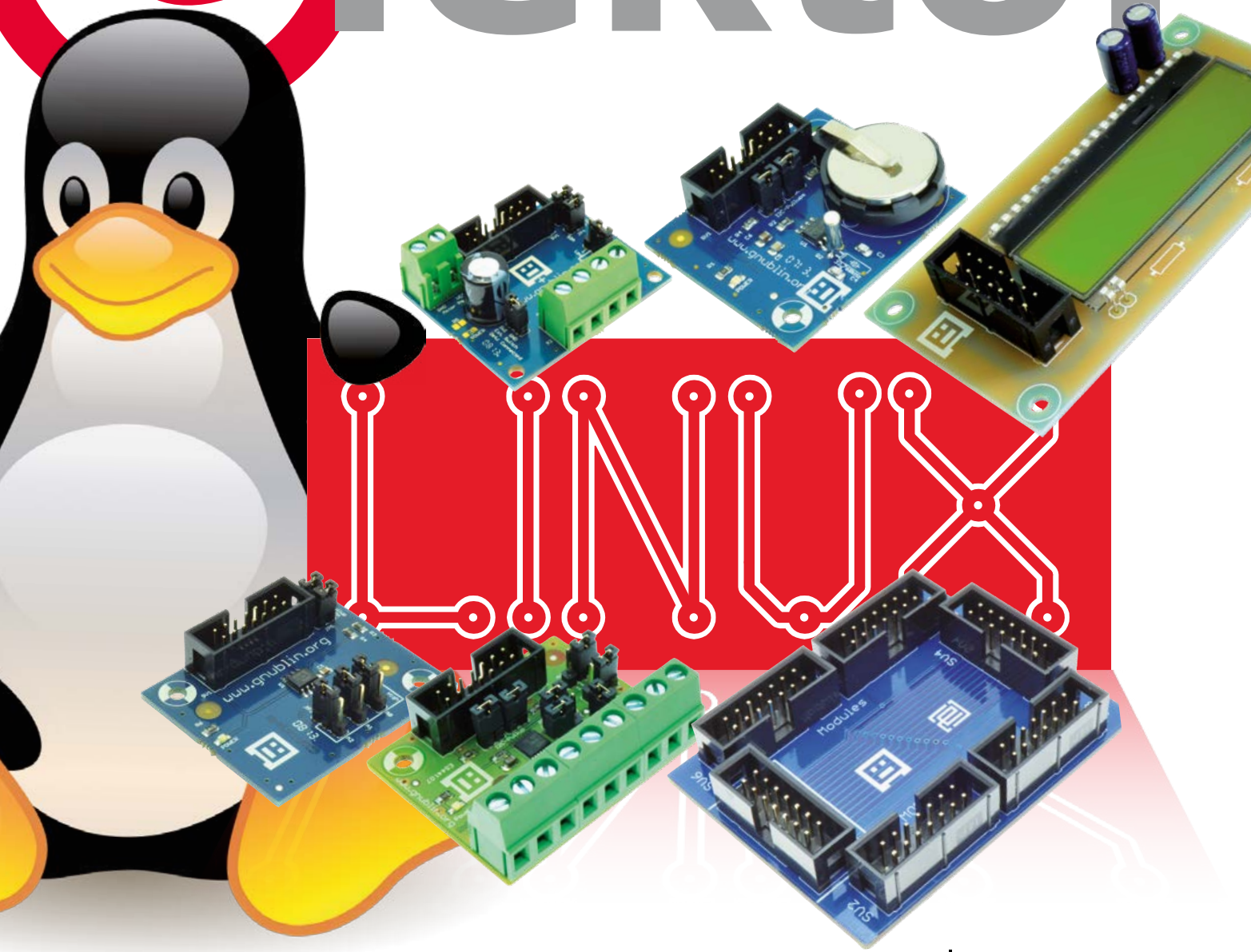
www.elektor-magazine.de

elektor magazine

September 2013 | Nr. 513

elektor



Kompakte Audio-Endstufe | Modulare Funkverbindung | **Fernauslöser mit Arduino** | **USB-Thermometer** | Programmierung für Windows und Linux | **Android-EKG** **Neues aus dem Labor** Retronik: Rauheitstester

G3078 E

(D) € 7,90 | CHF 14,80 | (A, B, L) € 8,90

4 198630 307906 09

eSTORE®
Beta LAYOUT

Entwickeln, Löten, Bestücken

€ 38,95*

Raspberry Pi Modell B
512 MB RAM

Iteaduino Leonardo V1.0, kompatibel

€ 18,60*

30 MHz, 2 CH Digital-Speicheroszilloskop, 250 MSa/s

€ 470,00*

Big Beta-Reflow-Kit

€ 129,00*

ERSA Lötstation i-CON NANO

€ 198,00*

*inkl. MwSt. und zzgl. Versandkosten

eSTORE® ist eine eingetragene Marke der Beta LAYOUT GmbH

www.beta-eSTORE.com

Beta LAYOUT
create:electronics

PROTOTYPES
Beta LAYOUT

Entwerfen, Bestellen, Anfassen

3D-Druck online

mit neuester Lasersinter-Technik

Ab sofort stellen wir alle 3D-Modelle im Hi-Tech Lasersinterverfahren her.

Die Vorteile:
- Höhere Präzision
- Glattere Oberflächen
- Feinere Strukturen
- Komplexe Modelle sind realisierbar
- Flexibel bei Wandstärken von nur 0,4 mm - 2 mm

www.beta-prototypes.com

Beta LAYOUT
create:electronics

Neues Angebot für ECAD-MCAD Integration

Mit Altium Designer können Sie jetzt viel einfacher mit Ihrem MCAD-Tool kommunizieren und Änderungen in beide Richtungen übertragen, z.B. mit:

- SolidWorks
- Solid Edge
- Pro/E
- Inventor

Gratis Webinar, Testversion und weitere Details: http://go.altium.com/elektor

Informationen unter Tel. +49 721 8244 300 / sales.de@altium.com

Altium

NEU

Raspberry Pi

45 Experimente mit Hard- und Software für Elektroniker

Der Raspberry Pi stellt einen sehr preiswerten, aber doch vollständigen Computer dar, an den auf einfache Weise verschiedenste Elektronik angeschlossen werden kann. Dieses Buch geht auf eine der Stärken des Raspberry Pi ein: die Kombination von Elektronik und Software.

Dieses Buch beschreibt 45 spannende und interessante Projekte, wie zum Beispiel ein Wechselblinklicht, eine Motorregelung, ein digitales Thermometer, ein Lichtmesser. Aber auch kompliziertere Projekte wie eine Motor-Geschwindigkeitsregelung, ein Webserver mit CGI (Common Gateway Interface) und Client-Server-Programme werden vorgestellt. Sie können dieses Buch als Projektbuch verwenden und die Projekte nachbauen, um sie dann in der Praxis einzusetzen. Durch die ausführliche Beschreibung mit Schaltplänen und Fotos gestaltet sich der Aufbau auf dem Steckbrett recht einfach.

Dieses Buch eignet sich auch gut als Lehrbuch. Bei jedem Projekt wird erklärt, warum es auf diese Art und Weise ausgeführt ist. Sie lernen viel über den Raspberry Pi, Python und die verwendeten Bauteile, so dass Sie selbst die Projekte anpassen, nach eigenen Wünschen erweitern oder mehrere Projekte miteinander kombinieren können.

Software-SD-Karte und Hardware-Starterkit zum Buch sind optional bei Elektor erhältlich.

272 Seiten (kart.) • Format 17 x 23,5 cm • ISBN 978-3-89576-273-4

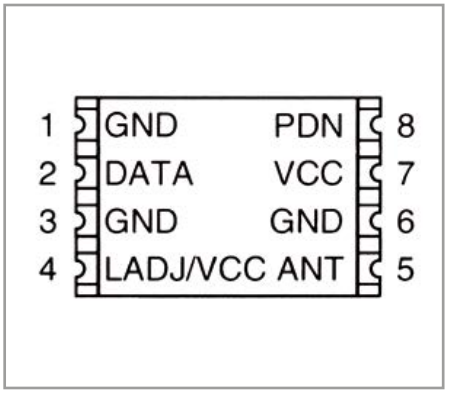
€ 39,80 • CHF 49,40

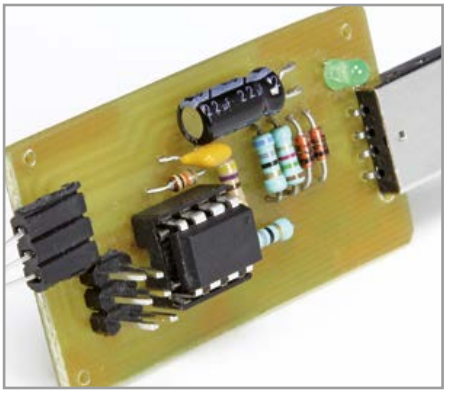
elektor

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

Inhalt

elektor magazine





Community

6 Impressum

10 Elektor World
Neues aus der Elektor-Community

Industry

8 Aktuell
Neue Produkte

Labs

42 DesignSpark: Libraries
In diesem Teil des Tutorials geht es darum, wie man Libraries zur Schaltplan- und Layout-Erstellung in DesignSpark richtig einsetzt.

54 Elektor.Labs
Wir zeigen Tipps und Tricks, um mehr aus Ihrer Online-Publikation zu machen…

56 90° und steigend
Beim Ausprobieren eines Batterietesters war das Elektor-Labor überrascht von der Temperatur eines 7-W-Serienwiderstands...

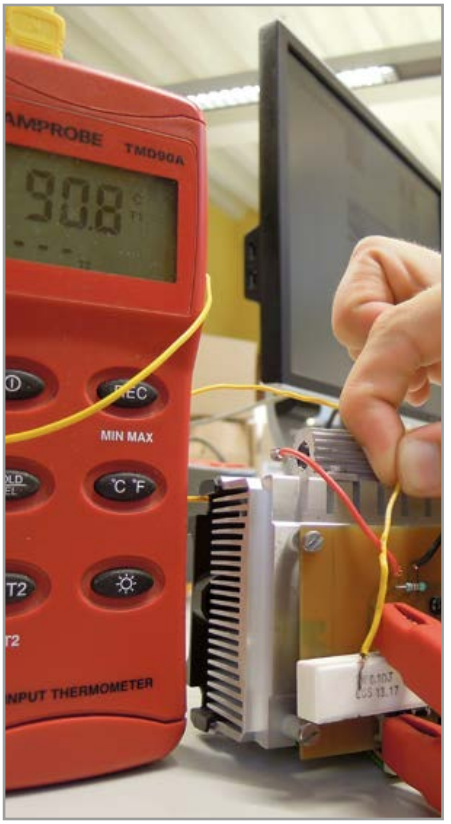
Projects

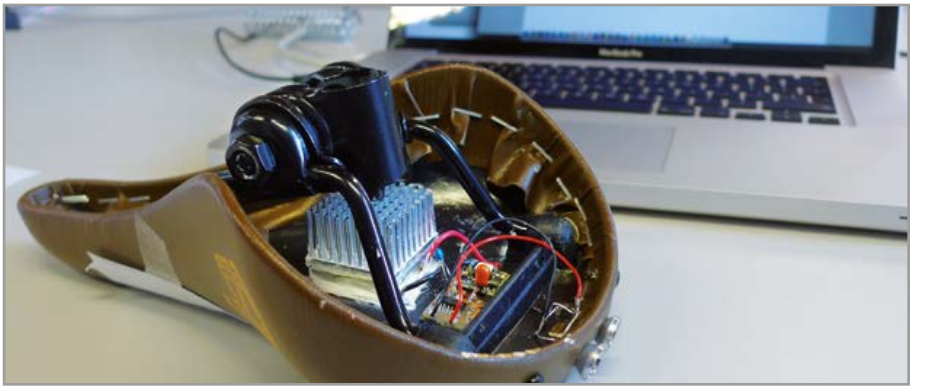
12 Für Linux-Board und Raspberry Pi
In der letzten Ausgabe haben wir bereits eine Relais-Platine vorgestellt, die sich an das Elektor-Linux-Board, den Raspberry Pi und weitere Mikrocontrollerboards anschließen lässt. Doch das war nur ein Beispiel aus einer breiten Palette von Erweiterungsplatinen. Praktische Kommandozeilen-Tools und eine C/C++-API ergänzen das Ganze.

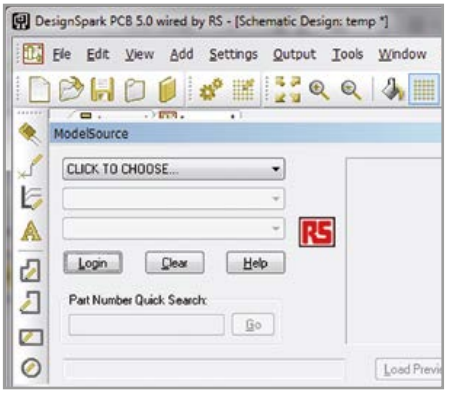
16 Kompakte Audio-Endstufe
Mit übersichtlicher Schaltungstechnik und kompaktem Aufbau leistet die Endstufe mehr als 200 W an 4 Ω. Die äußerst niedrigen Verzerrungen sind dem Einsatz eines speziellen Treiberbausteins von Texas Instruments zuzuschreiben.

26 USB-Thermometer
Wenn man heute eigene Elektronik mit einem PC verbinden will, kommt USB zum Zuge. Doch die PC-Seite verlangt nach einem passenden Treiber. Dieses Projekt zeigt, wie man das Problem elegant löst.

32 Elektor Android EKG (2)
Das Elektor Android EKG wird zusammen mit einem Android-







Smartphone oder -Tablet zu einem mobilen Kardioskop mit Touchscreen-Bedienung. Dieser Teil behandelt die Software und die Funktionen des Mikrocontrollers. Ein weiteres Thema ist die Android-App, die für die Bedienung und Darstellung zuständig ist.

46 Modulare Funkverbindung mit Manchester-Code (1)
Hier geht es um ein Verfahren, Daten per Funk in den ISM-Bändern 315 MHz oder 433 MHz mit einer Datenrate von 5.000 bit/s zu übertragen und dabei preiswerte aber zuverlässige Komponenten zu verwenden. Mit dem Manchester-Code wird ein Protokoll implementiert, mit dem Entfernungen bis zu 200 m überbrückt werden können.

58 Bewegungsgesteuerter Fernauslöser - mit Arduino
Ein preiswertes Nachtlicht mit Bewegungssensor, eine Infrarot-LED, nur wenige passive Bauelemente und dazu ein passend programmiertes Arduino-Board: Daraus entsteht ein System, das bei Bewegungen eines Objekts drahtlos eine Fotokamera auslöst.

64 Schnell noch speichern 2.0
Beim Ausschalten oder sonstigen Spannungsausfällen ist es bei manchen Schaltungen mit Mikrocontroller sinnvoll, den aktuellen Status oder bestimmte Werte in dessen EEPROM abzulegen. Zur Detektion dieses Spannungsabfalls wurde bereits eine Schaltung in Elektor veröffentlicht; hier kommt eine verbesserte und sogar noch einfachere Lösung.

66 Programmierung mit Mono
Elektroniker müssen zunehmend auch Programme für den PC erstellen. Häufig will oder darf man sich dabei nicht auf ein Betriebssystem festlegen. Mit dem Open-Source-Framework „Mono“ lassen sich Anwendungen für Windows, Linux, Mac OS und sogar für Android und das iPhone entwickeln.

Magazine

72 Retronik
Philips PR9150/PR9151 Oberflächen-Rauheitstester

78 Hexadoku
Sudoku für Elektroniker

82 Vorschau
Nächsten Monat in Elektor

Aus technischen Gründen mussten die Projekte „Xmega-Webserver-Board“ und „Numitron-Uhr“ auf die nächste Ausgabe verschoben werden.

Impressum

44. Jahrgang, Nr. 513 September 2013
Erscheinungsweise: 10 x jährlich
(inkl. Doppelhefte Januar/Februar und Juli/August)

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

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

Hauptsitz des Verlags
Elektor International Media
Allee 1
NL-6141 AV Limbricht

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. 43 ab 01.01.2013

Distribution:
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

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

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

Linux-API / BeagleBone Black / Kompakte Endstufe

Viele Leser dürften ja inzwischen wissen, dass ich selbst begeistert programmiere. Mit Interesse verfolge ich daher, was Benedikt Sauter und sein Team gerade auf die Beine stellen. Zum Gnublin/Elektor-Linux-Board gibt es inzwischen eine stattliche Anzahl von Erweiterungsplatinen; dazu brauchte es – wie jeder weiß, der selbst schon programmierbare Hardware entwickelt hat – natürlich auch Unterstützung in Form von Software (Seite 12). Mit der C/C++-API von Benedikt und seinen Mitstreitern möchte ich demnächst selbst einmal Anwendungen entwickeln. Wie versprochen wurde, komme ich damit auch als absoluter Linux-Anfänger schnell zum Ziel. Neu – und neben den Erweiterungsboards ebenfalls bald im Elektor-Shop erhältlich - ist eine Adapterplatine für den Raspberry-Pi-Konkurrenten „BeagleBone Black“ (rund 50 Euro). Man wird das Relais-Board, den Schrittmotortreiber und die anderen Erweiterungsplatinen also auch dort anschließen können. Mein Kollege Thijs Beckers hat übrigens gerade ein BeagleBone-Black-Board auf dem Tisch liegen. Heute hat er mir vorgeführt, wie leicht sich ein Fernseher damit zu einem Smart-TV aufrüsten lässt, das dann zum Surfen und vielem mehr genutzt werden kann. Über seine Erfahrungen wird Thijs in der nächsten Ausgabe berichten.

Auch für die Liebhaber von analogen Schaltkreisen haben wir in diesem Heft etwas Schönes zu bieten: Es handelt sich um eine Endstufe mit 200 W Ausgangsleistung (an 4 Ω), die komplett im Elektor-Labor entwickelt wurde. Mein Kollege Ton Giesberts hat dabei den Verstärker nebst Schutzschaltung auf einer Platine von nur 8,5 mal 13 cm untergebracht. Warum er sich dabei für ein halbdiskretes Konzept entschieden hat, lesen Sie ab Seite 16.

Jens Nickel

Unser Team

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

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

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

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

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

USA
Hugo Van haecke
+1 860-289-0800
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
+35 19 14 46 55 77
j.martins@elektor.com

Portugal
João Martins
+35 19 14 46 55 77
j.martins@elektor.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
zkoksal@beti.com.tr

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

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

Unser Netzwerk

Die Elektor-Community

Unsere Partner und Sponsoren

Altium — *www.altium.com* 3

Batronix — *www.batronix.com/go/26* 31

Beta Layout — *www.beta-prototypes.com* 2

Intelligent Soc s.l. — *www.soclutions.com* 53

LeitOn — *www.leiton.de* 53

Linx Technologies — *www.linxtechnologies.com* 9

LPKF — *www.lpkf.de/prototyping* 45

Reichelt — *www.reichelt.de* 84

Schaeffer AG — *www.schaeffer-ag.de.* 63

Sie möchten Partner werden?

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

ElektorLive!

Am 12. Oktober 2013 findet unser Seminar- und Ausstellungstag „ElektorLive!“ zum zweiten Mal statt, und zwar im Congress Park in Hanau bei Frankfurt am Main. Für die Profis und die Hobbyisten unter unseren Lesern haben wir ein attraktives Seminarprogramm auf die Beine gestellt:

- Android-Grundlagen (eigene Apps entwickeln)
- LabVIEW für den Praktiker (Anschluss eigener Elektronik)
- Rapid SMD-Prototyping (PCB-Pool und Target 3001!)
- Röhrenverstärker (mit Menno van der Veen)
- AVR-Mikrocontroller (Tipps und Tricks)
- Schaltplan- und Layout-Design mit EAGLE v6
- LPKF ProConduct (Durchkontaktierungen ohne Chemie)
- Raspberry-Pi-Grundlagen (Einsatz in eigenen Projekten)
- Arduino-Praxisprojekte (Programmieren und Anschluss eigener Hardware)
- Sensoren ans Internet (Entwickeln mit den TCP/IP-Modulen von WizNet)
- ARM Cortex-M0-Mikrocontroller (Einstieg und Programmierung)
- FPGA-Erweiterungsboard (Peripherieplatine für das Elektor FPGA-Board)
- Anwendungen mit Raspberry Pi und dem Elektor-Linux-Board

Im Eintrittspreis inbegriffen sind zwei Seminare. Insgesamt sind drei Seminarblöcke mit parallel stattfindenden Seminaren geplant. Einer der Zeitslots dauert in diesem Jahr zwei Stunden, was den Dozenten Gelegenheit gibt, das jeweilige Thema noch etwas weiter zu vertiefen. Begleitet wird das Ganze wieder durch eine Ausstellung, auf der man sich über neue Elektronik-Produkte informieren kann. An den Ständen wird es wie im letzten Jahr interessante Demos geben!
Der Eintrittspreis beträgt für Abo-Mitglieder 29,50 Euro (49,50 Euro für Nicht-Mitglieder), Studenten, Azubis und Schüler zahlen in jedem Fall nur 19,50 Euro.
Unter folgender Internetadresse kann man sich anmelden:

www.elektor.de/live-anmeldung

Preiswerte TCP/IP-Lösung

WIZnet stellt mit dem W5500 einen neuen Internet-Interface-Chip vor, der die Produktpalette nach unten um eine preiswerte (unter 2 Euro) und coole (wird nur ca. 45 °C warm) SPI-Version ergänzt.
Mit dem 10/100-Mbit-fähigen W5500 wird jetzt eine Lösung geboten, die preislich klassischen MAC-PHY-Ethernet-Controllern entspricht. Vergleicht man den W5500 mit einem reinen MAC-PHY-Ethernet-Controller, dann bekommt man hier das TCP/IP-Protokoll (acht parallele Engines) und 32 KB Dual-Ported-RAM als Buffer dazu. Der Zugriff auf den MAC-Layer ist für spezielle industrielle Protokolle aber trotzdem gegeben. Zudem ist Hardware-TCP/IP bezüglich Performance und Sicherheit den Software-Lösungen überlegen. Samples des W5500 sind ab sofort verfügbar, ab September ist der Chip in Stückzahlen lieferbar. Es wird sogar ein W5500-Arduino-Ethernet-Shield geben.

Ultrakompakte RL78/G10-MCUs

Besonders für die Steuerung batteriebetriebener Kleingeräte geeignet sind die nur 3,6 x 4,4 mm großen Ultra-Low-Power-MCUs der neuen RL78/G10-Familie von Renesas im 10-Pin-SSOP-Gehäuse. Die Mikrocontroller sind je nach Version mit 1 bis 4 KB Flash-Speicher ausgestattet und lassen sich mit bis zu 20 MHz takten. Der maximale Stromverbrauch im Betrieb beträgt dabei lediglich 46 µA/MHz – im Stop-Modus sinkt der Strombedarf auf 0,56 µA.
Darüber hinaus zeichnen sich die Mitglieder der RL78/G10-Familie durch eine Vielzahl peripherer Funktionseinheiten wie 16-bit-Multifunktions-, 12-bit-Interval- und Watchdog-Timer, 10-bit-A/D-Wandler, I²C-, UART- und CSI/SPI-Interfaces, einen bereits auf dem Chip integrierten Hochgeschwindigkeitsoszillator und intelligente Sicherheitsfunktionen aus. Die für einen Betriebstemperaturbereich von -40 bis +85 °C spezifizierten RL78/G10-MCUs arbeiten mit 2,0 bis 5,5 V Versorgungsspannung.

www.renesas.com/products/mpumcu/rl78/rl78g1x/rl78g10/index.jsp

Wo gibt's die neue Elektor?

Elektor-Mitglieder bekommen das neue Heft 10 Mal im Jahr frisch ins Haus geliefert (www.elektor.de/mitgliedschaft). Wer kein Mitglied ist, kann die aktuelle Ausgabe im Elektor-Shop beziehen – versandkostenfrei! Darüber hinaus kann man die aktuelle Ausgabe in vielen Kiosken, Buchhandlungen und im Zeitschriftenhandel kaufen. Wo genau, das verrät Ihnen eine clevere Suchmaschine unter dem Link www.pressekaufen.de. Alle Verkaufsstellen bekommt man hier auf einer interaktiven Karte angezeigt.

Thermoelement-ADC mit 1-Wire-Interface

Maxim gibt die Bemusterung der kaltstellenkompensierten Thermoelement-Digital-Wandler MAX31850/MAX31851 mit 1-Wire-Schnittstelle bekannt. Die Bausteine kommen auf eine Genauigkeit von ±2 °C (die Sensor-Nichtlinearität nicht eingerechnet). Sie enthalten alle Funktionen, die für eine Thermoelement-Digital-Umwandlung benötigt werden.

Mit dem 1-Wire-Interface der Bausteine MAX31850/MAX31851 ist es möglich, eine einzige Datenleitung für die Kommunikation und Stromversorgung mehrerer Sensoren zu nutzen, was den Verdrahtungsaufwand deutlich reduziert.

Die Digitalisierung von Thermoelement-Signalen kann zu einer echten Herausforderung werden, da Thermoelemente nur geringe Spannungen abgeben und eine Kaltstellen-Kompensation erfordern. Mit den Wandlern steht die erste Lösung zur Verfügung, die einen diskreten ADC, eine hochpräzise Referenz, einen Temperatursensor für die Kaltstellen-Kompensation und eine 1-Wire-Busschnittstelle auf einem Chip integriert. Die Wandler können mit verschiedenen Thermoelement-Typen (K, J, N, T und E bzw. R und S) verwendet werden. Die ICs erkennen Unterbrechungen und Kurzschlüsse am Thermoelement, die zu falschen Temperaturwerten oder Systemausfällen führen können.

www.maximintegrated.com/datasheet/index.mvp/id/7953

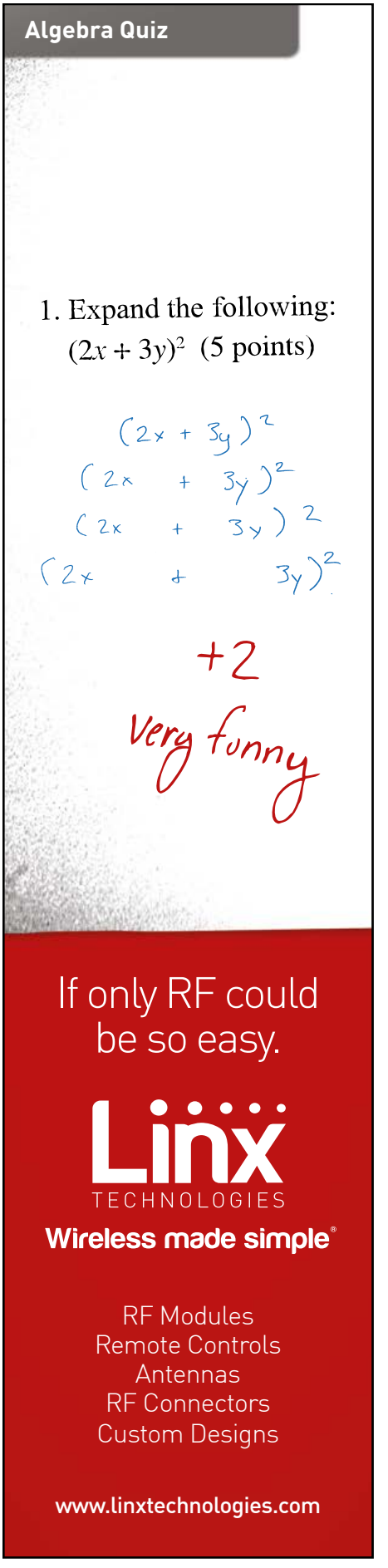
Updates und Korrekturen

Digitaluhr mit Sound

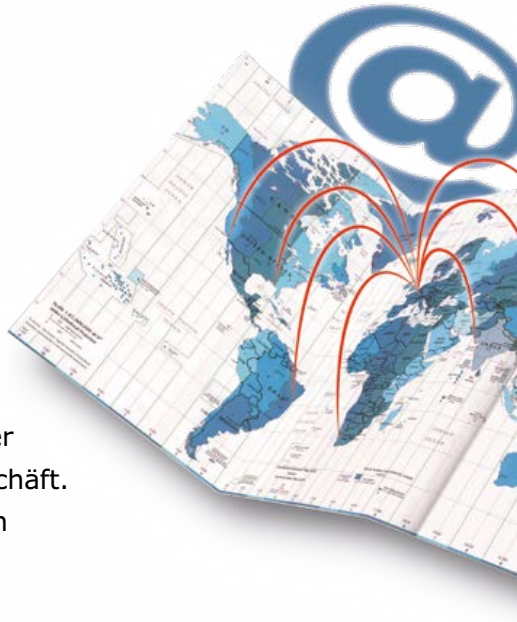
Elektor 1-2/2013, S. 84 (120276)

In der Schaltung ist T2 - der MOSFET BS250 - falsch eingezeichnet: Drain und Source wurden vertauscht. Auf der Platine und dem Bestückungsaufdruck ist hingegen alles richtig. Vom BS250 gibt es wohl auch eine flachere Version, als die im typischen TO-92-Gehäuse, wobei hier ebenfalls Drain und Source vertauscht sind. Anders sind die Rückmeldungen nicht zu erklären, die besagen, dass T1 und T2 heiß werden und ein „Zirpen" zu hören ist. Bei einem verpolten BS250 fließt durch seine parasitäre Diode Strom, wenn T1 eingeschaltet wird. Es kommt nun darauf an, ob man die „halbrunde" oder „stärker runde" Ausführung des BS250 verwendet. Sicherheit erhält man durch Messen mit einem Multimeter im Dioden-Prüfmodus. Zunächst schließt man beim BS250 alle drei Pins kurz, um das Gate zu entladen. Bei der Messung ist dann Source die Kathode der parasitären Diode und Drain die Anode. Der dritte Pin ist dann folglich das Gate. Im Leser-Forum („Meine Schaltung funktioniert noch nicht") wurde das Thema ausführlich besprochen.

Anzeige



Elektor World

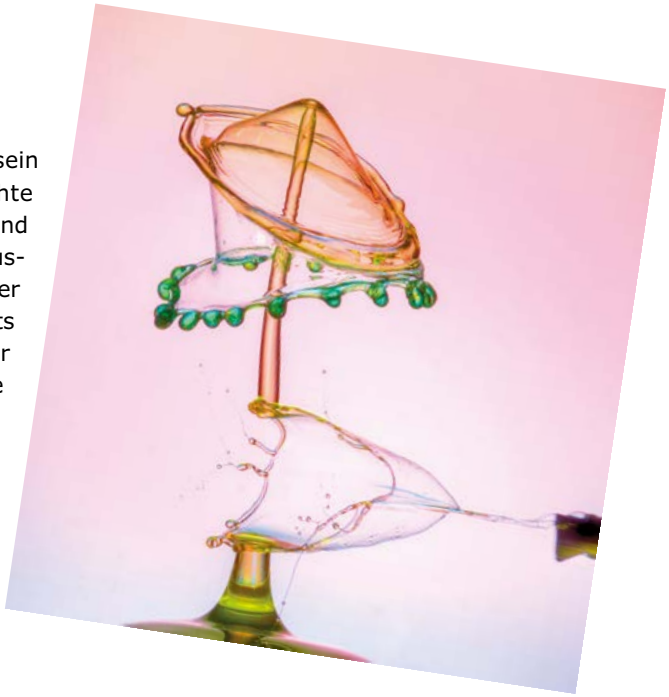


Zusammengestellt von
Wisse Hettinga

Täglich, stündlich - ja in jedem Moment denken sich Elektroniker auf der ganzen Welt etwas aus, optimieren, analysieren oder entwickeln Neuartiges. Motivation ist sicherlich der Spaß daran, doch manchmal wird daraus ein Geschäft. Wir helfen gerne bei der Verbindung dieser beiden Sphären: Fun + Business.

Yes He CAN

In Vorbereitung eines CAN-Projekts hat Kollege Jan Visser aus dem Elektor-Labor den Projekt-Entwickler Hugo Stiers eingeladen, um einige Aspekte zu diskutieren. Daraus wurde ein ganzer Nachmittag, der sich nur um CAN drehte! Hugo ist ein CAN-Experte und war früher Ausbilder im Bereich DAF-LKWs. Er ist einer der Techniker, der nur glaubt, was er anfassen kann. Daher brachte er eine ganze LKW-Ladung an Schachteln, Platinen, Kabeln und Notebooks mit, die ihn im Außeneinsatz begleiteten.
Hugo und Jan zu beobachten war eine faszinierende Erfahrung. In Nullkommanichts materialisierten sie um sich herum ein eigenes Raum-Zeit-Kontinuum, indem eben die Zeit keine Rolle spielte. Das Ergebnis vieler Stunden Bastelei war: Alles funktioniert wie es soll! Jan machte sich dann an die letzten Details des CAN-Projekts, das in dieser Elektor-Ausgabe abgedruckt ist. An Hugo: vielen Dank für deinen Besuch im Elektor-Labor!



Fotoino

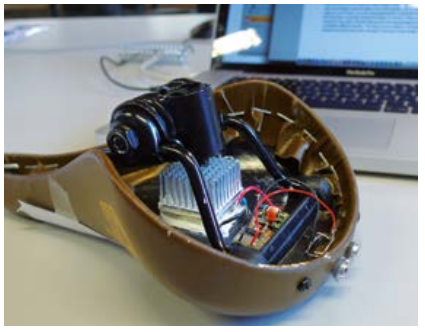
Falls Sie sich fragen sollten, wozu Arduino alles gut sein könnte: Elektor-Mitglied Huib Theunissen überraschte uns mit Schnappschüssen, die per Arduino ausgelöst und gesteuert wurden. Er verwendete alle sechs Board-Ausgänge zur Auslösung von Tropfen unterschiedlich gefärbter Flüssigkeiten, eines Gewehrschusses und des Blitzlichts seiner Kamera. Nachdem das Timing auf Bruchteile einer Sekunde genau stimmte, wurde dieses beeindruckende Foto aufgenommen.
Huib wählte dieses Foto mit dem Namen „The Speed of Life“ für uns aus, mit dem er schon einen ersten Preis bei einem Nikon-Fotowettbewerb gewonnen hatte. Glückwunsch! Auf www.facebook.com/druppelfotos findet man weitere Bilder von ihm. Nun liegt es an Ihnen, sich eine Anwendung für Arduino auszudenken.

Fotofinish

„The Kite“ in Oxford ist ein Pub direkt neben dem Bahnhof, der auch Zimmer für müde Reisende anbietet. Johan Dijk, einer unserer Manager, und ich waren mit der Nachbereitung eines fruchtbaren Treffens mit Vertretern von RS Components zugange. Wir diskutierten, wie wir unsere skizzierten Vorschläge in einen Vertrag verwandeln könnten. Die junge Frau hinterm Tresen hat uns dabei wohl zuhören müssen. Und sie war hilfsbereit: „Sie können sich beide mit dem Entwurf hierher setzen (auf den Tresen mit der Zapfanlage deutend) und dann mache ich ein Foto als Beweis.“ Und hier ist er!

Schlossgeist

Wir waren schon etwas überrascht, als Leute der LPI (League of Paranormal Investigators) um das Schloss herum liefen. Auf Geistersuche! Sie suchten nach feinstofflichen Resten französischer Soldaten, die hier im Schloss gestorben sind. Und sie wollten den Geist von Entgen Luyten finden, der letzten „offiziellen“ holländischen Hexe. Laut einer Sage treibt sie im Schlosskeller ihr Unwesen. Mart, seines Zeichens Fotograf bei Elektor, folgte den Geisterjägern und machte dabei einige Bilder. Und es zeigte sich etwas Ungewöhnliches. Mit Hilfe seines Kamera-Blitzes konnte er den gejagten Geist fixieren: „Da, direkt vor Ihnen!“



Popolicht

Die Niederlande sind ohne Fahrrad undenkbar. Tagsüber ist alles in Ordnung, doch in der Nacht kämpft man mit antiker Beleuchtung, wenn man der Polizei und einem Knöllchen entgehen will. Störend ist die Energiegewinnung per Pedal und Dynamo. Da klemmt gerne die Mechanik oder ist ein Kabel unterbrochen. Wouter Eisema von der Hanzehogeschool in Groningen (NL), einer technischen Fachhochschule, präferiert eine komplett andere Lösung. Die Wärme des Sattels (genau genommen: Ihres Hinterteils) wird von Peltier-Elementen zur Stromerzeugung genutzt, mit der LEDs hinten am Sattel versorgt werden. Clever, nicht wahr? In naher Zukunft werden wir das Projekt veröffentlichen.

Übermondrian

1987 veröffentlichten wir Mondrian, einen dreifarbigen Selbstbauplotter. Einer der vielen Leser, die Mondrian nachgebaut hatten, war der junge Künstler Jonas Vos. Es war sein erstes Elektor-Projekt und auch sein erster Versuch, mit Maschinen Kunst zu produzieren. Jetzt hat sich ein Dozent der Jan van Eyck Academy in Maastricht (NL) dazu entschlossen, seine eigene CNC-Fräsmaschine zu bauen – eine richtig große! Sie kann mit allen Arten von Materialien bis zum Format 1.800 x 2.400 x 800 mm umgehen! Studenten können damit wirklich große Objekte bearbeiten. Das Bild zeigt Jonas bei der Arbeit mit seiner Riesenmaschine.

Mehr zur Jan van Eyck Academy: www.janvaneyck.nl.

Für Linux-Board und Raspberry Pi

Erweiterungsplatinen, Kommandozeilen-Tools und eine C/C++-API

Von
Benedikt Sauter [1]

In der letzten Ausgabe haben wir bereits eine Relais-Platine vorgestellt, die sich an das Elektor-Linux-Board, den Raspberry Pi und weitere Mikrocontrollerboards anschließen lässt. Doch das war nur ein Beispiel aus einer breiten Palette von Erweiterungsplatinen, die das Team von Embedded Projects entwickelt hat. Praktische Kommandozeilen-Tools und eine C/C++-API für eigene Anwendungen ergänzen das Ganze.

Wenn man sich auf eine Stecker-Spezifikation einigt, lassen sich Controllerboards und Erweiterungsplatinen beliebig kombinieren. Über den 14-poligen Gnublin-Connector auf dem Elektor-Linux-Board, der sich als „Embedded Extension Connector“ auch auf dem in der nächsten Ausgabe vorgestellten Xmega-Webserver-Board befindet, haben wir in der letzten Ausgabe schon berichtet [2]. In diesem Artikel wollen wir weitere Extension-Boards vorstellen, aber vor allem auch zeigen, wie einfach das Ansprechen der Erweiterungen unter Linux ist.

Für einen schnellen Test der Funktionen hat das Entwicklerteam zu jedem Modul ein kleines Kommandozeilen-Programm gestrickt. Zum Entwickeln eigener Anwendungen wird außerdem eine komplette C/C++-API [3] bereitgestellt. Diese API basiert auf den Standard-Gerätetreibern für I2C, SPI, GPIO, ADC usw.; mit diesen muss sich der Anwendungsentwickler aber nicht mehr auseinandersetzen. Vielmehr kann er sein Gerät (zum Beispiel einen Schrittmotor) bequem über komfortable Funktionsaufrufe ansprechen (Bild 1). Aktuell arbeitet das Team auch an einer Python-API, mehr Infos dazu findet man unter [4].

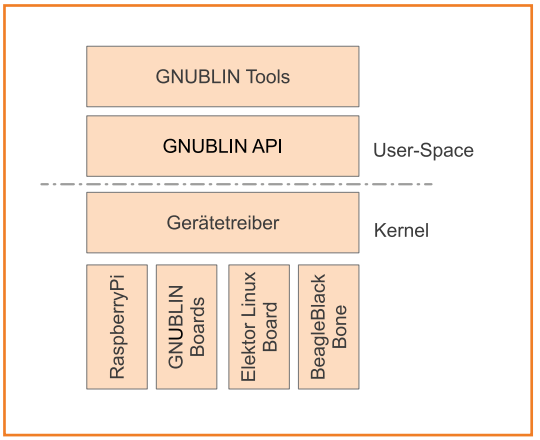


Bild 1.
Dank der Gnublin-API muss man sich nicht mehr mit den SPI-, I2C- und sonstigen Gerätetreibern auseinandersetzen.

Das Konzept

Linux ist eine sehr gute Abstraktionsschicht, um Anwendungen zu entwickeln, die unabhängig vom verwendeten Prozessor sind. Man entwickelt die Applikation einfach „für Linux“. Die neuen Erweiterungsplatinen, die unter **www.elektor.de/gnublin** erhältlich sind, machen dieses Konzept nun auch für Projekte mit Motoren, Displays, Temperatursensoren, Relais und vielem mehr zugänglich. Die Boards werden ganz einfach über ein Flachbandkabel mit dem Elektor-Linux-Board verbunden:

- Module-Relay (acht Relais schalten) (130212-91, Bild 2)
- Module-Temperature (Temperatursensor) (130212-95, Bild 3)
- Module-LCD 4x20 (vierzeiliges Text-Display) (130212-92, Bild 4)
- Module-Step (Schrittmotor-Treiber) (130212-

93, Bild 5)
- Module-IOExpander (16 digitale Ein- und Ausgänge) (130212-94, Bild 6)
- Module-Extension (Display, Taster, RTC, Piepser und Portexpander) (120596-91, Bild 7)

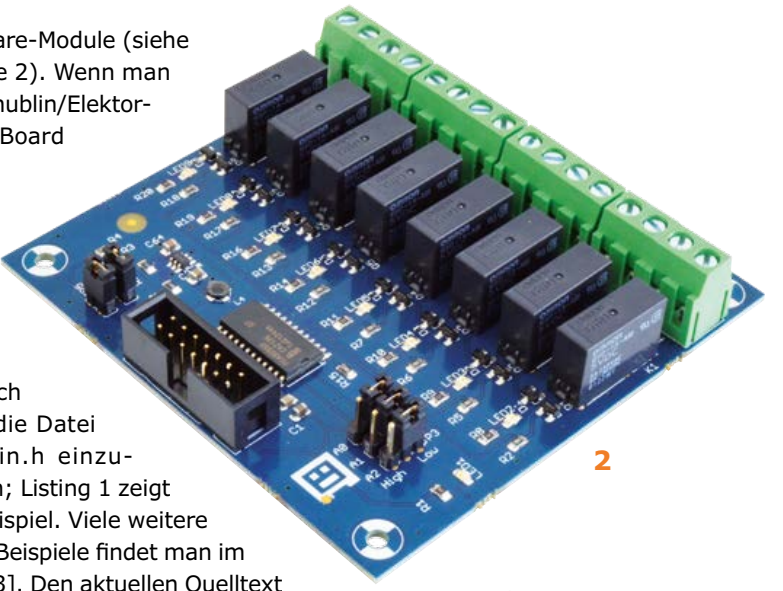
Zum Anschluss mehrerer Erweiterungsboards kann man bei Elektor eine Verteilerplatine unter der Nummer 130212-71 beziehen (Bild 8).
Wie schon in der letzten Ausgabe erwähnt, gibt es auch eine Adapterplatine zum Anschluss der Extension-Boards an den trendigen Kleinst-Computer Raspberry Pi (130212-72). Ganz neu ist eine Adapterplatine für das BeagleBone-Black-Board (130212-74).

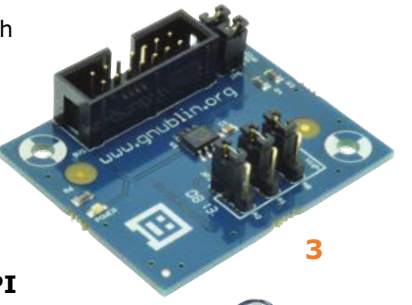
Die Kommandozeilen-Tools

Die Kommandozeile ist das A und O bei der Arbeit mit einem Linux-Board. Von hier aus kann man Anwendungen starten, stoppen, Linux administrieren, Systemmeldungen lesen und vieles mehr. Für die Gnublin-Erweiterungsplatinen und einige interne Funktionen des Linux-Boards hat das Entwicklerteam eigene kleine Kommandozeilen-Tools entwickelt. Gibt man auf der Kommandozeile gnublin- ein und drückt dann auf „Tab“, so werden automatisch alle diese Mini-Programme aufgelistet. In Tabelle 1 ist ein Auszug dargestellt. Die Tools sind für einen ersten Test sehr praktisch; so kann man sicher sein, dass die Hardware richtig angebunden ist. Man erspart sich eine ewige Fehlersuche in der Software, um dann etwa festzustellen, dass nur die Stromversorgung gefehlt hat. Jeder kennt das bestimmt!

Die C/C++-API

Hat man die Module angeschlossen und getestet, dann kann man mit dem Schreiben von Anwendungen beginnen. Bei der erwähnten C/C++-API werden durchweg einfach verständliche Funktions-Aufrufe verwendet, aber beispielsweise keine Pointer und Strukturen. Für alle Schnittstellen und Erweiterungsboards gibt es eigene Software-Module (siehe Tabelle 2). Wenn man das Gnublin/Elektor-Linux-Board nutzt, dann ist lediglich noch die Datei gnublin.h einzubinden; Listing 1 zeigt ein Beispiel. Viele weitere Code-Beispiele findet man im Wiki [3]. Den aktuellen Quelltext der kompletten API kann man sich unter [5] ansehen.
Unter [3] ist auch nachzulesen, wie man sich eine Entwicklungsumgebung aufbaut, um mit der API arbeiten zu können.


2



3

Installation der Tools & API

Im Prinzip kann man die API zusammen mit jedem Embedded-Linux-Board verwenden, das über Treiber für I²C und SPI verfügt. Denn die meisten Prozessoren haben diese Schnittstellen integriert, der Zugriff darauf ist fast immer als Gerätetreiber implementiert.
Nachfolgend zeigen wir, wie man die Tools und die API zusammen mit dem Elektor-Linux-Board und dem Raspberry Pi verwendet.

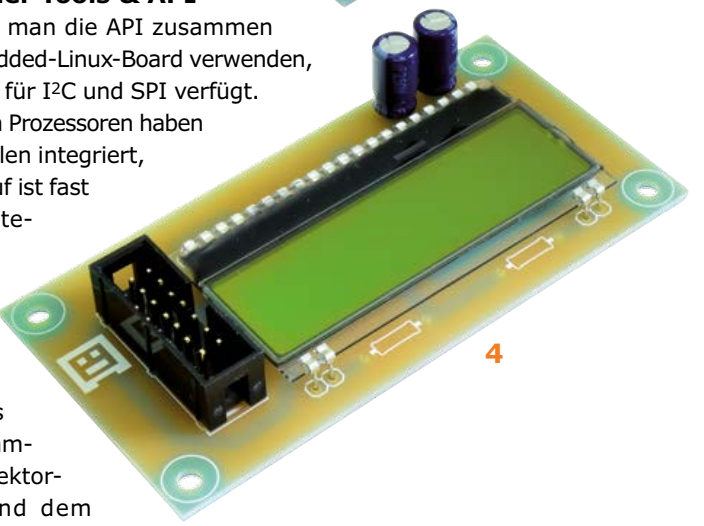
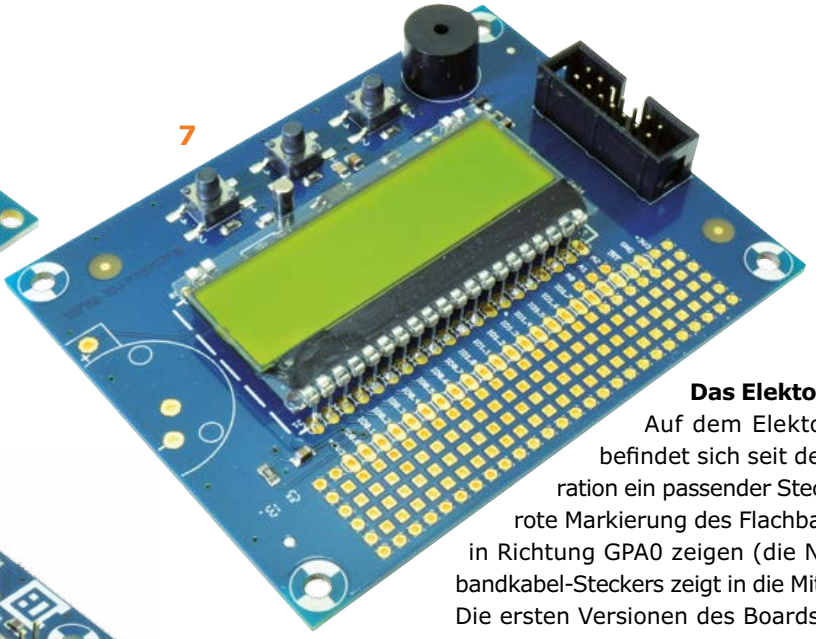

4

Tabelle 1: Gnublin-Tools (Auszug)

Tool	Beispiel Aufruf	Beschreibung
gnublin-lm75	Aufruf ohne Kommando	Ausgabe Temperatur
gnublin-relay	gnublin-relay -p 1- o 1	Relais 1 einschalten
gnublin-adcint	gnublin-adcint -c 1	Abfrage interner ADC, Kanal 1
gnublin-step	gnublin-step -p 3000	Fahre Schrittmotor auf Position 3000

5



7

6

Das Elektor-Linux-Board

Auf dem Elektor-Linux-Board befindet sich seit der ersten Generation ein passender Steckverbinder. Die rote Markierung des Flachbandkabels muss in Richtung GPA0 zeigen (die Nase des Flachbandkabel-Steckers zeigt in die Mitte des Boards). Die ersten Versionen des Boards haben wir mit einem ELDK-Dateisystem ausgeliefert, doch in der Zwischenzeit wurde komplett auf ein Debian-Image gewechselt. Wie man ein Update einer „alten" Speicherkarte macht, ist unter [6] beschrieben.

Das Compilieren der Tools für die Erweiterungsplatinen dauert auf dem Linux-Board gut und gerne fünf Minuten. Daher haben wir ein Debian-Paket vorbereitet, bei dem sich die Tools einfach installieren lassen.

Zuerst muss man sich das Debian-Paket herunterladen. Das geht per PC, danach muss man die Datei mit einem Kartenleser auf die SD-Karte speichern.

Ist man mit dem Board direkt online, so kann man das Paket aber auch per Kommandozeile herunterladen:

```
wget https://github.com/embeddedprojects/
gnublin-api/raw/master/gnublin-tools.deb
```

Die Installation der Tools geschieht so:

```
root@gnublin:~# dpkg -i gnublin-tools.deb
```

Will man das Paket wieder entfernen, so muss man eingeben:

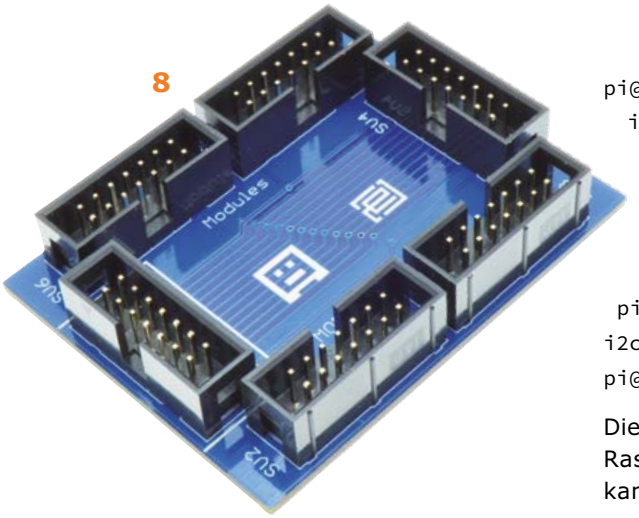
```
root@gnublin:~# dpkg -r gnublin-tools
```

RaspberryPI

Um die Software-Module auf dem Raspberry Pi nutzen zu können, verbindet man sich am besten direkt mit dem Quelltext-Repository. Der Raspberry Pi muss gestartet und mit dem Internet verbunden sein. Mit git kann man jetzt das Repo-

Listing 1: Ansteuerung eines I²C-Bausteins

```
#define BOARD_GNUBLIN
//#define BOARD_RASPBERRYPI

#include "gnublin.h"

int main()
{
   gnublin_i2c i2c;

   i2c.setAddress(0x42);        //i2c slave address

   char buffer[8];
   char RxBuf[8];

   buffer[0]=0x22;

   i2c.send(buffer,5);
   i2c.send(0x12, buffer, 2);   //send 2 bytes register 0x12

   i2c.receive(RxBuf, 3);       // read 3 bytes
   i2c.receive(0x23, RxBuf, 3); // read from register

}
```



8

sitory klonen. Sollte git auf dem Raspberry Pi nicht installiert sein, so macht man das wie folgt:

```
pi@raspberrypi ~ $ sudo apt-get install
git
```

Jetzt kann man das Repository abholen:

```
pi@raspberrypi ~ $ git clone https://
github.com/embeddedprojects/gnublin-api.
git
```

Nun wechselt man in das Verzeichnis …

```
pi@raspberrypi ~ $ cd gnublin-api
```

… und übersetzt und installiert dort die Programme, die Beispiele und die API:

```
pi@raspberrypi ~ $ make && sudo make
  install
```

Die API benötigt folgende Treiber:

```
    pi@raspberrypi ~ $ sudo modprobe
   spi-bcm2708
 pi@raspberrypi ~ $ sudo modprobe
i2c-bcm2708
pi@raspberrypi ~ $ sudo modprobe i2c-dev
```

Diese sind bereits in der aktuellen Version der Raspberry-Pi-Distribution enthalten. Optional kann man die Treiber fest in die Datei „/etc/modules“ eintragen (pro Zeile ein Modul):

```
spi-bcm2708
i2c-bcm2708
i2c-dev
```

Mit den kleinen Kommandozeilen-Tools kann man eine angeschlossene Erweiterungsplatine sofort testen.

(130212)

Weblinks

[1] sauter@embedded-projects.net

[2] www.elektor.de/130157

[3] http://wiki.gnublin.org/index.php/API

[4] http://en.gnublin.org/index.php/API_Python

[5] https://github.com/embeddedprojects/gnublin-api

[6] http://en.gnublin.org/index.php/GNUBLIN-Elektor

Tabelle 2: Software-API-Objekte

Modul	Schnittelle	Bemerkung
gnublin_gpio	intern	
gnublin_adc	intern	Aktuell nur bei Elektor-Linux-Board (nicht Raspberry Pi)
gnublin_i2c	I2C	Standard-I2C-Bus
gnublin_spi	SPI	Standard-SPI-Geräte
gnublin_pwm	intern	Aktuell nur bei Elektor-Linux-Board
gnublin_module_lm75	I2C	Temperatursensor
gnublin_module_relay	I2C	Relais-Karte
gnublin_module_pca9555	I2C	Portexpander mit 16 digitalen Ein-/Ausgängen
gnublin_module_step	I2C	Schrittmotor
gnublin_module_lcd	I2C	4x20-Zeichen-Display

Kompakte Audio-Endstufe

Satte Leistung, ultraniedrige Verzerrungen

Für alle Audio-Freunde präsentiert das Elektor-Labor einen neuen analogen Endverstärker, er wurde vollständig im eigenen Haus entwickelt. Mit übersichtlicher Schaltungstechnik und kompaktem Aufbau leistet die Endstufe mehr als 200 W an 4 Ω. Die äußerst niedrigen Verzerrungen sind dem Einsatz eines speziellen Treiberbausteins von Texas Instruments zuzuschreiben.

Von **Ton Giesberts**
(Elektor-Labor)

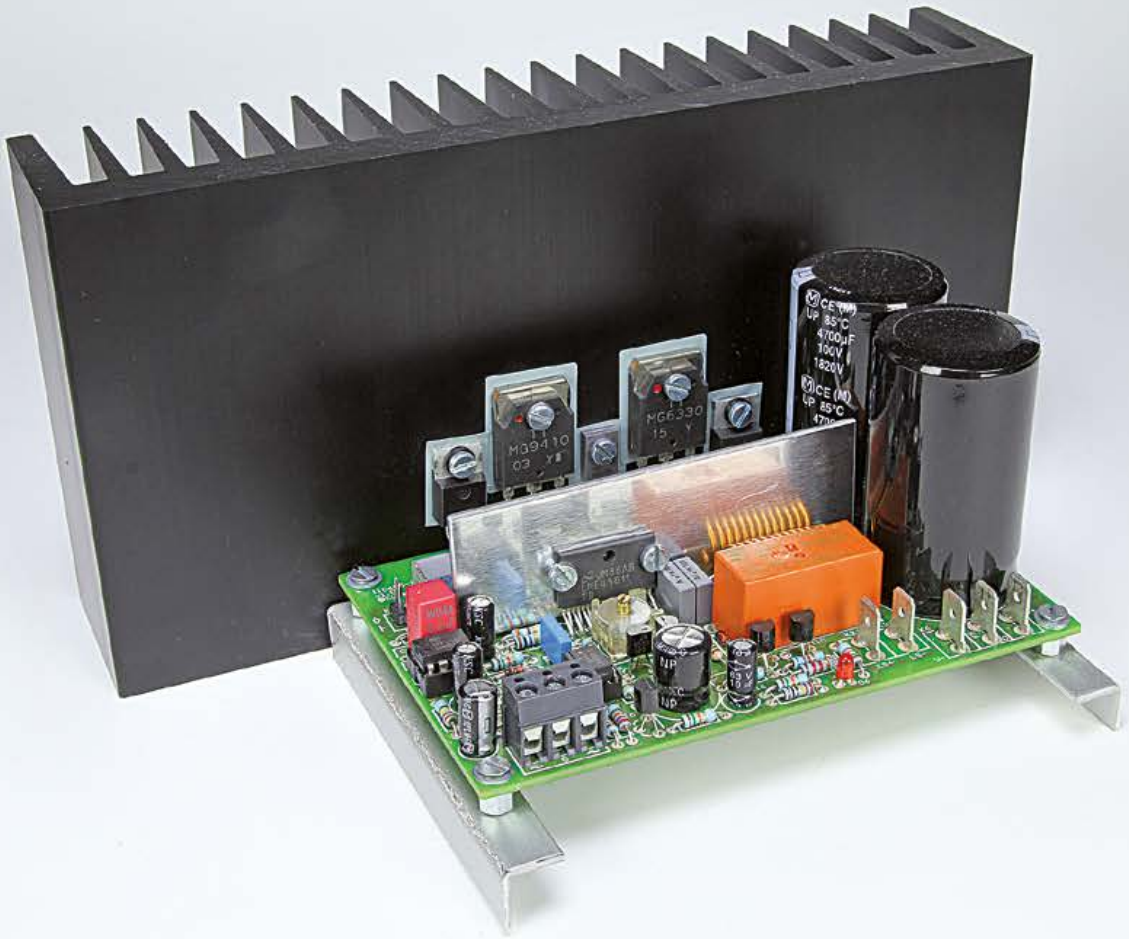
Audio-Endverstärker haben bei Elektor eine Jahrzehnte alte Tradition. Erinnert sei nur an den legendären „Edwin“ aus den siebziger Jahren, mit dem unzählige Audio-Enthusiasten erste Gehversuche unternahmen. In den vergangenen Jahren ist es auf diesem Terrain ruhiger geworden, was aber nicht bedeutet, dass das Interesse geschwunden wäre. Im Gegenteil, wir wissen, dass viele Leser auch heute ins Schwärmen geraten, wenn der in eigener Regie entstandene Audio-Verstärker seine Qualitäten entfaltet.

Da in der Sparte der diskret aufgebauten Audio-Endstufen kaum noch Neues zu erwarten ist, haben wir uns für ein halb-diskretes Konzept entschieden. Dieser Ansatz kann zwei schlagkräf-

Messwerte

(Stromversorgung mit Netztrafo 2 · 40 V~/500 VA (Nuvotem 0500P1-2-040) und Netzteil-Elkos 4 · 10 000 µF/100 V

• Eingangsempfindlichkeit:	0,88 V (137 W/8 Ω, THD+N = 0,1 %)
	0,91 V (145 W/8 Ω, THD+N = 1 %)
• Eingangsimpedanz:	15 kΩ
• Dauerausgangsleistung:	137 W an 8 Ω (THD+N = 0,1 %)
	145 W an 8 Ω (THD+N = 1 %)
	220 W an 4 Ω (THD+N = 0,1 %)
	233 W an 4 Ω (THD+N = 1 %)
• Spitzen- und Musikleistung:	218 W an 8 Ω (THD+N = 10 %)
(DC-Stromversorgung ±56,8 V)	175 W (8 Ω, THD + N = 1 %)
	165 W (8 Ω, THD + N = 0,1 %)
	395 W (4 Ω, THD + N = 10 %)
	316 W (4 Ω, THD + N = 1 %)
	299 W (4 Ω, THD + N = 0,1 %)
• Leistungsbandbreite:	2,1 Hz...125 kHz (50 W/8 Ω)
• Slew-Rate:	26,7 V/µs
• Anstiegszeit:	2,4 µs
• Signal-Rausch-Verhältnis:	> 94 dB (linear, B = 22 Hz...22 kHz)
(Referenzwert 1 W/8 Ω)	> 97 dBA
• Harmonische Verzerrungen	0,0033 % (1 kHz, 1 W/8 Ω)
+ Rauschen:	0,0006 % (1 kHz, 50 W/8 Ω)
(B = 80 kHz)	0,006 % (20 kHz, 50 W/8 Ω)



	0,0047 % (1 kHz, 1 W/4 Ω)
	0,0009 % (1 kHz, 100 W/4 Ω)
	0,009 % (20 kHz, 100 W/4 Ω)
• Intermodulationsverzerrungen: (50 Hz : 7 kHz = 4 : 1)	0,002 % (1 W/8 Ω)
	0,0009 % (50 W/8 Ω)
	0,003 % (1 W/4 Ω)
	0,0026 % (100 W/4 Ω)
• Dynamische IM-Verzerrungen: (Rechteck 3,15 kHz + Sinus 15 kHz)	0,0033 % (1 W/8 Ω)
	0,0022 % (50 W/8 Ω)
	0,0045 % (1 W/4 Ω)
	0,0027 % (100 W/4 Ω)
• Dämpfungsfaktor	560 (1 kHz/8 Ω)
	311 (20 kHz/8 Ω)
• Wirkungsgrad: (DC-Stromversorgung)	70,6 % (8 Ω, THD+N = 0,1 %)
	72,5 % (8 Ω, THD+N = 1 %)
	68,5 % (4 Ω, THD+N = 0,1 %)
	70,5 % (4 Ω, THD+N = 1 %)
• DC-Schutzschaltung:	+0,55 V/-0,86 V
• DC-Offset am Ausgang:	0,2 mV (max. 0,6 mV)
• Einschaltverzögerung:	6 s

tige Pluspunkte für sich verbuchen: Die Schaltung ist überschaubar und der Aufbau ist kompakt. Sorgfältige Auswahl der Komponenten vorausgesetzt, lassen sich Audio-Endverstärker mit überragenden Qualitäten realisieren.

Vorgeschichte

Es begann mit dem Messfilter für Klasse-D-Verstärker, das Elektor in der Juli/August-Ausgabe 2011 veröffentlichte. Dieses Filter war im Elektor-Labor entwickelt worden, um an Klasse-D-Ver-

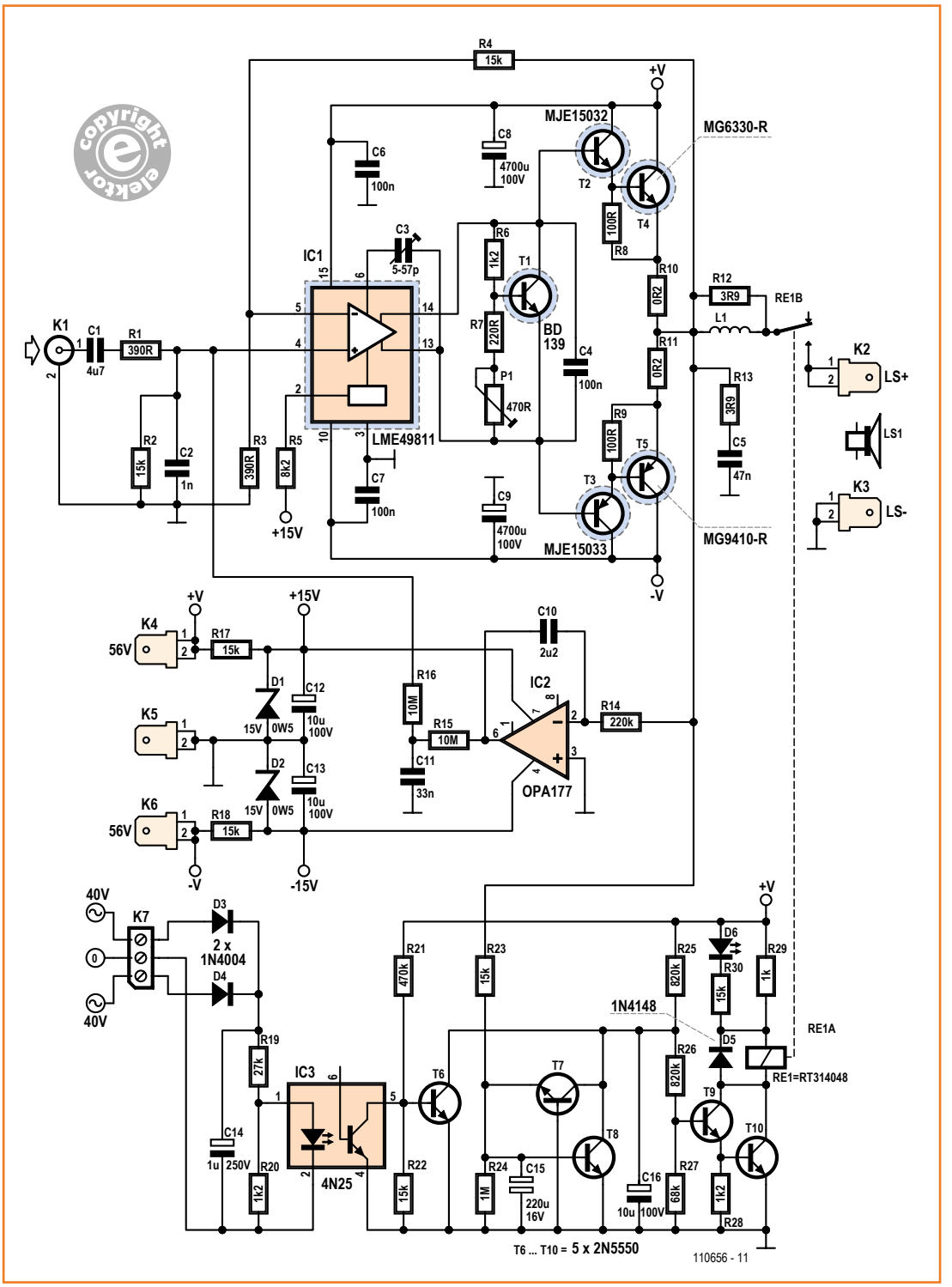


Bild 1.
Schaltung unseres neuen Audio-Endverstärkers. Trotz der mäßigen Anzahl der Bauelemente sind die Eigenschaften überragend!

stärkern hohe Ausgangsspannungen bis 70 V_{eff} messen zu können. Damals konnte das Filter nicht umfassend getestet werden, da ein geeigneter Klasse-D-Verstärker fehlte. Für die Elektor-Entwickler war diese Situation wenig befriedigend. Wir begannen, einen durchgängig diskret aufgebauten Endverstärker zu entwickeln, der an der hohen symmetrischen Betriebsspannung ±110 V arbeitete. Bestückt war dieser Endverstärker mit 23 Transistoren der Typen MJE340, MJE350, MPSA42 und MPSA92. Für den ersten Musteraufbau wurde sogar eine Platine entworfen. Zeitweise stagnierte das Projekt, denn die Materie war einigermaßen komplex. Allmählich drängte sich die Frage auf, ob das Testen eines Messfilters die viele Mühe rechtfertigte.

An den Testverstärker wurden hohe Anforderungen gestellt. Er musste Ausgangssignale bis 70 V_{eff} bei Frequenzen bis 20 kHz liefern, extrem niedrige Verzerrungen waren obligatorisch. Da die Eingangsimpedanz des Messfilters bei 1 kΩ lag, musste der Verstärker Spitzenausgangströme von mindestens 100 mA bereitstellen.

Später hielten wir nach einer weniger aufwendigen Alternative Ausschau. Gab es einen integrierten Baustein, der die hohe Ausgangsspannung und genügend Leistung zur Verfügung stellte? Die Recherchen führten uns zum LME49811 von Texas Instruments, der im Datenblatt vielversprechend mit *Audio Power Amplifier Series High Fidelity 200 Volt Power Amplifier Input Stage with Shutdown* übertitelt war. Die propagierten Eigenschaften konnten uns überzeugen, es war lediglich unklar, ob sich die angegebenen Messwerte allein auf den Baustein oder den vollständigen Endverstärker bezogen. Jedenfalls war es uns die Mühe wert, mit dem LME49811 einen Audio-Endverstärker aufzubauen.

Leistungstransistoren

Die nächste Entscheidung galt den externen Leistungstransistoren (T4/T5), ohne die ein Endverstärker dieser Dimension nicht auskommt. Zu den geforderten Eigenschaften beim Einsatz in Audio-Leistungsverstärkern gehört eine möglichst weite *Safe Operating Area* (SOA). Bei Semelab, einem britischen Halbleiter-Spezialisten, fanden wir den NPN-Typ MG6330-R sowie den dazu komplementären PNP-Typ MG9410-R. Mit der Kollektor-Emitter-Spannung 200 V können über diese Transistoren Kollektorströme von 600 mA und mehr fließen. Solche extremen Situationen sind beispielsweise möglich, wenn der Verstärker ohne Last voll ausgesteuert wird. Die genannten Leistungstransistoren erlauben, dass die Endstufe in Klasse AB mit ausgeprägtem Klasse-A-Verhalten betrieben werden kann. Die Gleichstromverstärkung verhält sich bei Strömen bis zu einigen Ampere nahezu linear, wobei der PNP-Typ geringfügig nachhinkt. Die Typen MG6330-R und MG9410-R von Semelab erfüllen die Anforderungen, die unser linearer Endverstärker stellt.

Die Treiber-Transistoren (T2/T3) müssen vergleichbare Anforderungen erfüllen. Die Typen MJE15032 (NPN) und MJE15033 (PNP) vertragen Spannungen bis 250 V, auch hier verläuft die Gleichstromverstärkung praktisch linear. Allen genannten Typen sind hohe Grenzfrequenzen eigen: Bei den MJE-Typen beträgt sie 30 MHz, für den MG6330-R gibt der Hersteller 60 MHz an, während die Grenzfrequenz des MG9410-R bei 35 MHz liegt. Der Ruhestrom der Endstufe wird mit einem BD139 eingestellt.

Audio-Endverstärker

Schon bald wurde in unserer Redaktion die Frage laut, ob der als Testsignalgeber konzipierte Endverstärker nicht auch als Audio-Endverstärker für den allgemeinen Einsatz nutzbar sein könnte. Die Idee lag nahe, zumal am originalen Entwurf nur wenige Änderungen vorgenommen werden mussten. Die Betriebsspannung wurde auf ±56 V herabgesetzt (Netztrafo 2 · 40 V~), und einige Bauelemente waren neu zu dimensionieren. Die Schaltung in **Bild 1** zeigt das Ergebnis. Bei der niedrigen Betriebsspannung ±56 V ist der Endverstärker in der Lage, höhere Leistungen mit nur einem Paar komplementärer Endtransistoren abzugeben (mehr als 300 W Musikleistung an 4 Ω). Außer dem LME49811 (IC1), der mit vier Transistoren bestückten Leistungsstufe (T2...T5) und der Ruhestromeinstellung mit nur einem Transistor (T1) sind noch einige weitere Komponenten nötig, um den Endverstärker zu komplettieren.

Die Gegenkopplung mit R4/R3 ist so dimensioniert, dass die Eingangsempfindlichkeit für die maximale Aussteuerung ±55 V bei der Betriebsspannung ±60 V etwa 1 V_{eff} beträgt. Dies ist eine Signalspannung, die jeder moderne Vorverstärker zur Verfügung stellt. Die Werte der Widerstände halten die Verlustleistung an R4 bei maximaler Aussteuerung unter 0,25 W. Um die Gleichtaktunterdrückung am Eingang des LME49811 zu optimieren, stimmt die Dimensionierung von R1 und R2 mit R3 und R4 überein. Mit den angegebenen Werten liegt die Eingangsimpedanz bei 15 kΩ.

Die Bandbreite des Eingangssignals hängt an der unteren Grenze von Koppelkondensator C1 ab, die Eckfrequenz liegt hier rechnerisch bei 2,2 Hz. An der oberen Bereichsgrenze bestimmt Kondensator C2 die Eckfrequenz. C1 und C2 bewirken auch eine Begrenzung der Slew-Rate, so dass Problemen des Endverstärkers bei extrem schnellen Eingangssignalen vorgebeugt wird. Außerdem dämpfen diese Maßnahmen eventuelles hochfrequentes Rauschen. Für die Frequenzkompensation des LME49811 ist Kondensator C3 verantwortlich. Dies ist ein Trimmer mit dem Dielektrikum PTFE, ein hervorragendes Material, wenn es um Kondensatoren für Audio-Anwendungen geht. Im Testbetrieb waren die Messwerte optimal, wenn C3 auf etwa ein Drittel der maximalen Kapazität (ungefähr 18 pF) stand. Auf der Platine sind auch versilberte Glimmer-Kondensatoren mit dem Rastermaß 5,9 mm montierbar.

Die mit IC2 aufgebaute DC-Regelung sorgt für die Gleichspannungseinstellung des Endverstärkers. Die Regelung vergleicht die Ausgangsspannung mit der Massereferenz und korrigiert das am nicht invertierenden Eingang des LME49811 (Pin 4) liegende Signal, indem sie einen sehr kleinen Strom hinzufügt. Die Korrektur wird am nicht invertierenden Eingang vorgenommen, weil hier die Impedanz hoch ist. Die Impedanz des invertierenden Eingangs hängt hauptsächlich von R3 ab, sein Wert beträgt 390 Ω. Die Ausregelzeit liegt in der Größenordnung einiger zehntel Sekunden. Der Regelverstärker ist ein OPA177, der Bias-Strom 2,8 nA und die Offsetspannung 60 µV kennzeichnen die hervorragenden DC-Eigenschaften. Die maximale Offsetspannung am Verstärkerausgang beträgt deshalb rechnerisch nur 0,6 mV. Mit angeschlossener Lautsprecherbox ist diese Spannung vernachlässigbar klein. An unserem Musteraufbau haben wir eine Offsetspannung von nur 0,2 mV gemessen.

Die Betriebsspannung ±15 V des Opamp, der für die DC-Korrektur zuständig ist, wird aus der Hauptbetriebsspannung mit Widerständen und Zenerdioden abgeleitet (R17, R18, D1 und D2). Wenn der Endverstärker an einer abweichenden Betriebsspannung arbeiten soll, müssen R17 und R18 angepasst werden. Dabei ist der Strom 1,5 mA zu berücksichtigen, den IC1 über Anschluss 2 dem +15-V-Zweig entnimmt.

Die am Ausgang liegende RC-Kombination R13/C5 trägt dazu bei, dass der Endverstärker auch ohne Last oder mit induktiver Last stabil arbeitet. Induktivität L1 sorgt für zusätzlichen Schutz beim Anschluss kapazitiver Lasten, und Widerstand R12 wirkt eventueller Schwingneigung entgegen. R12 wird durch die Windungen von L1 gesteckt, was Platz spart.

Die Elkos C8 und C9 (4700 µF) sind auf der Platine untergebracht, sie liegen im positiven und negativen Zweig der Betriebsspannung. Die ESR-Werte dieser Elkos müssen möglichst niedrig sein. Der Netztrafo, der Brückengleichrichter und die vier Netzteil-Elkos 10 000 µF/100 V befinden sich nicht auf der Platine, sie werden an einem frei wählbaren Ort in der Nähe montiert.

Der Netztrafo stellt auf der Sekundärseite die

Leitungen

Weil in Audio-Endstufen hohe Spitzenströme über die Betriebsspannungsleitungen fließen, werden die externen Netzteil-Elkos von zwei Elkos mit niedrigem ESR auf der Platine nahe den Endstufentransistoren unterstützt.

In Audio-Endverstärkern sollen Betriebsspannungsleitungen auf der Platine und außerhalb möglichst keine magnetischen Störfelder verursachen. Solche Felder können zu den Verzerrungen beitragen, indem sie Ströme in den Rückkoppelkreis und andere Schaltungsteile induzieren. Eine Methode, diesem Effekt entgegen zu wirken, ist das Führen der drei Leitungen möglichst nahe parallel zueinander sowie die Entkopplung der Spannungen unmittelbar an den Endtransistoren. Wegen der Verstärkerbetriebsart AB fließen auf der Platine einseitig gleichgerichtete Ströme. Wenn die Kupferbahnen des positiven und negativen Zweigs in unmittelbarer Nähe verlaufen, ist das resultierende magnetische Feld fast sinusförmig, was nur äußerst geringe Verzerrungen verursacht. Das Layout einer doppelseitigen Platine lässt sich so gestalten, dass die Leiterbahnen auf beide Seiten verteilt sind und exakt übereinander liegen. Beim Entwurf eines Audio-Endverstärkers mit extrem niedrigen Verzerrungen darf der genannte Effekt nicht vernachlässigt werden.

Eine weitere, konstruktiv wichtige Maßnahme ist die sternförmige Masse, die sich auf der Platine in der Nähe von C5 befindet. An diesem Punkt kommen die Masseleitungen des Eingangs, der Gegenkopplung, der RC-Kombination R13/C5, des Lautsprecheranschlusses und der Stromversorgung zusammen. Die Platine ist für einen einkanaligen Audio-Endverstärker ausgelegt. Für Stereo-Betrieb sind zwei Endverstärker nötig, sie können zusammen mit der (möglichst getrennten) Stromversorgung in einem gemeinsamen Gehäuse untergebracht werden.

Wechselspannung 2 · 40 V~ bereit. Für den Musteraufbau des Endverstärkers kam ein preisgünstiger 500-VA-Trafo zum Einsatz, dessen Ausgangsspannung bei hoher Belastung relativ stark einbrach. Mit einem Netztrafo, der sich stabiler verhält, sind Ausgangsleistungen höher als angegeben erreichbar.

Schutzmaßnahmen

Zwar ist zu hoffen, dass der Endverstärker stets problemlos seinen Zweck erfüllt. Doch leider lehrt die Erfahrung, dass jedes elektronische System (auch eine Audio-Endstufe...) ausfallen kann. Bei fortdauernder Vollaussteuerung und unzureichender Kühlung können sich die Endtransistoren stark aufheizen, was die Lebensdauer dieser Halbleiter verkürzt. Defekte Transistoren verhalten sich meistens wie Kurzschlüsse zwischen den Anschlussleitungen. Wenn die Ströme nicht von Sicherungen abgeschaltet werden, gelangen hohe Gleichspannungen zum Verstärkerausgang, sie können die angeschlossene Lautsprecherbox irreparabel beschädigen. Deshalb gehört eine DC-Sicherung grundsätzlich in jeden Audio-Endverstärker.

Nach dem Einschalten braucht der Endverstärker einige Sekunden, bis sich die Gleichspannungseinstellung stabilisiert hat. Wie auch bei anderen Endverstärkern üblich wird die Lautsprecherbox von einem Relais angeschaltet. Die Relaiskontakte dürfen erst schließen, wenn die Betriebsspannung ihre volle Höhe erreicht hat und der Ausgang nahezu gleichspannungsfrei ist. Bei unserem Endverstärker wird nur der positive Zweig der Betriebsspannung überwacht, denn sie ist die Betriebsspannung der Schutzschaltung (T6...T10). Wenn die Betriebsspannung ausfällt, kann durch die Relaiswicklung kein Strom fließen.

Die DC-Sicherung besteht aus den Transistoren T7 und T8 sowie der RC-Kombination R23/C15, die Zeitkonstante beträgt 3,3 s. Das erscheint zwar recht lang, doch mit steigender Gleichspannung am Ausgang verkürzt sich die Zeit, bis T7 oder T8 den Kondensator C16 entladen. Sobald eine positive Offset-Spannung am Ausgang den Wert 0,55 V übersteigt, leitet T8, so dass das Relais über T9 und T10 abgeschaltet wird. Bei einer negativen Offset-Spannung am Ausgang unter -0,85 V wird T7 aktiv.

Ferner werden die Trafospannungen überwacht, so dass das Relais nach Unterbrechen der Netzspannung oder Ausfall einer Sicherung abschaltet. Damit keine Masseschleife entsteht, sorgt Optokoppler IC3 für die galvanische Trennung zwischen Trafo- und Schutzschaltung. Spannungsteiler R19/R20 ist so bemessen, dass die LED des Optokopplers bereits bei Ausfall einer Trafo-Halbspannung verlischt. C16, R25 und R26 bestimmen die Zeit, die vergeht, bis das Relais die Lautsprecherbox nach Einschalten der Netzspannung anschaltet (etwa 6 s).

Das Relais ist ein Typ, der mit 48 V gesteuert wird, seine Wicklung liegt über 1 kΩ (R29) an der positiven Betriebsspannung +56 V. Falls ein 48-V-Relais nicht beschaffbar ist, kann ein 24-V-Typ als Ersatz dienen. Dann muss R29 den Wert

Kühlung

Nicht nur für die Treiber- und Endstufentransistoren, auch für den LME49811 (IC1) müssen Maßnahmen zur Kühlung getroffen werden. Von IC1 wird die Wärme über eine 2,5 · 8 cm große und 2 mm starke Kühlplatte aus Aluminium abgeleitet. Bei der Betriebsspannung ±56 V werden hier etwa 2 W in Wärme umgesetzt.

Der Kühlkörper der Endtransistoren stellt einen Kompromiss zwischen der geschätzten mittleren Verstärkeraussteuerung und den Abmessungen dar. Permanente Vollaussteuerung würde einen überdimensionalen Kühlkörper oder eine Kühlung mit einem leistungsstarken Lüfter erfordern. Der empfohlene Kühlkörper von Fischer erlaubt Spitzenbelastungen des Endverstärkers während kurzer Perioden bis hin zu einigen Minuten. Mit 10 cm Höhe ist dieser Kühlkörper noch immer vergleichsweise stattlich. Der thermische Widerstand beträgt 0,7 W/K, wegen der potentiellen Gefahr der Überhitzung darf er keinesfalls höher sein.

Im Detail betrachtet: Mit der stabilisierten Betriebsspannung ±56,8 V liefert der Audio-Endverstärker an 4 Ω fast 300 W, wobei die Verzerrungen 0,1 % betragen. Wenn der Wirkungsgrad 68,5 % beträgt, müssen 137 W als Wärme abgeführt werden. Unter der Annahme, dass das Signal sinusförmig ist, würde die Temperatur der Endtransistoren im Dauerbetrieb die Umgebungstemperatur um mehr als 90 °C übersteigen. Auch die Emitterwiderstände R10 und R11 (5-W-Typen) wären an der Grenze ihrer Belastbarkeit angekommen. Eine solche extreme Situation kann bei der Wiedergabe von Musiksignalen nicht eintreten. Ergänzend sei hinzugefügt, dass die Hersteller von Audioverstärkern die Kühlung ihrer Endstufen niemals für Dauerspitzenbelastungen mit sinusförmigen Signalen auslegen.

Stückliste

Widerstände:

(0,25 W/5 %, wenn nicht anders angegeben)
R1,R3 = 390 Ω
R2,R4,R17,R18,R22,R23,R30 = 15 k
R5 = 8k2
R6,R20,R28 = 1k2
R7 = 220 Ω
R8,R9 = 100 Ω
R10,R11 = 0Ω2, 1 %, 5 W, induktionsarm (Vishay Dale LVR05R2000FE73)
R12,R13 = 3Ω9, 5 %, 5 W
R14 = 220 k
R15,R16 = 10 M
R19 = 27 k
R21 = 470 k
R24 = 1 M
R25,R26 = 820 k
R27 = 68 k
R29 = 1 k
P1 = 470 Ω Trimmpoti, liegend

Kondensatoren:

C1 = 4µ7/63 V, MKT, Raster 5/7,5 mm
C2 = 1 n/400 V, MKT, Raster 5/7,5 mm
C3 = Trimmer 5...57 p/250 V, liegend (Vishay BCcomponents BFC280908003)
C4,C6,C7 = 100 n/100 V, Raster 5/7,5 mm
C5 = 47 n/400 V, Raster 5/7,5 mm
C8,C9 = 4700 µ/100 V, Raster 10 mm, Snap-in, ø 30 mm (Panasonic ECOS2AP472DA)
C10 = 2µ2/63 V, Raster 5/7,5 mm
C11 = 33 n/63 V, Raster 5/7,5 mm
C12,C13,C16 = 10 µ/100 V, Raster 2,5 mm, ø 6,3 mm
C14 = 1 µ/250 V, Raster 2,5 mm, ø 6,3 mm
C15 = 220 µ/16 V bipolar, Raster 5 mm, ø 10 mm

Induktivitäten:

L1 = 450 nH, 13 Windungen lackierter Kupferdraht ø 1,5 mm, Wicklung innen ø 7 mm

Halbleiter:

D1,D2 = Zenerdiode 15 V/0,5 W
D3,D4 = 1N4004
D5 = 1N4148
D6 = LED 3 mm, rot
T1 = BD139
T2 = MJE15032
T3 = MJE15033
T4 = MG6330-R
T5 = MG9410-R
T6...T10 = 2N5550
IC1 = LME49811TB/NOPB
IC2 = OPA177GPG4
IC3 = 4N25

Außerdem:

K1 = Stiftkontaktleiste 2-polig, Raster 2,54 mm
K2...K6 = 6,3-mm-Faston-Stecker für Platinenmontage, Raster 5,08 mm
K7 = Schraubklemmverbinder 3-polig, Raster 5 mm
RE1 = Relais für Platinenmontage, SPCO, 16 A, Wicklung 48 V/5,52 kΩ (TE Connectivity/Schrack RT314048)
Isolierscheiben TO-220 für T1...T3, Kapton MT-Film, 0,15 mm, 6 kV
Isolierscheiben TO-3P für T4 und T5, Kapton MT-Film, 0,15 mm, 6 kV
Isolierringe TO-220, 3 mm, für T2 und T3
Kühlkörper 0,7 K/W (z. B. Fischer SK 47/100 SA)
Kühlplatte für IC1, 2,5 · 8 cm, Aluminium, 2 mm stark
Platine 110656-1, siehe www.elektor.de/110656

Stromversorgung (für *eine* Endstufe):

Netztrafo 2 · 40 V~/ 500 VA (z. B. Nuvotem 0500P1-2-040)
Brückengleichrichter 200 V/35 A (z. B. Fairchild GBPC3502)
4 Elkos 10 000 µF/100 V (2 Elkos parallel in beiden Spannungszweigen)

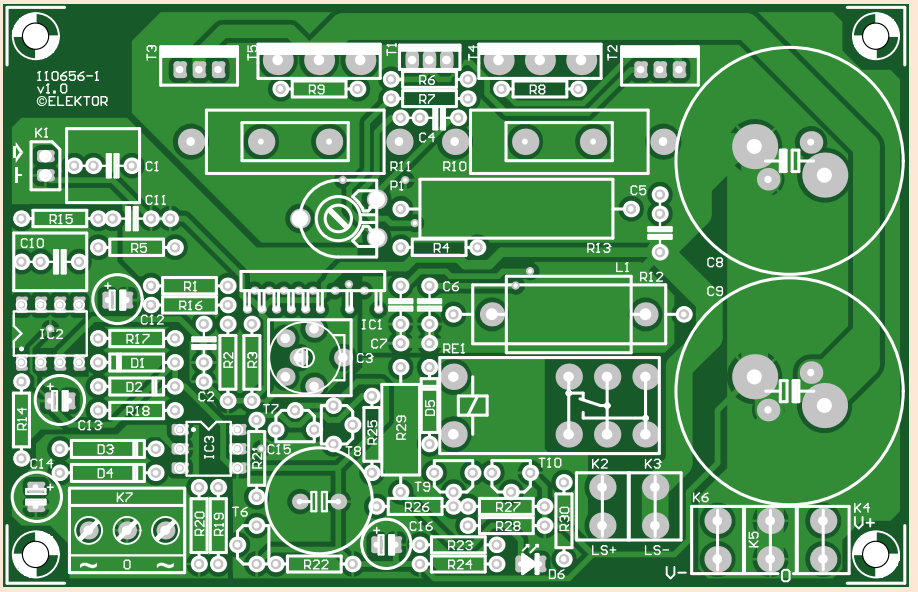


Bild 2.
Auf der Platine hat der Endverstärker einschließlich Schutzschaltung seinen Platz. Stromversorgung und Leistungstransistoren werden separat montiert.

2,2 kΩ/1 W erhalten.
Die Schutzschaltung ist für die Betriebsspannung ±56 V dimensioniert. Beim Betrieb an einer niedrigeren Spannung müssen mehrere Widerstände angepasst werden. Dies gilt auch für die Widerstände im Gegenkoppelzweig, wenn die Eingangsempfindlichkeit 1 V_{eff} beibehalten werden soll. Außerdem ist zu berücksichtigen, dass die Verstärkung des LME49811 mindestens 20 (= 26 dB) betragen muss.

Konstruktion

Das Layout der für den Audio-Endverstärker entworfenen Platine ist in **Bild 2** wiedergegeben. Der Aufbau ist nicht schwierig, trotzdem sind einige Besonderheiten zu beachten. Die Transistoren T1...T5 sowie die Komponenten der Stromversorgung (Netztrafo, Gleichrichter, Elkos C8 und C9) befinden sich nicht auf der Platine. Für den Anschluss der Lautsprecherbox und der symmetrischen Betriebsspannung sind Flachstecker vorgesehen (Faston 6,3 · 0,8 mm), sie werden unmittelbar auf die Platine gelötet.
Induktivität L1 besteht aus 13 Windungen lackierter Kupferdraht, Drahtdurchmesser 1,5 mm. Als Wickellehre dient der Schaft eines 7-mm-Spiralbohrers. Die Anschlüsse sollen so lang sein, dass L1 von der Platine etwas abstehen kann, sie müssen in der Mitte nach unten verlaufen. Widerstand R12 wird in die Wicklung gesteckt, seine Anschlussdrähte werden so gebogen, dass sie in die zugehörigen Platinenbohrungen passen. Es ist zweckmäßig, beide Bauelemente gleichzeitig zu montieren. L1 soll über der Platine „schweben“, und R12 soll seinen Platz im Zentrum von L1 haben (siehe **Bild 3**).
Vor dem nächsten Schritt ist zu klären, in welches Gehäuse der Endverstärker eingebaut werden soll. Das Gehäuse entscheidet über die Anordnung der Platine und des Kühlkörpers. Eine zweckmäßige Lösung ist die Montage der Platine mit Bügeln auf dem Kühlkörper. Die Platine bleibt zugänglich, auch wenn die Transistoren bereits auf dem Kühlkörper montiert sind.
Montieren Sie die Platine so gegen den Kühlkörper, dass die Anschlüsse der Transistoren nahe den zugehörigen Platinenanschlüssen liegen. Mit einer geeigneten Zange biegen Sie in die Anschlüsse von T1...T5 flache, halbkreisförmige Bögen. Die Anschlüsse sollen etwas nach vorn stehen, so dass sie ohne mechanische Spannung in die zugehörigen Platinenbohrungen passen. Der erste Bogen muss möglichst nahe dem Gehäuse

Bild 3.
Nahaufnahme der Induktivität L1, darin befindet sich der Leistungswiderstand R12.

liegen. Setzen Sie die Anschlüsse niemals einer Biegekraft aus, ohne mit einem flachen Werkzeug gegenzuhalten. Anderenfalls können Mikrorisse in der Anschlussdurchführung entstehen, sie können die Funktion des Transistors nachhaltig beeinträchtigen. Der zweite Bogen muss über den Platinenbohrungen liegen. Aus **Bild 4** ist ersichtlich, wie dies im Detail aussieht.
Sie können die Isolierscheiben der Transistoren provisorisch zwischen Transistor und Kühlkörper schieben, bevor Sie die genaue Position des zweiten Bogens festlegen. Dies ist jedoch nur wichtig, wenn Sie keramische Isolierscheiben verwenden. Erst wenn Sie die Transistoren auf den Isolierscheiben fest mit dem Kühlkörper verschraubt haben, verlöten Sie die Anschlüsse mit der Platine.

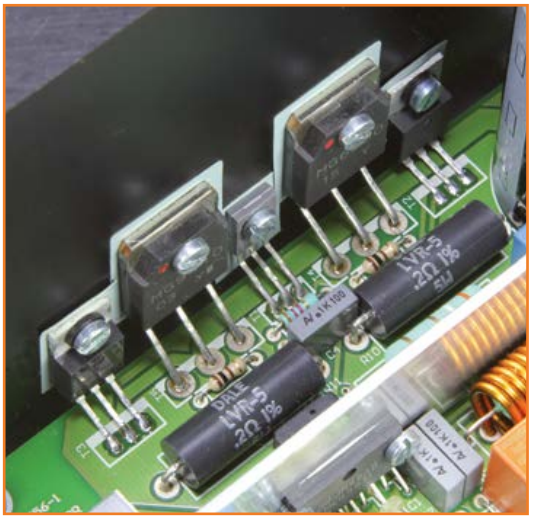


Bild 4.
Die Anschlüsse der Transistoren auf dem Kühlkörper sind bogenförmig, damit sie mechanisch spannungsfrei in die zugehörigen Platinenbohrungen gesteckt werden können.

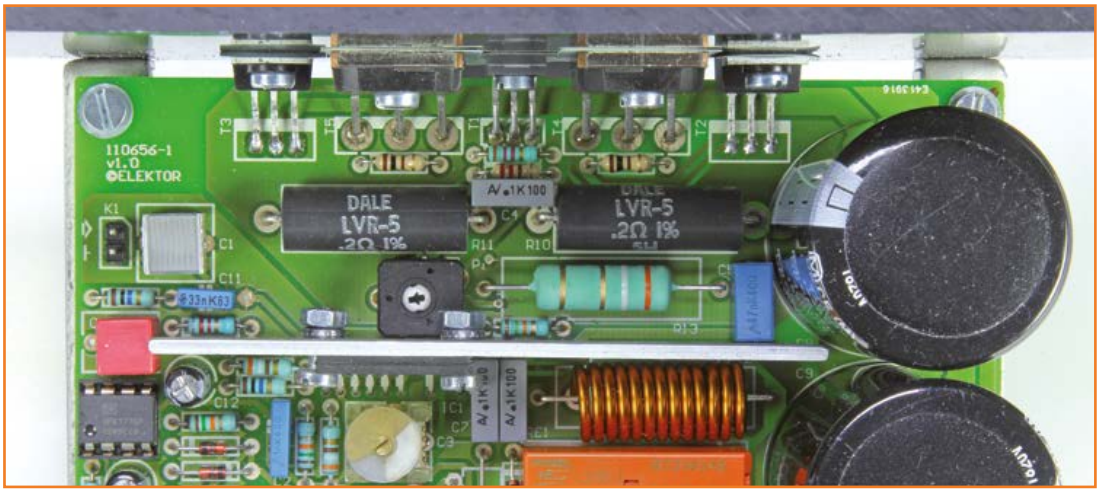


Bild 5.
Auf der Platine wurde Raum reserviert, in den das Treiber-IC mit der Kühlplatte hineinpasst.

Das nächste zu montierende Bauelement ist IC1. Zuerst schrauben Sie das IC gegen eine 2,5 · 8 cm große und 2 mm starke Kühlplatte aus Aluminium. Bringen Sie die Platte so an, dass sie nach der Montage von IC1 nicht auf der Platine aufliegt, sie darf auch R1, R4 oder R5 nicht berühren. Wichtig ist hier, dass die metallene Rückseite von IC1 mit der negativen Betriebsspannung verbunden ist! Aus Gründen der Sicherheit empfehlen wir, IC1 und die Kühlplatte in jedem Fall voneinander zu isolieren. Wenn IC1 einbaufertig ist, montieren Sie das IC einschließlich Kühlplatte auf der Platine. Der dort reservierte Raum ist für diese Konstruktion genau passend bemessen (siehe **Bild 5**). Biegen Sie L1 von der Kühlplatte weg.

Wenn Sie erst jetzt die etwas voluminösen Netzteil-Elkos C8 und C9 in Position bringen und anschließen, sind sie bei den vorangegangenen Arbeiten nicht im Weg.

Probelauf

Bevor Sie den Endverstärker unmittelbar mit der Betriebsspannung verbinden können, müssen Sie den Ruhestrom der Endstufe einstellen. Dazu brauchen Sie zwei Leistungswiderstände 47 Ω/5 W, die Sie in die Leitungen der positiven und negativen Betriebsspannungszweige legen. Falls ein gravierender Fehler auftritt, beispielsweise ein Kurzschluss, wird der Endverstärker nicht beschädigt. Im schlimmsten Fall werden lediglich die Leistungswiderstände defekt. Alternativ können Sie den Endverstärker an ein stabilisiertes ±56-V-Netzteil mit Strombegrenzung anschließen (falls Sie ein solches Netzteil besitzen). Legen Sie ein als Strommesser geschaltetes Multimeter in den positiven Spannungszweig. Vor dem ersten Einschalten der über die Widerstände geleiteten Betriebsspannung drehen Sie Trimmpoti P1 gegen den Uhrzeigersinn bis zum linken Anschlag. Vergessen Sie nicht, die Trafo-Anschlüsse (2 · 40 V~) mit der Kontaktklemme K7 zu verbinden. Nach dem Einschalten müssen im positiven Zweig bei aktiviertem Lautsprecher-Relais ungefähr 30 mA fließen. Drehen Sie P1 vorsichtig im Uhrzeigersinn, bis der Strom im positiven Zweig 60 mA beträgt. Dieser niedrige Wert reicht für den AB-Betrieb der Endstufe aus. Wenn während des Betriebs die Kühlkörpertemperatur steigt, nimmt auch der Ruhestrom etwas höhere Werte an, normalerweise bleibt er jedoch unter 90 mA. Bei sehr hohen Ausgangsleistungen steigt die Sperrschichttemperatur der Endtransistoren schneller als die Temperatur des Kühlkörpers. Der für den Ruhestrom verantwortliche Transistor kann diesen Effekt nicht vollständig kompensieren. Die Folge ist ein Ruhestrom, der kurzzeitig bis auf einige hundert Milliampere hochlaufen kann. Sobald die Temperatur sinkt, geht auch der Ruhestrom auf Normalwerte zurück. Unsere Audio-Endstufe zeigt auch hier ihre Stärke: Wenn die Ausgangsleistung hohe Werte erreicht, verschiebt sich der Betrieb als Klasse-AB-Verstärker in Richtung Klasse-A-Verstärker. Wir wünschen Ihnen mit unserem neuen Endverstärker viele klangvolle Hörerlebnisse!

(110656)gd

Weitere Informationen zum Audio-Endverstärker enthält die englischsprachige Projektseite:

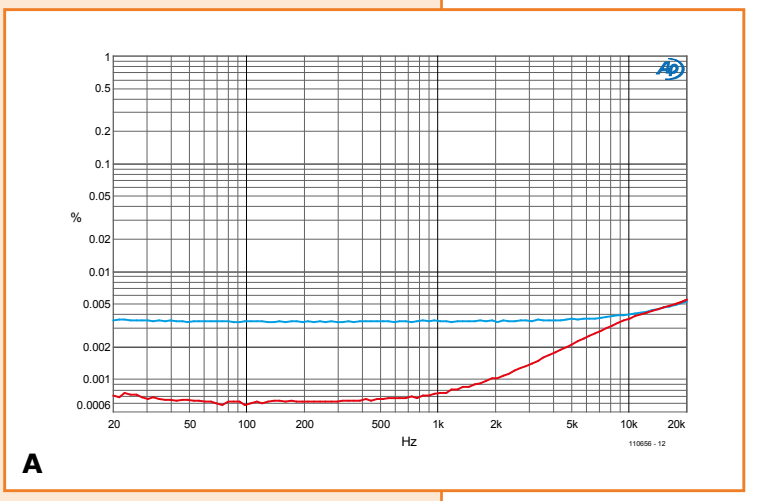
www.elektor-labs.com/node/3247

Messkurven

Messgerät: Precision System Two Cascade Plus 2722 Dual Domain

Kurve A

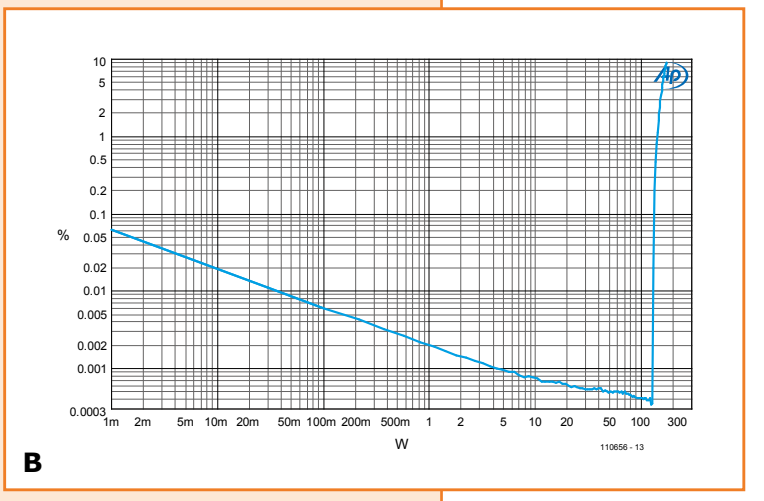
THD+N bei den Ausgangsleistungen 1 W/8 Ω und 50 W/8 Ω, B = 80 kHz. Die zur Leistung 1 W gehörende Kurve zeigt hauptsächlich Rauschen (THD+N = 0,0034%). Erst kurz vor der Frequenz 20 kHz übersteigen die Verzerrungen geringfügig das Rauschen (THD+N = 0,0052%). Bei 50 W (das sind 20 V, so dass die Messkurven mit den Angaben im Datenblatt des LME49811 leicht verglichen werden können) liegt der Rauschteppich im Verhältnis zur Ausgangsspannung wesentlich niedriger. Hier ist erkennbar, dass die Verzerrungen bei höheren Frequenzen früher ansteigen. Bei 1 W liegen die Verzerrungen immer unter dem Rauschen. Die Verzerrungen oberhalb 10 kHz sind mit der 1-W-Kurve fast identisch. Die Kurve für die Leistung 100 W ist nicht dargestellt, da sie mit der Kurve für 50 W praktisch übereinstimmt. Bis kurz vor Einsetzen der Begrenzung sind die Verzerrungen bei allen Leistungen ähnlich niedrig.



A

Kurve B

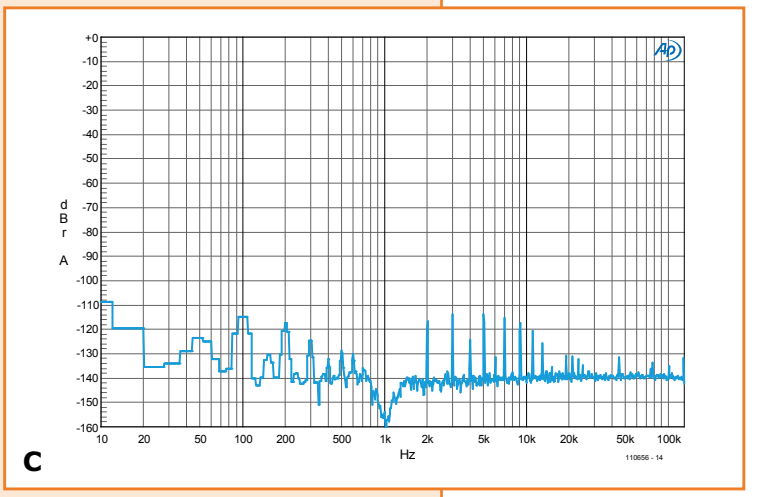
THD+N als Funktion der Ausgangsleistung (1 kHz/8 Ω, B = 22 kHz). Die Messbandbreite ist reduziert, damit der Anstieg der Verzerrungen optimal sichtbar wird. Auch hier ist ersichtlich, dass die Verzerrungen extrem niedrig bleiben, während der Rauschteppich mit zunehmender Ausgangsspannung sinkt. Bei der Leistung 127 W setzt die Begrenzung ein, darüber steigen die Verzerrungen schnell an. THD+N erreichen bei 137 W den Wert 0,1 %, dies ist für hochwertige Musikwiedergabe immer noch akzeptabel. Wenn der Verstärker übersteuert wird, beträgt die Leistung bei 10 % THD+N sogar 174 W. Dazu ist anzumerken, dass mit dem verwendeten preisgünstigen Netztrafo die Betriebsspannung bei Vollaussteuerung merklich einbrach (±51,5 V bei 10 % THD+N). Mit stabiler arbeitenden Netztrafos sind noch etwas höhere Leistungen erreichbar.



B

Kurve C

FFT für 1 kHz bei 50 W/8 Ω (20 V_{eff}). Die Restwelligkeit der Betriebsspannung und die Harmonischen von 1 kHz haben extrem niedrige Werte, sie sind für das menschliche Ohr nicht hörbar. Die dritte Harmonische liegt bei -113,8 dB, was 0,0002 % entspricht! Bei dieser Leistung haben THD+N den Wert 0,0006 % (B = 80 kHz).



C

USB-Thermometer

Externe Geräte einfach via USB auslesen

Von
Dipl.-Ing.
Michael Odenwald
(D)

Früher war die serielle RS232-Schnittstelle für Elektroniker eine Art McGyver-Universalanschluss. Das hat sich in den letzten Jahren zunehmend geändert, denn bei modernen PCs sind die neunpoligen Sub-D-Stecker seltener geworden. Wenn man heute eigene Elektronik mit einem PC verbinden will, kommt USB zum Zuge. Doch die PC-Seite verlangt nach einem passenden Treiber. Dieses Projekt zeigt, wie man das Problem elegant löst.

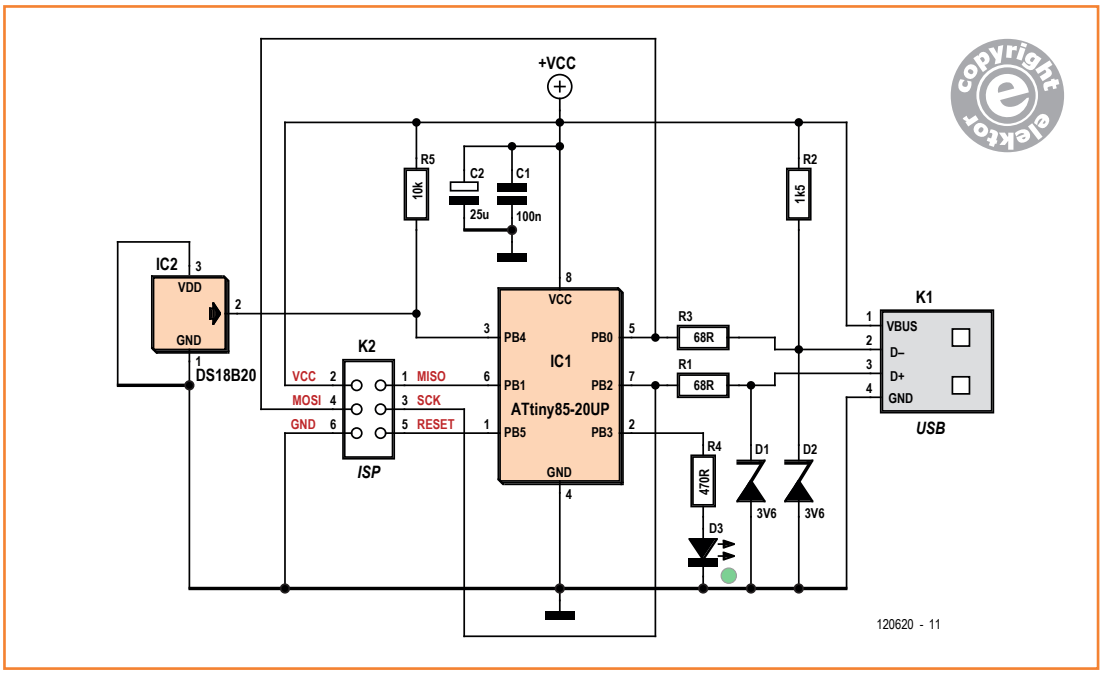


Bild 1.
Die einfache Schaltung des USB-Thermometers.

Die Entwicklung von Systemtreibern – und das auch noch für unterschiedliche Betriebssysteme – ist alles andere als trivial. Hinzu kommen Feinheiten wie digitale Signaturen, was die Kette USB-Gerät/Treiber/Betriebssystem/Anwenderprogramm zusätzlich kompliziert. Dieser Aufwand ist gerade für kleine Projekte sicherlich zu groß. Aus diesem Grund wird gerne ein virtueller COM-Port verwendet, der aber (Konfigurations-) Nachteile mit sich bringt und zudem viele tolle Möglichkeiten des USB nicht nutzt.
Ein USB-Modus, der immer und überall funktioniert, der Prototypen wie Kleinserien unterstützt und dennoch viel Komfort bietet, ist die Verwendung der sogenannten USB-HID-Geräteklasse (**H**uman **I**nterface **D**evice). USB-HID umfasst aber nicht nur Maus und Tastatur (siehe **Kasten**). Der USB-Standard [1] sieht explizit auch „sonstige Geräte" vor. Dies können Aktoren und Sensoren jeder Art sein.
Da jedes aktuelle Betriebssystem schon USB-HID-Treiber mitbringt, kann man diese prima für eigene Zwecke mitnutzen. Man muss dann „nur" noch das passende Anwenderprogramm zum jeweiligen Betriebssystem erstellen.

Daten erfassen

Per USB-HID kann man sehr gut externe Größen per PC erfassen. Die Temperaturmessung ist ein gutes Beispiel. Als intelligente Basis zur Generie-

rung solcher Daten reicht schon ein kleiner Mikrocontroller der ATtiny-Klasse. In der Schaltung von **Bild 1** ist dies ein ATtiny85-20 (IC1). Er dient zur korrekten Behandlung des USB-Protokoll-Stacks bzw. zur Kommunikation sowie der Abfrage und Aufbereitung von Sensordaten.
IC1 wird wie der Temperatursensor IC2 direkt von den 5 V aus der USB-Buchse K1 versorgt. Der Mikrocontroller wird per interner PLL mit 16,5 MHz getaktet, sodass man sich den Quarz sparen kann und das Tempo trotzdem für USB ausreicht.
IC2 ist ein Temperatursensor mit 1-Wire-Schnittstelle des Typs DS18B20 von Dallas. Wenn Sie staunen, warum VDD mit GND verbunden ist: Das ist der „parasite power mode" [2], bei dem IC2 aus der Datenleitung versorgt wird, um die Eigenerwärmung so gering wie möglich zu halten. Die USB-Verbindung wird über die beiden Schutzwiderstände R1 und R3 hergestellt, die bei einem Kurzschluss den Strom der Datenleitungen begrenzen. Die 3,6-V-Z-Dioden D1 und D2 begrenzen den Pegel der Datenleitungen auf USB-gemäße Spannungen.
Für die USB-Enumeration sorgt Widerstand R2, der dem USB-Host (im PC) ein „low speed device" mit einer maximalen Datenrate von 1,5 Mbit/s signalisiert.
Die Kondensatoren C1 und C2 sorgen für die Pufferung und Entkopplung der Versorgungsspannung aus der USB-Buchse. K2 ist der sechspolige ISP-Sockel zur Programmierung von AVR-Controllern. Die LED D3 signalisiert einen aktiven Messzyklus, der bei maximaler Auflösung des Temperatursensors etwa 750 ms dauert.

Firmware

Die Firmware für das USB-Thermometer ist in C geschrieben. Sie lässt sich mit WinAVR [3] übersetzen und als Compilat in den Flash-Speicher des Mikrocontrollers übertragen. Der USB-Stack wird dabei mit Hilfe der V-USB-Software [4] realisiert. Die Funktionen zur Abfrage des Temperatursensors nutzen eine Library von Martin Thomas [5].
Nach der Initialisierung der Hardware und des USB-Software-Stacks wird die USB-Enumeration durchlaufen. Anschließend wechselt die Software in den internen Betriebsmodus, der durch eine „state machine" realisiert wird. Die verschiedenen Zustände dieses „endlichen Automaten" sind: USB-Protokoll, Abfrage des Sensors und Warten. Sie werden nacheinander mit einer voreingestellten Zeit durchlaufen.
Der gesamte Zyklus dauert 10 s. In diesem Intervall ändert sich der vom Host abfragbare Wert nicht. Eine Abfrage bringt also lediglich alle 10 s neue Daten. Die Messung wird vom Mikrocontroller gesteuert und muss nicht vom PC aus initiiert werden. Der Temperatursensor hat daher Zeit, wieder abzukühlen.
Das wichtigste Element der Firmware ist der USB-HID-Descriptor.

USB-HID-Geräteklasse

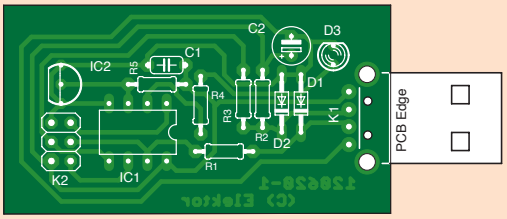
Die HID-Geräteklasse (**H**uman **I**nterface **D**evice) ist eine Teildefinition des USB-Standards, die Geräte zur Benutzereingabe beschreibt. Typische Vertreter sind Tastatur, Maus oder Joystick. Der USB-Standard sieht neben solchen „natürlichen" HID-Eingabegeräten aber auch explizit „sonstige Systeme" vor. Dies können Sensoren, Messgeräte oder auch Telefone und Headsets sein. Möglich sind auch Lesegeräte sowie diverse Fun- und Werbeartikel.
USB-HID-Geräte haben die angenehme Eigenschaft, dass die zugehörigen Systemtreiber schon Bestandteil des Betriebssystems sind - zumindest bei Windows, Linux und OS X - und deshalb automatisch, also ohne Anwendereingriff, einem neu angeschlossenen USB-HID-Gerät zugeordnet werden.
HID-Geräte haben, das sei nicht verschwiegen, auch Nachteile: Die Datenübertragungsrate ist nicht besonders hoch, und es gibt nur wenige USB-Endpoints, über die auch nur wenige Daten übertragen werden können.

Der USB-HID-Descriptor.

```
/*
 * The USB Hid report descriptor
 */
PROGMEM char usbHidReportDescriptor[33] =
{
    0x06, 0x00, 0xff,        // USAGE_PAGE (Generic Desktop)
    0x09, 0x01,              // USAGE (Vendor Usage 1)
    0xa1, 0x01,              // COLLECTION (Application)
    0x15, 0x00,              //   LOGICAL_MINIMUM (0)
    0x26, 0xff, 0x00,        //   LOGICAL_MAXIMUM (255)
    0x75, 0x08,              //   REPORT_SIZE (8)
    0x85, 0x0a,              //   REPORT_ID (10)
    0x95, 0x04,              //   REPORT_COUNT (4)
    0x09, 0x00,              //   USAGE (Undefined)
    0xb2, 0x02, 0x01,        //   FEATURE (Data,Var,Abs,Buf)
    0x85, 0x14,              //   REPORT_ID (20)
    0x95, 0x0a,              //   REPORT_COUNT (10)
    0x09, 0x00,              //   USAGE (Undefined)
    0xb2, 0x02, 0x01,        //   FEATURE (Data,Var,Abs,Buf)
    0xc0                     // END_COLLECTION
};
```

Stückliste

Widerstände: (alle 0,25 W)
R1,R3 = 68 Ω
R2 = 1k5
R4 = 470 Ω
R5 = 10 k



Kondensatoren:
C1 = 100 n, keramisch, RM 5 mm
C2 = 25 µ/16 V, Elko, RM 2,5 mm

Halbleiter:
IC1 = ATtiny85-20UP, 8-pol. DIL (programmiert: 120620-41 [6])
IC2 = DS18B20, 3-pol. TO92-Gehäuse
D1,D2 = ZF 3,6; Zener-Diode 0,5 W
D3 = LED, grün, 5 mm

Außerdem:
K1 = USB-Buchse, Typ A, für Platinenmontage
K2 = 6-pol. Stiftleiste, 2x3, RM 1/10“
Platine 120620-1 [6]

Bild 2.
Der Bestückungsplan der Platine.

Der Descriptor umfasst 33 Byte. Er definiert die möglichen Report IDs (10 und 20), mit denen das Anwendungsprogramm mit dem Thermometer kommuniziert. Die Reports sind dabei als sogenannte „Feature Reports“ mit unterschiedlich langen Informationsblöcken (4 und 10 Byte) realisiert. Ein Feature Report erlaubt es, Werte vom USB-HID-System zu lesen oder zu schreiben, wobei hier nur gelesen wird.

Report ID 10 dient zum Abfragen der Temperatur und gibt vier Byte zurück. Report ID 20 fragt einen Ident-String ab, der aus 10 Bytes im Datumsformat (yyyy-mm-tt) besteht.

Die zentrale Klasse des USB-Thermometers

```
namespace WindowsApp
{
  /// <summary>
  /// Implementation of the usbDevice with service methods
  /// based on the class usbGenericHidCommunication
  /// </summary>
  class usbDevice : usbGenericHidCommunication
  {
      private int tval;

      /// <summary>
      /// Class constructor - place any initialisation here
      /// </summary>
      /// <param name="vid"></param>
      /// <param name="pid"></param>
      public usbDevice(int vid, int pid)
          : base(vid, pid)
      {
      }

      /// <summary>
      /// USB HiD Temperatur Modul Methode GetTemeratur()
      /// </summary>
      public int GetTemperatur()
```

Vom Aufbau zum Treiber

Die Schaltung ist unkritisch und es sind lediglich bedrahtete Bauteile vorgesehen. Von daher ist der Aufbau mit der Platine von **Bild 2** ein Kinderspiel. Die Layout-Dateien der Platine kann man sich von der Elektor-Webseite zu diesem Artikel [6] herunterladen. **Bild 3** zeigt den fertig bestückten Prototypen. Ein Abgleich ist nicht erforderlich.

Nach Bestückung und Überprüfung der Platine wird der Mikrocontroller über K2 und einem passenden AVR-ISP-Programmer mit der Firmware programmiert, die von [6] als Source-Code und direkt programmierbare Hex-Datei kostenlos erhältlich ist. Bei den Fuse-Bits ist vor allem darauf zu achten, dass der Taktteiler „/8" eines jungfräulichen ATtiny deaktiviert und der richtige interne Takt gesetzt wird. Die Einstellungen sind richtig, wenn man für die „low fuse" den Wert 0xE1 und für die „high fuse" den Wert 0xDD eingibt, was mit so ziemlich jeder Programmer-Software möglich sein sollte.

Direkt nach der Programmierung des Mikrocontrollers kann man die Schaltung per USB-Kabel an den PC anschließen. Das Betriebssystem erkennt dann ein neues HID-Device und installiert umgehend die HID-Systemtreiber - so einfach ist das! Es spielt keine Rolle, ob es sich um ein 32- oder 64-bittiges Windows, OS X oder Linux handelt: Die HID-Systemtreiber sind immer vorhanden, immer digital signiert und werden ohne

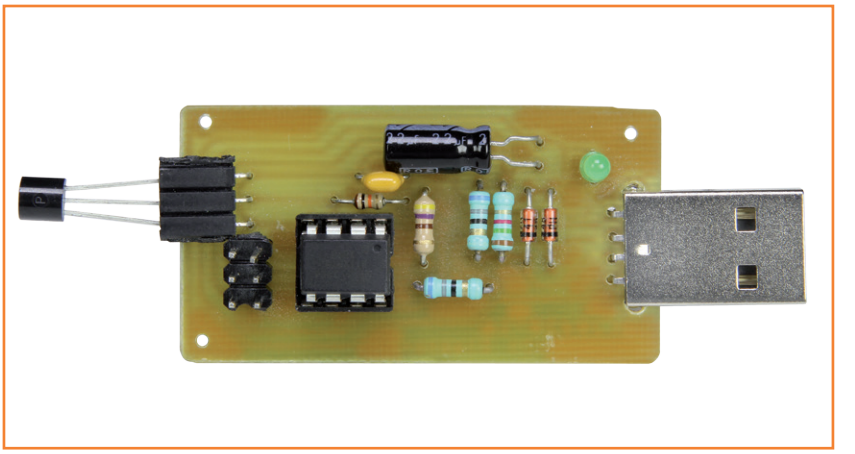


Bild 3.
Prototyp des USB-Thermometers.

```
      {
          // Declare a input buffer
          Byte[] inputBuffer = new Byte[5]; // we expect 5 byte; 1 x ReportID and 4 Byte temperature

          inputBuffer[0] = 10; // Read ReportID 10

          // Perform the Read Command
          bool success;
          success = getFeatureReport(inputBuffer);

          if (success == false)
          {
              Debug.WriteLine("Error during getFeatureReport");
              return tval;  // Error during USB HiD_GetFeature Request so return the old value
          }

          tval  = inputBuffer[1] << 24;
          tval |= inputBuffer[2] << 16;
          tval |= inputBuffer[3] << 8;
          tval |= inputBuffer[4];

          return tval; // Return the new value
        }
    }
}
```

Nachfrage sofort installiert. Spätestens ein paar Sekunden nach dem Anschluss ist die Schaltung einsatzbereit.

Anwendungsprogramm

Das (Windows-)Programm zur Abfrage des USB-Thermometers und Anzeige der Temperatur ist in C# geschrieben und ein gutes Beispiel dafür, wie mit HID-Geräten kommuniziert werden kann. Die Host-Software bedient sich der usbGenericHIDDevice-Funktionen der Library aus [7], welche die API-Funktionen von Windows kapseln. Die Software kann mit der Express-Version von Visual Studio 2010 [8] übersetzt werden. Wichtig ist die Basis-Klasse usbGenericHidCommunication, von der eine eigene Klasse für das

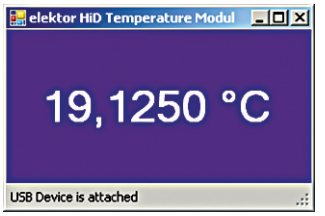


Bild 4.
Temperaturanzeige des Anwendungsprogramms unter Windows.

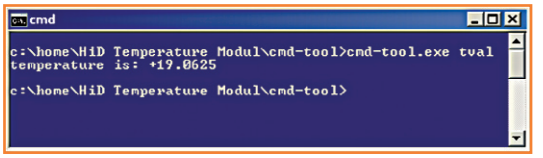


Bild 5.
Temperaturausgabe via Kommandozeilen-Tool.

eigene HID-Gerät abgeleitet werden muss. In dieser Klasse werden dann die Methoden implementiert, die ausgeführt werden sollen. Im Kasten die zentrale Klasse des USB-Thermometers.

Das HID-Gerät wird mit den Parametern Vendor ID = 0x0C7D und der Product ID = 0x0011 identifiziert und aufgerufen. Mit der Methode GetTemperatur() kann man die Temperatur abfragen. Der Messbereich des Sensors beträgt -55...+125 °C. Die Temperatur wird dabei als „signed longint" mit dem Wertebereich -550.000...+1.250.000 übertragen. Das Programm muss diesen Wert dann durch 10.000 teilen. Resultat ist die Temperatur mit einer Auflösung von 12 bit bzw. 0,0625 °C. Das sollte man aber nicht überbewerten, denn die Grundgenauigkeit des Sensors liegt bei lediglich 0,5 °C. In **Bild 4** ist die Temperaturanzeige des Programms unter Windows zu sehen.

Das Windows-Programm ist als Source-Code samt nötiger Library ebenfalls via [6] downloadbar. Neben diesem „richtigen" fenster-basierten Programm kann man die aktuelle Temperatur auch ganz schlicht per Command-Line-Tool abfragen. Ein Aufruf dieses kleinen Programms liefert dann das Ergebnis als Text (siehe **Bild 5**). Der zugehörige Source-Code samt Compilat und zusätzlich auch das Tool für Linux sind ebenfalls unter [6] erhältlich.

Fazit

Schaltung und zugehörige Software des USB-Thermometers demonstrieren ein einfaches, praktisches Konzept zur PC-Anbindung eigener USB-HID-Geräte. Beides kann man leicht für eigene Zwecke abändern, und schon hat man ein (fast) beliebiges Gerät stressfrei an einen PC angebunden, wenn dabei nicht allzu viele Daten in zu hohem Tempo übertragen werden sollen.

(120620)

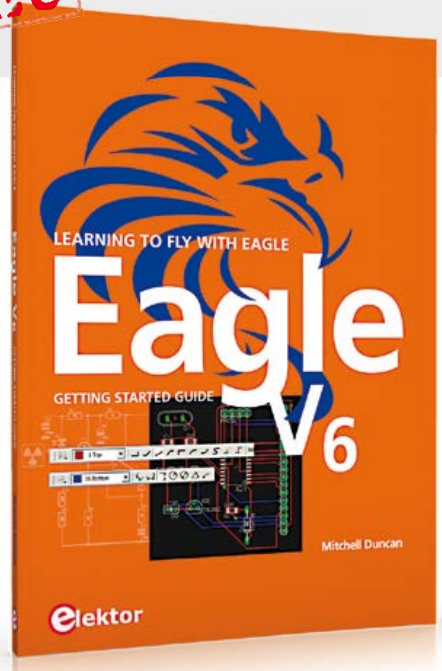
Weblinks

[1] www.usb.org/developers/hidpage/
[2] http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf
[3] http://winavr.sourceforge.net/
[4] www.obdev.at/products/vusb/index.html
[5] www.siwawi.arubi.uni-kl.de/avr_projects/tempsensor/index.html
[6] www.elektor.de/120620
[7] www.waitingforfriday.com/index.php/Open_Source_Framework_for_USB_Generic_HID_devices_based_on_the_PIC18F_and_Windows
[8] www.microsoft.com/germany/express/products/windows.aspx

NEU

EAGLE V6

Neues Fachbuch in englischer Originalsprache



EAGLE ist ein anwenderfreundliches, leistungsfähiges und preiswertes Software-Paket für die Entwicklung von Platinen. Es bietet einen vergleichbaren Leistungsumfang wie andere Pakete zu einem attraktiveren Preis. Für Einsteiger und Hobbyisten steht sogar eine kostenlose Version von EAGLE zur Verfügung.

Ein weiterer Vorteil ist die Plattform-Unabhängigkeit: EAGLE gibt es für die wichtigsten Betriebssysteme wie Microsoft Windows (XP, Vista oder Windows 7); Linux (ab Kernel Version 2.6) und Apple OS X (ab Version 10.6). EAGLE läuft auf jeder Hardware, die von diesen Betriebssystemen unterstützt wird.

Dieses neue (englischsprachige) Buch enthält eine solide Einführung in die Fähigkeiten von EAGLE und eignet sich sowohl für Einsteiger als auch für Profis mit Erfahrung im Platinen-Design, die sich in EAGLE einarbeiten wollen, da sie das CAD-Paket wechseln.

206 Seiten, inkl. Software-CD · Format 17 x 23,5 cm (kart.)
ISBN 978-1-907920-20-2
€ 34,50 · CHF 42,80

elektor

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

BATRONIX

FÜHRENDE PROGRAMMIER- UND MESSLÖSUNGEN

Rigol DS1102E Speicheroszilloskop — 387,94 *
100 MHz, 1 GSa/s, 1 Mpts, Bestseller, FFT, USB, LAN, automatische Messungen, 3 Jahre Garantie

Rigol DS2072 Speicheroszilloskop — 844,90 *
3 Modelle (70 - 200 MHz), 2 GSa/s, 14 Mpts, Bestseller, 9" Farb-TFT, FFT, USB, LAN / LXI, autom. Messungen, serielle Bustrigger (RS232, I²C und SPI), 3 Jahre Garantie

Rigol DG4062 Funktionsgenerator — 773,50 *
3 Modelle (60 - 160 MHz), 2 Kanäle, 500 MSa/s, voll arbiträr, 16 kpts, Bestseller, 3 Jahre Garantie

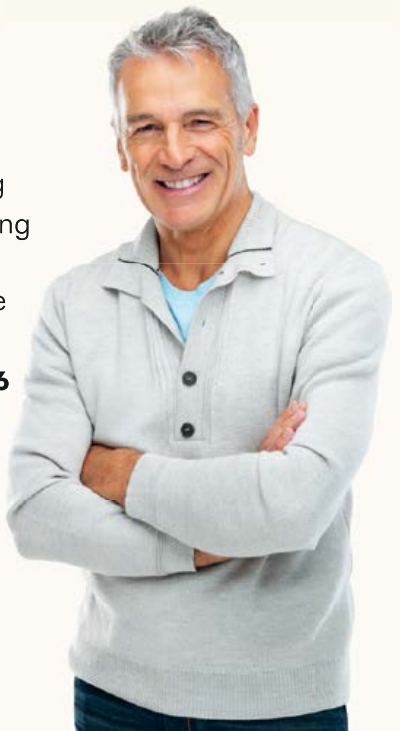
Rigol DSA815 Spektrumanalysator — 1259,02 *
9 kHz bis 1.5 GHz, intuitive Bedienung, unschlagbares Preis-Leistungs-Verhältnis, 3 Jahre Garantie

* Alle Preise sind bereits inkl. MwSt. und kostenlosem Versand in alle EU Länder.

Machen Sie Ihr LEBEN leichter.
Führende LABORTECHNIK
mit Zufriedenheitsgarantie.

Batronix Elektronik
- Große Auswahl
- Günstige Preise
- Kompetente Beratung
- Lieferung auf Rechnung
- 30 Tage testen mit Geld zurück Garantie

Jetzt Angebote nutzen:
www.batronix.com/go/26



BATRONIX
RIGOL
Premium Distributor
★★★★★

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

Elektor Android EKG

Teil 2

Draht- und knopflos mit Bluetooth und Touchscreen

Von
Marcel Cremmel (F),
in Zusammenarbeit mit
Raymond Vermeulen
(Elektor-Labor)

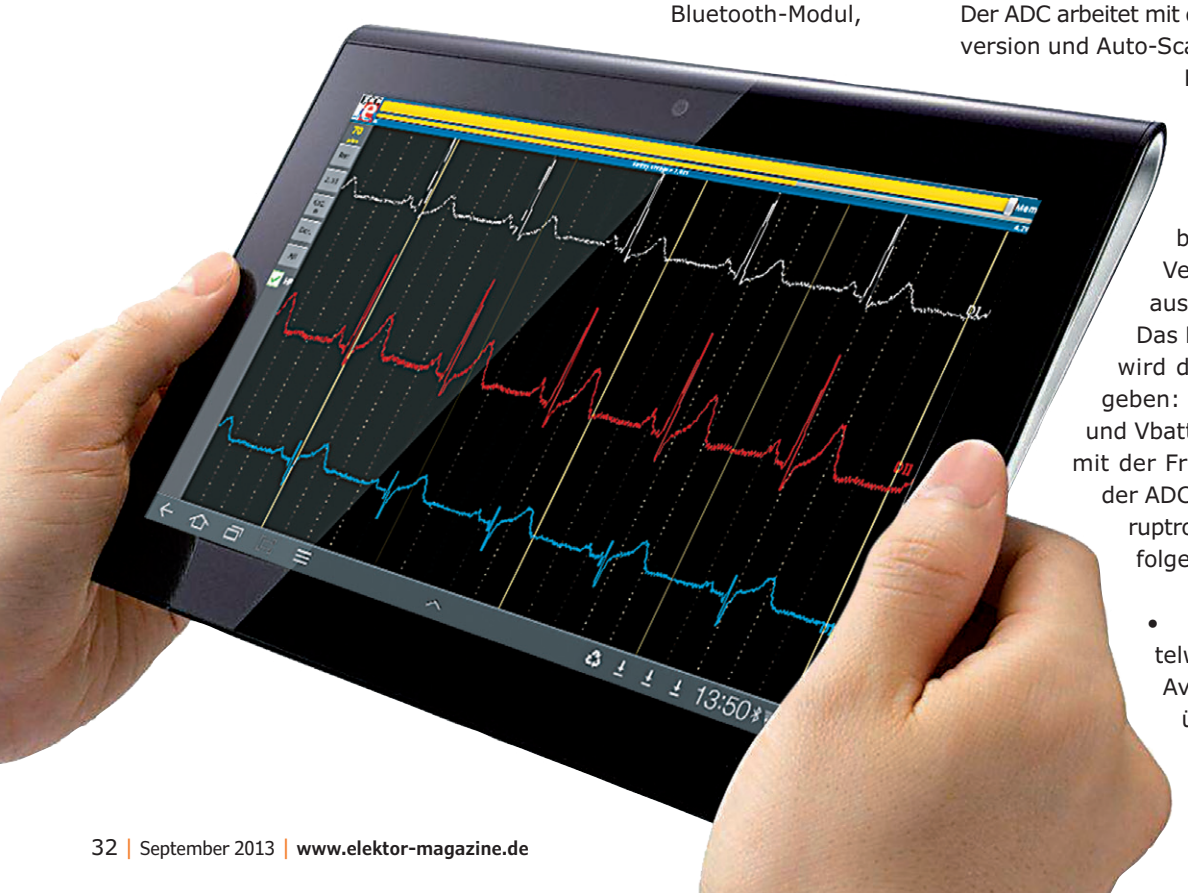
Das Elektor Android EKG wird zusammen mit einem Android-Smartphone oder -Tablet zu einem mobilen Kardioskop mit Touchscreen-Bedienung. Im ersten Teil hatten wir die Hintergründe und die Hardware besprochen, in diesem Teil geht es um die Software und die Funktionen des Mikrocontrollers. Ein weiteres wichtiges Thema ist die Android-App, die für die Bedienung und Darstellung auf dem Display zuständig ist. An dem Beispiel zeigen wir gleichzeitig, wie Apps für Android entstehen.

Funktionen des Mikrocontrollers

Analoge Signale abtasten und übertragen (Bild 5)

Diese Funktionen machen von drei Komponenten des Mikrocontrollers Gebrauch:

- 10-bit-ADC mit zugehörigem analogen Multiplexer,
- UART (*Universal Asynchronous Receiver Transmitter*) für die Kommunikation mit dem Bluetooth-Modul,
- Timer1 für das Generieren der Signale P2HZ und CAL.



Der verwendete PIC24-Typ hat nur einen ADC an Bord, der intern vorgeschaltete Multiplexer kann jedoch drei analoge Signale abtasten: DI, DII und BATT_LEV. Das Signal BATT_LEV wird über den Spannungsteiler R16/R17 aus der Akkuspannung abgeleitet.

Der ADC arbeitet mit den Betriebsarten Auto-Konversion und Auto-Scan. Dies bedeutet, dass der Prozessorkern beim Durchschalten, Abtasten und Konvertieren der drei Eingangssignale nicht beteiligt ist. Die Abtastfrequenz beträgt 2000 Hz, was für die Verarbeitung der EKG-Signale ausreicht.

Das Ergebnis der Konvertierung wird drei 16-bit-Variablen übergeben: Channel_DI, Channel_ DII und Vbatt. Nach jedem Abtastzyklus mit der Frequenz 2000 Hz generiert der ADC einen Interrupt. Die Interruptroutine `_ADC1Interrupt` führt folgende Aufgaben aus:

- Berechnen des Mittelwerts von AvgSampleDI, AvgSampleDII und AvgVbatt über acht Abtastwerte (f =

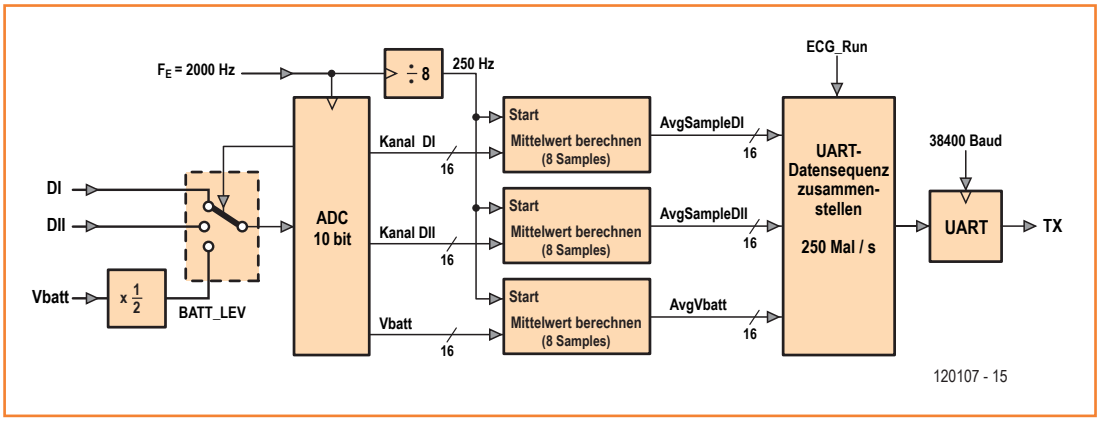


Bild 5.
Generieren und Senden der Abtastwerte von DI und DII.

PIC: Mikrocontroller mit starkem Herz

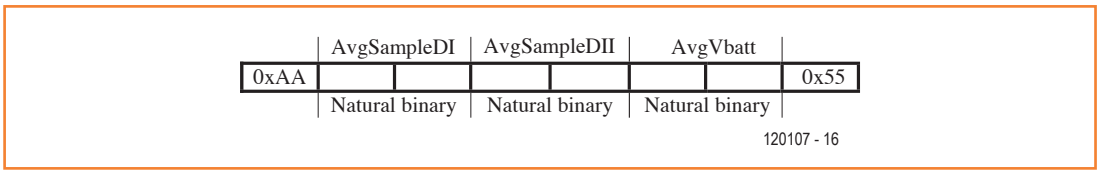


Bild 6.
Datenformat bei der Bluetooth-Übertragung.

250 Hz), so dass die Signalwerte geglättet werden können,

- Aufbauen von Daten-Frames und asynchrones Übertragen zum Bluetooth-Modul.

Ein Frame hat die Breite 8 bit, **Bild 6** zeigt die Struktur. Den Daten geht der Wert 0xAA voran, am Schluss folgt der Wert 0x55. Die Android-App wird durch diese Byte-Werte synchronisiert. Die Byte-Werte der Daten liegen im Bereich 0x0000...0x03FF. Dies entspricht positiven 10-bit-Binärzahlen, die zu 16-bit-Binärzahlen durch Nullen ergänzt sind. Damit werden Synchronisationsfehler ausgeschlossen.

Einstellen der Zeitkonstante für Auto-Zero (Bild 7)

Diese Softwarefunktion stellt fortlaufend die Geschwindigkeit nach, mit der die Signale DI und DII angepasst werden (siehe Absatz „Schaltung mit Herz“ in Teil 1 des Beitrags), so dass das EKG auf dem Display stabil dargestellt wird. Die Funktion **MovingAverageCalc()** berechnet aus den digitalen Signalen **AvgSampleDI** und **AvgSampleDII** laufende Mittelwerte über 4 s lange Intervalle. Die Ergebnisse **DI_Average** und **DII_Average** werden mit den Null-Niveaus verglichen. Anhand der Differenzen wird eine passende Zeitkonstante gewählt. Der Integrator lie-

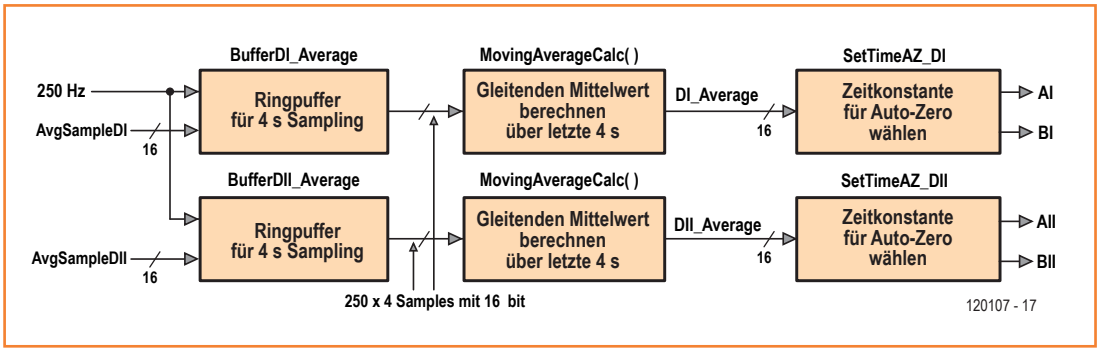


Bild 7.
Einstellen der Zeitkonstanten für Auto-Zero.

Hinweis: Die Nummerierung der Bilder und Weblinks schließt an den ersten Teil des Beitrags an.

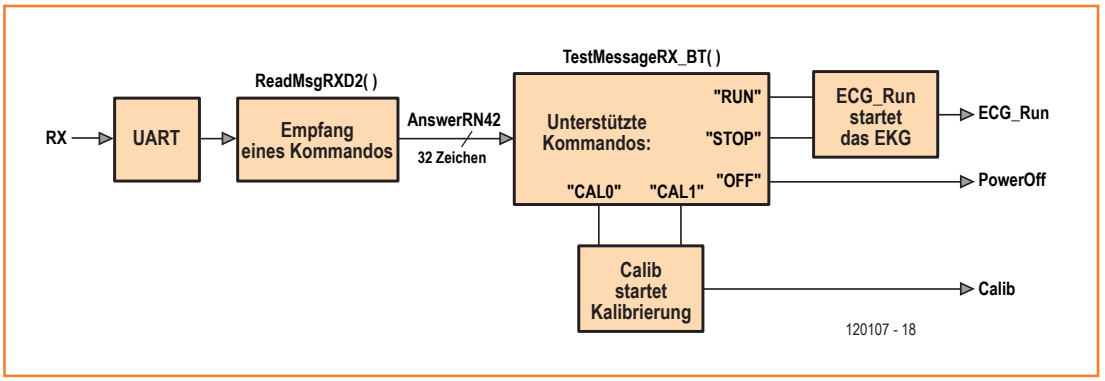


Bild 8.
Empfang der Kommandos vom Android-Gerät.

fert eine Korrekturspannung für den Offset, sie ist proportional zur gewählten Zeitkonstanten. Die passende Zeitkonstante wird mithilfe von AI und BI beziehungsweise AII und BII gewählt.
Der laufende Mittelwert wird wie folgt ermittelt: Die Abtastwerte von **AvgSampleDI** und **AvgSampleDII** werden in einem Ringpuffer der Länge 4 s gespeichert, er hat die Kapazität 4 · 250 = 1000 Worte der Länge 16 bit. Die Funktion **MovingAverageCalc()** berechnet fortlaufend den Mittelwert der letzten 1000 im Puffer befindlichen Abtastwerte. Der letzte Abtastwert korrespondiert mit dem Zeitpunkt der Berechnung, er ist folglich mitlaufend.

Empfangen der Terminal-Kommandos (Bild 8)

Der Anwender kann über das Smartphone oder Tablet folgende Kommandos senden:

- Run/Stop für das Akzeptieren oder Blockieren von Daten-Frames,
- Abschalten der EKG-Hardware, das Einschalten ist nur mit dem dort befindlichen Taster möglich,
- Erzeugen der Kommandos **CAL0** und **CAL1** für das Generieren von Kalibriersignalen.

Der UART konvertiert die empfangenen Bytes vom seriellen in das parallele Format. Die Bibliotheken von Microchip enthalten zwar Funktionen für den Byte-Empfang, sie arbeiten jedoch nicht mit Interrupts. Damit der Prozessorkern von unnötigen Warteschleifen frei bleibt, werden die empfangenen Bytes von der Interruptroutine **_U2RXInterrupt** in einem 256 Byte großen Zwischenspeicher abgelegt. Diese Bytes werden unmittelbar danach von der Funktion **ReadMsgRXD2()** gelesen. Sobald eine Zeichenkette mit dem Abschluss CR-LF vollständig empfangen wurde, wird ihr die Stringvariable **AnswerRN42** zugewiesen. Die Funktion **TestMessageRX_BT()** prüft, ob es sich bei der Zeichenkette um ein Kommando handelt. Falls dies zutrifft, folgt das Ausführen des empfangenen Kommandos. Die Flag **ECG_Run** signalisiert, dass Daten-Bytes empfangen werden können, die Flag **Calib** ist gesetzt, wenn eine Kalibrierprozedur gestartet werden muss. Das Signal /Power Off schaltet die EKG-Hardware ab.

Generieren der Kalibriersignale (Bild 9)

Die Signale P2HZ und CAL steuern den analogen Multiplexer IC9 (F2 in Bild 3) in der Weise, dass das von den EKG-Elektroden kommende Signal periodisch gegen ein Kalibriersignal der Impulshöhe 1 mV getauscht wird. Die Frequenz von P2HZ beträgt 2 Hz, und der Duty-Cycle ist 20 %, was einem EKG-Signal entspricht. Dieses vom Mikrocontroller generierte Signal wird von

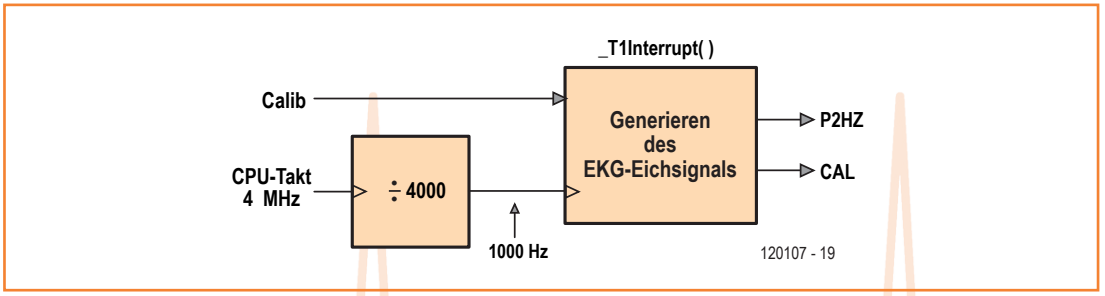


Bild 9.
Generieren der Kalibriersignale.

Heartbeats auf Ihrem Android-Smartphone oder -Tablet

R21, R22 und R65 auf 1 mV abgeschwächt, der Mittelwert über eine Periode ist Null.
Nach einem vom Anwender gegebenen Kalibrierkommando wird Signal CAL während jeder Minute für 10 s aktiv. Die Signale werden von einem Sequenzer des Mikrocontrollers erzeugt. Folgende Komponenten sind beteiligt:

- Frequenzteiler durch 4000, implementiert mit Timer1,
- Interruptroutine **_T1Interrupt**, sie wird 1000 Mal in der Sekunde aufgerufen. Wenn die Flag **Calib** gesetzt ist, werden die Zähler inkrementiert und die Inhalte mit den Konstanten verglichen, die für die Zeiten der Signalzustandswechsel festgelegt sind. Daraus entstehen die Signale P2HZ und CAL.

Status der Bluetooth-Verbindung (Bild 10)
Das Bluetooth-Modul gibt mit dem Signal STATUS an, ob die Verbindung besteht. Wenn die Verbindung unterbrochen ist, müssen keine EKG-Signale verarbeitet und gesendet werden.
Ändert sich STATUS, wird die Interruptroutine **_CNInterrupt** durchlaufen, die das Flag-Bit ECG_Run sowie das Enable-Bit ADON des ADCs steuert. Wegen der Interrupt-Funktion muss das Signal nicht periodisch geprüft werden, was zusätzliche Prozessorzeit kosten würde.

Aufbau der Firmware
Die Firmware hat eine herkömmliche Struktur, im Gegensatz zur Android-App, wie später noch deutlich werden wird. Nach dem Start werden folgende Aufgaben ausgeführt:

- Initialisieren der Variablen, der I/O-Ports, des Timer1 für das Eichsignal und des UART2 für die Kommunikation mit dem Bluetooth-Modul,
- Konfigurieren des Bluetooth-Moduls für die Geschwindigkeit 38400 Baud,
- Initialisieren des 10-bit ADCs: Abtastfrequenz 2000 Hz, Auto-Konversion und Auto-Scan für drei analoge Eingänge,
- Validieren des CN-Interrupts.
- Anschließend mündet das Programm in folgende Endlosschleifen ein:
  - Aufruf von **TestMessageRX_BT()**: Prüft, ob ein Kommando vorliegt und führt es gegebenenfalls aus,
  - Aufruf von **MovingAverageCalc()**: Berechnet den laufenden Mittelwert **DI_Average**,
  - Aufruf von **SetTimeAZ_DI()**: Ermittelt die Auto-Zero Zeitkonstante für DI,
  - Aufruf von **MovingAverageCalc()** und **SetTimeAZ_DI()** für Auto-Zero von DII.

Wie es sich für solide Software gehört, werden zeitaufwendige Vorgänge nicht in Interruptroutinen ausgeführt. Die Routinen für die Mittelwertbildung sind Bestandteile der Hauptroutine, denn sie nehmen viel Zeit in Anspruch (26800 Mikroprozessor-Zyklen, das sind 6,7 ms). Anderenfalls müssten Interruptroutinen gleicher Priorität während dieser Zeit warten, was zu Problemen führen könnte.
Der Mikrocontroller durchläuft die Endlosschleife ungefähr 75 Mal in der Sekunde. Für das Berechnen der Mittelwerte und das Einstellen der Auto-Zero-Zeitkonstante genügt dies vollkommen.

Mensch-Maschine-Schnittstelle
Android-Geräte (und auch iPhones) zeichnen sich dadurch aus, dass sie besonders anwenderfreundliche Benutzeroberflächen bieten. In Elektor sind bereits einige Beiträge zu diesem Thema erschienen. Kein Wunder, dass auch das Buch *Android Apps programmieren – Schritt für Schritt* von Stephan Schwark [4] zu Elektors Bestsellern gehört. Das Android-Programm für das Kardioskop steht auf der Elektor-Website [3] zum Download bereit. Alle Leser sind eingeladen, sich dieses Programm intensiv anzuschauen. Wenn Sie Vorschläge für Verbesserungen oder Ergänzungen haben, lassen Sie es uns wissen!
Das Programm umfasst etwa 1900 Zeilen, was zu viel ist, um an dieser Stelle detailliert erläu-

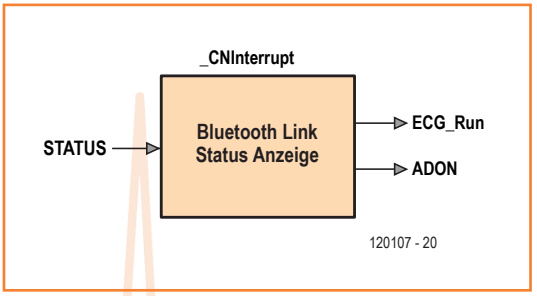


Bild 10.
Status der Bluetooth-Verbindung.

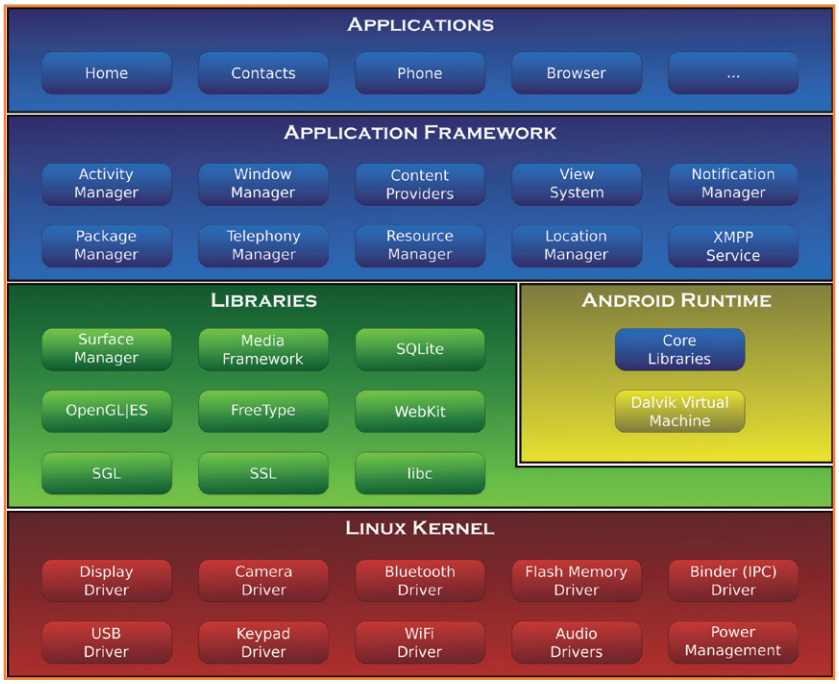
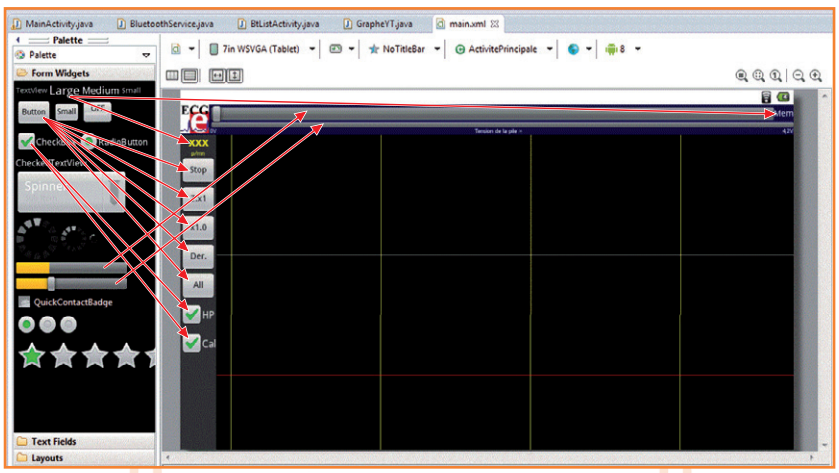


Bild 11.
Grundlegende Architektur von Android-Systemen.

tert zu werden. Stattdessen sollen hier nur die grundlegenden Strukturen und Funktionen erklärt werden. Erfahrenen Programmierern steht es selbstverständlich frei, eigene Ideen in das Programm einfließen zu lassen. Wenn Sie noch keine Erfahrungen mit Android sammeln konnten, aber erste Schritte wagen möchten, ist dies eine gute Gelegenheit.

Eine bekannte, kostenlose Entwicklungsumgebung für Android ist der App Inventor von MIT. Was die Grafik betrifft, kann der App Inventor leider nur mit mäßiger Qualität aufwarten. Die Wiedergabe eines fortlaufenden EKG stellt jedoch

Bild 12.
Die grafischen Elemente (links) werden in das Fenster (rechts) gezogen.



hohe Anforderungen. Deshalb schied App Inventor für unser Vorhaben aus. Zum Sammeln erster Erfahrungen oder zum Programmieren weniger anspruchsvoller Apps, beispielsweise das Steuern eines Mindstorm-Roboters über Bluetooth, ist der App Inventor gut geeignet.
Wegen seiner Performance haben wir zum Entwickeln der Kardioskop-App das ebenfalls kostenlose SDK von Google eingesetzt. Die SDK-Utilities gibt es für PC, Mac und Linux, sie sind Teil der verbreiteten IDE Eclipse. Die Installation von Eclipse ist zwar aufwendig, doch wenn die Anweisungen strikt beachtet werden, gelingt sie in angemessener Zeit.
Für unsere Kardioskop-App sind einige Kenntnisse der Programmiersprache Java notwendig, und das objektorientierte Programmieren (wie in C++) sollte kein Fremdwort sein. Wenn Sie in diesem Umfeld Newcomer sind, jedoch bereits in C programmieren, kommen Sie mit etwas Entdeckerfreude und Zielstrebigkeit auch hier zum Erfolg. Übrigens bietet das Internet zu den genannten Themen hervorragende Tutorien an [5]. Die ausführliche Dokumentation zum Elektor-Kardioskop finden Sie auf der Website des Autors [6].

Entwickeln für Android

Um Applikationen für eingebettete Betriebssysteme wie Android entwickeln zu können, müssen Kenntnisse der grundlegenden Architektur vorhanden sein (**Bild 11**). Der Anwender kommt ausschließlich mit den Applikationen in Berührung, die auf dem Gerät installiert sind. Dies ist die oberste Schicht des Programmiermodells. Entwickler können Applikationen in eigene Projekte übernehmen, darüber hinaus stehen ihnen umfangreiche Sammlungen in Java geschriebener APIs (*Application Programming Interfaces*) zur Verfügung. Über die APIs kann der Programmierer auf die Ressourcen und die Peripherie des Systems zugreifen. Ihrerseits machen die APIs von Bibliotheken Gebrauch, geschrieben in C und C++, die auf einem Linux-Kern laufen.
Charakteristisch für Android ist *Dalvik VM*, eine virtuelle Maschine, die Apps ausführen kann. Das Prinzip ähnelt der *Java Virtual Machine*, die auf PCs und Macs läuft: Ein Java-Compiler generiert *Bytecode*, der von der Prozessor-Plattform unabhängig ist. Anschließend wird der Bytecode in das so genannte *dex*-Format umgesetzt, wobei „dex“ für *Dalvik Executable* steht. Die Dex-Dateien werden auf dem Zielsystem installiert und ausgeführt. Durch diesen Mechanismus laufen Apps aus der

Sicht des Anwenders in genau gleicher Weise auf jeder darunter liegenden Hardware.
Eine Android-App läuft in einem eigenen Prozess unter einer eigenen *Instance* der Dalvik VM. Dalvik ist so optimiert, dass auch auf wenig leistungsstarken Systemen mit begrenztem Speicherumfang mehrere virtuelle Maschinen effizient laufen können.

Benutzeroberfläche
Mit dem Android SDK können wir die Benutzeroberfläche unserer Applikation entwerfen, ohne auch nur eine Zeile Programmcode schreiben zu müssen. Zu diesem Zweck sind umfangreiche Bibliotheken grafischer Elemente verfügbar, die lediglich an den gewünschten Orten platziert werden müssen (**Bild 12**). Die grafischen Elemente werden einfach im Drag-and-Drop-Verfahren in das aktive Fenster gezogen. So entsteht ein perfekt gestaltetes Bedienfeld, das allerdings noch keine Funktionen hat. Was noch fehlt, sind einige Zeilen Programmcode.

Ereignisse
In C enthalten Applikationen stets die Funktion `main()`, die in einer Endlosschleife die Hauptfunktionen aufruft. Im Gegensatz dazu basieren Android-Apps auf so genannten Ereignissen (*Events*). Unter Android werden nach Eintreten eines Ereignisses in Java Funktionen aufgerufen, wobei ein Ereignis beispielsweise das Anklicken einer Schaltfläche oder der Empfang einer SMS sein kann. Endlosschleifen existieren in dieser Umgebung nicht. Auch Initialisierungsfunktionen im Kopf der Apps geben nach der Ausführung die Steuerung an Android zurück. Die *Runtime Library* kann dann andere Apps abarbeiten, die gleichzeitig auf dem System laufen. Alle Events werden vom Betriebssystem verwaltet und gesteuert. In Entwicklungssystemen ist der Umgang mit Events überraschend unkompliziert.

Aktivitäten
Die meisten Android-Apps arbeiten mit Benutzeroberflächen, die aus mehreren Fenstern bestehen. Die Fenster enthalten grafische oder textorientierte Schaltflächen, die Events auslösen können. Die Reaktionen auf die Events bilden eine Aktivität. Da Android multitaskingfähig ist, müssen Aktivitäten in unterschiedliche Zustände versetzt werden können:
- Aktiv: Die Aktivität wird abgearbeitet,
- Aufgeschoben oder *Suspended*: Die Aktivität pausiert, da ein Ereignis höherer Priorität vorgezogen werden muss,
- Gestoppt: Eine *Aktivität* ist beendet, eine andere Aktivität kann starten. Die erste Aktivität kann für eine spätere Verwendung im Speicher bleiben. Falls die App beendet wird, folgt die Freigabe des belegten Speichers.

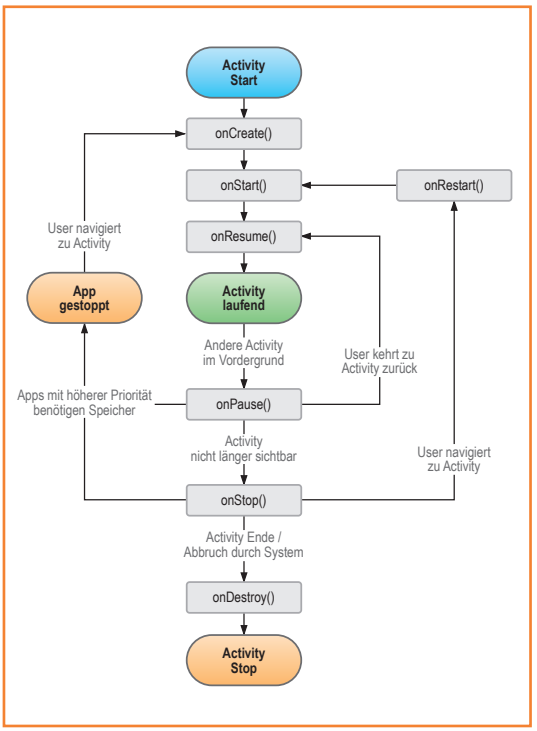


Bild 13.
Lebenszyklus einer Aktivität unter Android.

In **Bild 13** ist der Lebenszyklus einer Aktivität dargestellt, er ist für Multitaskingsysteme kennzeichnend. Unsere App ANDROECG für das Kardioskop besteht aus drei Aktivitäten:
- **MainActivity** wird zuerst ausgeführt, sie erstellt das Hauptfenster und die Schaltflächen (siehe Screenshot), außerdem richtet sie die von der App benötigten *Services* ein,
- **BtListActivity** wird nach Aufruf gestartet und gibt eine Liste der verfügbaren Bluetooth-Geräte aus, aus der Liste ist die gewünschte Schnittstelle wählbar,
- **FileListActivity** wird nach Aufruf gestartet, wenn EKG-Daten gespeichert oder abgerufen werden sollen. Eine Liste der vorhandenen Dateien erscheint, in ein Eingabefeld kann der Name einer neuen Datei eingetragen werden.

Services
Dienste, auch *Services* genannt, sind grundlegende Tasks. Die Dienste kommunizieren mit den Aktivitä-

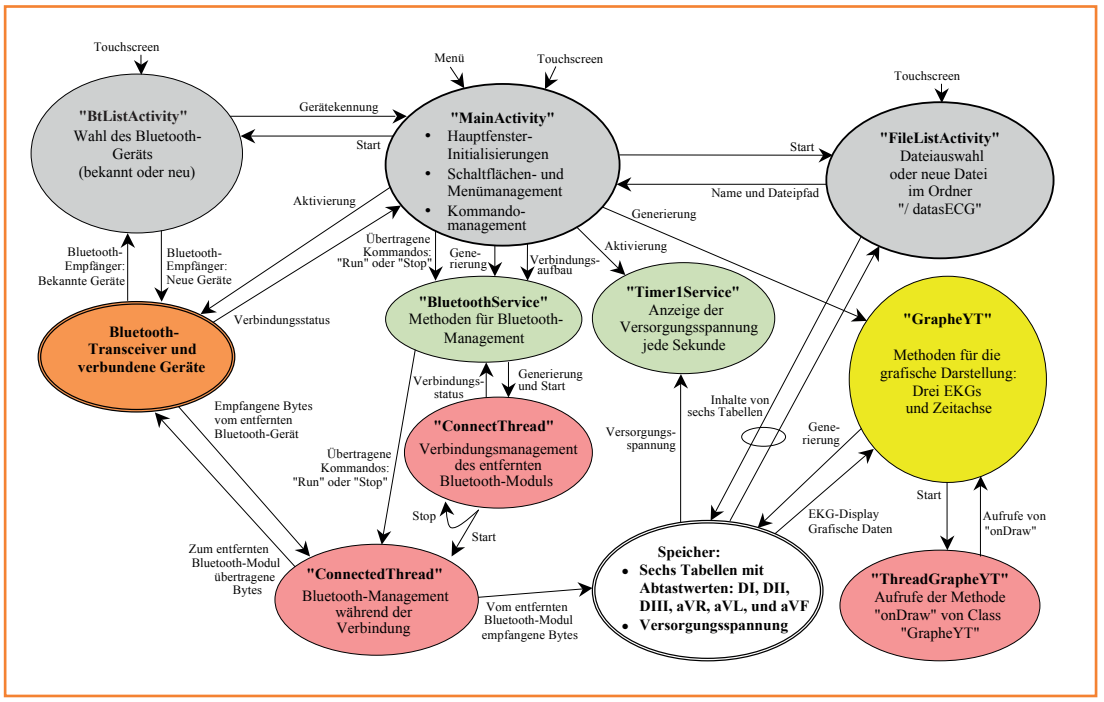


Bild 14.
Zustandsdiagramm mit den Wechselwirkungen der Aktivitäten, Dienste und Threads.

ten über so genannte *Intents* (wörtlich: Vorhaben). In ANDROECG gibt es beispielsweise den Dienst ***BluetoothService***, der das Bluetooth-Modul verwaltet. Er kann die Verbindung aufbauen, Daten senden und empfangen sowie die Verbindung beenden. ***Timer1Service*** ist eine Task, die periodisch ausgeführt wird. Dieser Dienst gibt die Höhe der Batterie- oder Akkuspannung einmal in der Sekunde auf dem Display aus.
Unter den Einstellungen des Android-Geräts sind im Menü *Applications* die Dienste aufgelistet, die zurzeit aktiv sind.

Threads

Ein *Thread* (Faden) ist die kleinste Einheit von Instruktionen, die der Task-Scheduler eines Betriebssystems als selbstständige Einheit verwalten kann. Größere Tasks bestehen aus Threads, durch Umschalten zwischen den Threads entsteht beim Anwender der Eindruck, dass die Tasks gleichzeitig ausgeführt werden. Ein Thread reagiert unabhängig von anderen Threads auf Events (zum Beispiel Anklicken von Schaltflächen, Empfangen von Daten), er führt seine Aufgabe selbstständig aus.
Ein Thread enthält immer die *Method* (Funktion) `run()`, die mit der Funktion „main“ in C vergleichbar ist. Obwohl Tasks auch gleichzeitig laufen, ist die Anzahl der Funktionen `run()` stets gleich der Anzahl der Threads.

Angenommen ein Service wird in einem Thread ausgeführt. Sobald eine App startet, erzeugt Android den Thread UI (*User Interface*). Er erkennt die vom Anwender beim Bedienen ausgelösten Events und sorgt für die Event-Behandlung. Unsere App ANDROIDECG arbeitet mit folgenden ergänzenden Threads:

- **ThreadGrapheYT** stellt das EKG fortlaufend auf dem Display dar und hat hohe Priorität, damit das EKG fließend im Bild erscheinen kann,
- **ConnectThread** stellt die Verbindung mit dem Bluetooth-Modul her,
- **ConnectedThread** steuert das Empfangen und Senden der Daten über die Bluetooth-Verbindung.

Die App: ANDROIDECG

Unsere App ANDROIDECG setzt sich aus Aktivitäten, Diensten und Threads zusammen. Die Wechselwirkungen der Elemente sind im Prinzip nicht so komplex wie vielleicht zunächst angenommen, was **Bild 14** belegt. Der Screenshot in **Bild 15** zeigt das Ergebnis.

MainActivity: Android erzeugt diese Aktivität nach Anforderung der App, indem die Methode ***onCreate()*** ausgeführt wird (Bild 13). Die Aktivität nimmt die Initialisierungen vor und legt unter anderem die Dienste *BluetoothService* und *Timer-*

1Service an. Weitere Methoden dieser Aktivität beschäftigen sich mit den Schaltflächen und Menüfunktionen. Die beiden letzten Methoden behandeln Informationen, die von anderen Aktivitäten und Diensten generiert werden. Dies können Reaktionen auf die Kommandos des Anwenders oder andere Ereignisse sein, beispielsweise die Unterbrechung der Bluetooth-Verbindung.

BtListActivity: Diese Aktivität wird nach Anklicken der Schaltfläche „Paired BT Devices" generiert (**Bild 15**). Die Bluetooth-Verbindung wird geprüft, und es erscheint eine Liste der gefundenen Bluetooth-Geräte (**Bild 16**). Nach einem weiteren Anklicken wird die Bluetooth-Verbindung noch einmal geprüft. Wird jetzt ein bestimmtes Gerät gefunden und sendet dieses Gerät seine Identität zur Hauptaktivität, schließt das Fenster. Danach startet die Hauptaktivität den *Bluetooth-Service*, um die Verbindung mit dem Kardioskop aufzubauen.

Bild 15.
Die Menüfunktionen (unten): Bluetooth-Geräte, Beenden, Speichern, Laden, EKG löschen.

BluetoothService: Dieser Dienst wird erzeugt, sobald das Bluetooth-Modul aktiv ist. Der Dienst baut die Verbindung auf und übernimmt die Steuerung, er kann auf zwei Threads zugreifen:
ConnectThread startet unmittelbar nach der Auswahl eines Geräts und fordert vom Bluetooth-Modul eine Verbindung mit dem Profil SPP an. Dies kann einige Sekunden dauern, danach ist der Thread beendet. Der Thread wird gelöscht, und der nächste Thread wird gestartet:
ConnectedThread ist aktiv, solange die Verbindung mit dem Kardioskop besteht. Dieser Thread enthält die Methoden *Write* und *Run*, sie senden oder empfangen Daten über die Bluetooth-Verbindung. Die Bluetooth-Schnittstelle arbeitet mit Frames (Bild 6), die mit der Frequenz 250 Hz aufeinanderfolgen. Aus den Frames werden sechs Tabellen mit Abtastwerten gebildet, eine Tabelle für jede EKG-Ableitung. Jede Tabelle korrespondiert mit zehn Minuten EKG-Zeit. Die Methode Run erkennt und verarbeitet die Frames. Dieser Thread wird erst beendet, wenn die Bluetooth-Verbindung abbricht oder die App geschlossen wird.

GrapheYT wird in der Hauptaktivität instantiiert. Diese *Class* (Klasse) deklariert die benötigten Variablen und Methoden für das Darstellen des EKGs. Hier sollen nur erwähnt werden:

- Die sechs Tabellen zum Speichern von zehn Minuten EKG,

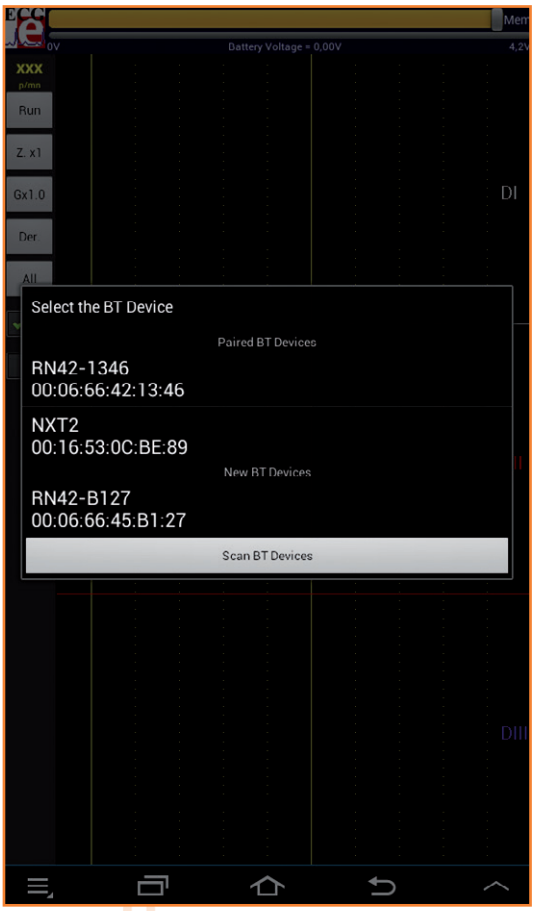


Bild 16.
Die Aktivität BtListActivity listet die gefundenen Bluetooth-Geräte und sucht nach neuen Geräten.

Refresh-Algorithmus für das EKG-Tracing

Die EKG-Daten sind in sechs Tabellen gespeichert, die einem Zeitintervall von zehn Minuten entsprechen und fortlaufend aktualisiert werden.
Jeder EKG-Ableitung DI, DII, DIII, aVR, aVL und aVF ist eine eigene Tabelle mit den Abtastwerten der letzten zehn Minuten zugeordnet.
In jeder Tabelle wird ein neuer EKG-Abtastwert gespeichert, wenn das nächste Daten-Frame eintrifft. Das geschieht 250 Mal in der Sekunde.
Im Normalbetrieb mit dem Cursor auf der rechten Seite wird der letzte Abtastwert am rechten Fensterrand dargestellt.
Die Funktion onDraw stellt die EKG-Kurve dar. Den Anfang bildet der letzte Abtastwert in der Tabelle, dann folgen in zeitlicher Reihenfolge die älteren Abtastwerte.
Bei der Zoomeinstellung „x1“ beträgt die Pixellaufgeschwindigkeit 250 Pixel in der Sekunde.

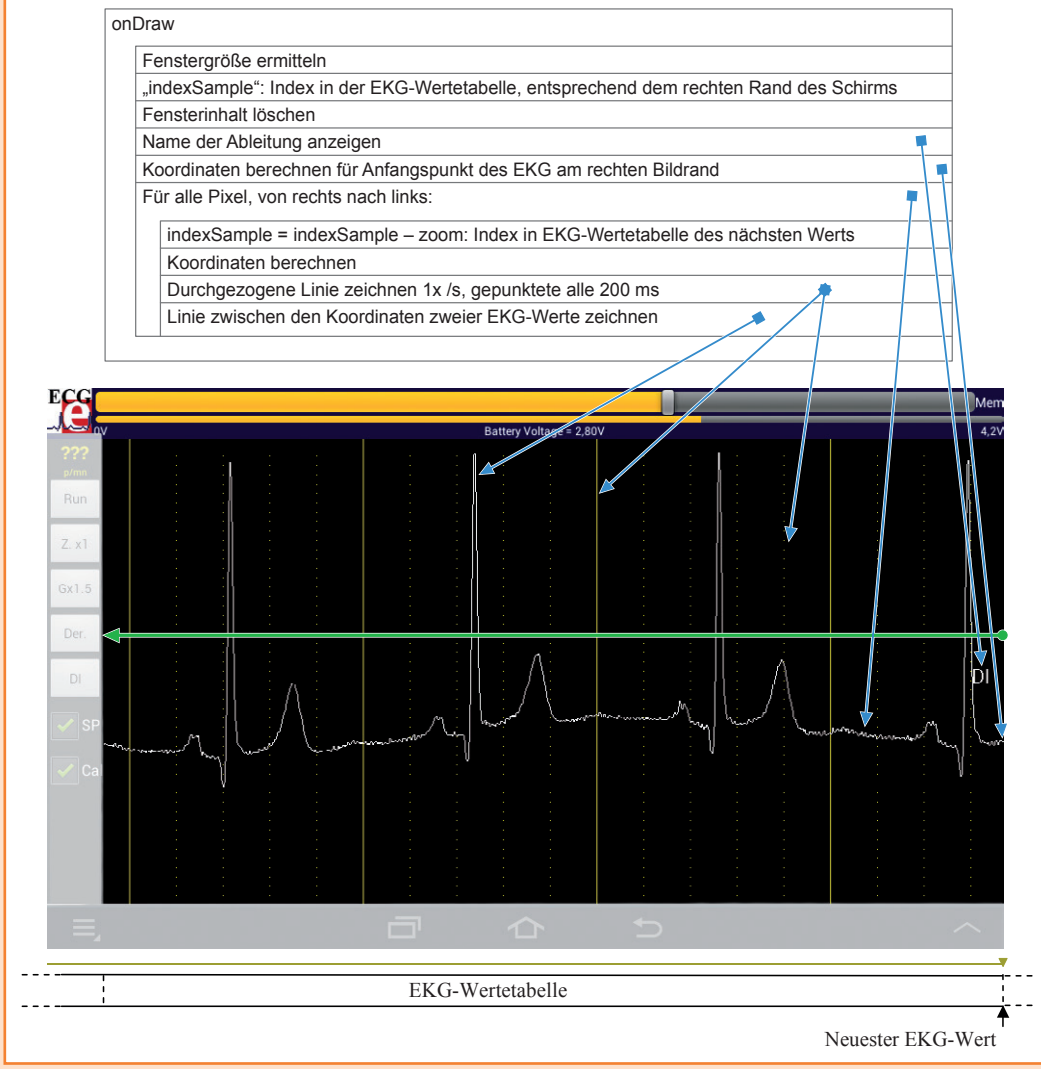


Bild 17. Algorithmus für den Display-Refresh bei fortlaufendem EKG.

Was muss das Android-Gerät leisten, um das EKG fortlaufend darzustellen?
Angenommen das Display ist 722 · 403 Pixel groß. Mit diesem Display muss die Funktion onDraw bei jedem Aufruf folgende Aktionen ausführen:
Löschen des gesamten Bildschirms, bestehend aus 722 · 403 = 290 966 Pixeln,
Darstellen der Ableitungsbezeichnungen,
Darstellen der vertikalen Bezugsachsen, die mit dem EKG mitlaufen,
Darstellen der Kurve für drei Herzschläge, unterteilt in 722 Segmente,
Berechnen der Herzfrequenz und Darstellen im Display.
Die Anzahl der Instruktionen, die der Prozessor mithilfe des grafischen Coprozessors ausführt, ist gigantisch! Hinzu kommt, dass onDraw mindestens zehnmal in der Sekunde aufgerufen werden muss, damit ein für das Auge fließender Kurvenvorschub entsteht. Vor noch nicht allzu langer Zeit waren leistungsstarke Desktop-PCs der Spitzenklasse mit dieser Aufgabe deutlich überfordert, heute laufen auf Geräten im Taschenformat nebenbei weitere anspruchsvolle Programme.

- die Methode *onDraw()*, die periodisch von *ThreadGrapheYT* aufgerufen wird und die gewählte EKG-Ableitung sowie die zugehörigen Achsen grafisch darstellt (siehe **Bild 17**, Textbox).

ThreadGrapheYT wird beim Erzeugen der Klasse GrapheYT gestartet, also zu Beginn der App. Enthalten ist die Methode onDraw(), sie wurde bereits erwähnt. Dieser Thread hat hohe Priorität, damit die Grafik fließend und fortlaufend wiedergegeben werden kann. Ob das tatsächlich geschieht, hängt von den gleichzeitig auf dem Android-Gerät laufenden Apps ab. Wenn andere Apps unerbittlich nach Rechenzeit verlangen, kann die EKG-Darstellung leicht ruckeln.

Timer1Service. Diese Klasse erzeugt einen Dienst, der einmal in der Sekunde eine recht einfache Task ausführt: Anzeigen der Spannung, die von der Batterie oder vom Akku kommt. Der Wert wird numerisch und grafisch oben im Fenster angezeigt.

FileListActivity. Diese Aktivität startet, wenn der Anwender ein EKG speichern möchte, oder wenn ein gespeichertes EKG wiedergegeben werden soll. Es erscheint eine Liste mit den Namen der gespeicherten Dateien sowie ein Eingabefeld für den Namen einer neuen Datei (**Bild 18**). Der Dateiname und der Vorgang (Speichern oder Wiedergeben) werden der Hauptaktivität übergeben, das Fenster schließt und diese Aktivität ist beendet. Für das anschließende Speichern oder Laden ist die Hauptaktivität zuständig.

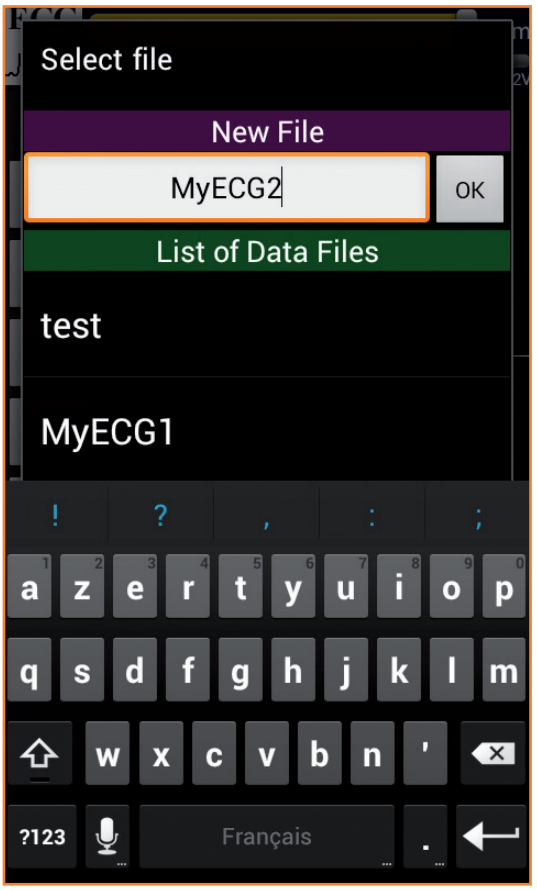


Bild 18.
Auswahl eines gespeicherten EKGs und Nameneingabe für ein neues EKG.

Technik, die von Herzen kommt!

Ein *herzliches* „Bis zum nächsten Mal!“
Damit sind wir am Ende des zweiten Teils dieses Beitrags angekommen. Im dritten Teil wird die Praxis im Vordergrund stehen, wir werden uns mit dem Bau, den Einstellungen und der Inbetriebnahme beschäftigen. Wem der Selbstbau zu zeitaufwendig ist, für den ist das Elektor Kardioskop inzwischen als startklares Modul im ***elektorPCBservice*** [7] erhältlich. Die Einstellungen sind nicht schwierig, es kommt dann nur noch darauf an, die Elektroden korrekt am Körper anzubringen. Auch dazu werden wir noch ausführliche Hinweise geben.

(130227)gd

Weblinks & Literatur

[3] www.elektor.de/120107 und www.elektor.de/130227
[4] Android Apps programmieren – Schritt für Schritt, von Stephan Schwark www.elektor.de/products/books/programming/android.2169212.lynkx
[5] Le Site du Zéro (französisch) www.siteduzero.com/informatique/tutoriels/apprenez-a-programmer-en-java oder http://goo.gl/OVZQY
[6] Website des Autors: http://electronique.marcel.free.fr
[7] www.elektorpcbservice.com

Libraries verwenden

Von **Neil Gruending**
(Kanada)

Die bisherigen Tipps und Tricks zu DesignSpark haben sich mit Setup und Konfiguration von DesignSpark aus der Perspektive einer frischen Installation beschäftigt. Nun geht es darum, wie man Libraries zur Schaltplan- und Layout-Erstellung in DesignSpark richtig einsetzt.

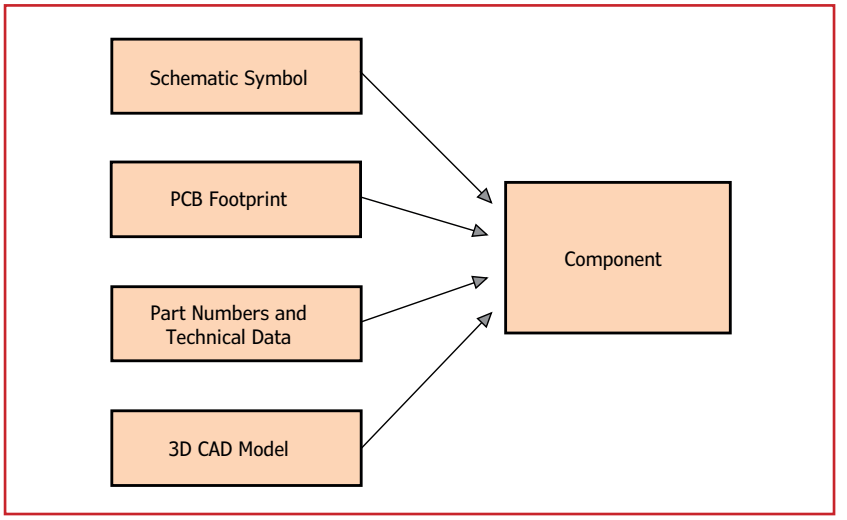


Bild 1.
Benötigte Eigenschaften eines Bauteils.

Was sind Libraries?

Als es um das Schriftfeld für Schaltpläne ging, begannen wir mit der Erstellung eines Schaltplan-Symbols in der „Schematic Symbol Library“ und einem passenden Bauteil in der „Component Library“, das mit dem Symbol verknüpft ist. In diesem Beispiel wurde eine Dokumentations-Komponente für den Schaltplan verwendet, mit der keine Layout-Informationen für die Platine verknüpft werden müssen. Normalerweise enthalten erstellte Bauteile die Informationen wie in **Bild 1**.

Das Schaltplan-Symbol ist in der Datei der „Schematic Symbol Library“ mit der Extension .ssl abgelegt. Die Layout-Informationen des Bauteils stecken in der „PCB Symbol Library“ und das 3D-CAD-Model ist in einer Datei des Typs „3D View Library“ mit der Extension .pkg gespeichert. Die oberste Ebene eines Bauteils steckt zusammen mit der Bauteilnummer und den technischen Infos in der „Component Library“ mit der Extension .cml. In dieser Library sind auch die Referenzen zu den Informationen aus den anderen Libraries enthalten, die zusammen das komplette Bauteil ausmachen.
DesignSpark nutzt also unterschiedliche Dateien für unterschiedliche Daten, sodass z.B. die Wiederverwendung von Design-Infos mit anderen Bauteilen sehr einfach ist. Man kann mehrere Widerstände erstellen, die alle das gleiche Symbol verwenden. Da diese Bauteile auf das gleiche Symbol verweisen, erreicht eine Änderung des Symbols automatisch auch die anderen Bauteile. Das gleiche Prinzip gilt auch für die Layout-Infos und die 3D-CAD-Modelle von Bauteilen. Auf der DesignSpark-Webseite findet sich ein gutes Tutorial zu Libraries und wie man sie verwendet (siehe [1]).

Library-Organisation

Die mit DesignSpark mitgelieferten Libraries sind üblicherweise unter „C:\Users\Public\Documents\DesignSpark PCB 5.0\Library“ abgelegt. Das ist ein gutes Beispiel einer großen Bauteile-Library, und zwar geordnet nach Herstellern. Ich ziehe es vor, meine Libraries nach Bauteil-Typ zu organisieren, da ich sie gleichzeitig als Datenbank für Bauteil-Nummern nutze. Ich habe beispielsweise den Typ 2N3904 in meiner Transistor-Library, aber damit habe ich verschiedene Teilenummern unterschiedlicher Hersteller assoziiert. Auf diese Weise muss ich mich nicht extra erinnern, welchen Transistor-Hersteller ich das letzte Mal genutzt habe. Auch Symbole und Layout-Informationen verwende ich in generischen Libraries wieder. Meine Library-Struktur sieht daher so aus:

- Gruppierte Component Libraries (Transistoren, Widerstände, Kondensatoren etc.)
- Eine generische Schematic Symbol Library (Widerstand, Kondensator etc.)
- Eine generische PCB Footprint Library für SMDs (0603-Layout, LQFP-Layout etc.)
- Eine generische PCB Footprint Library für bedrahtete Bauelemente (DIP-Layout, Layout für ¼-W-Widerstände etc.)

Nachdem über die Libraries gesprochen wurde, geht es jetzt an deren Verwendung, beginnend mit ModelSource.

ModelSource

ModelSource ist eine Online-Datenbank mit Bauteilen, die sich in vielen Platinen-CAD-Programmen - inklusive DesignSpark - nutzen lassen. Meine DesignSpark-Installation verbindet sich direkt mit ModelSource, sodass ich nach Bauteilen suchen kann, ohne DesignSpark zu verlassen (ein Tutorial gibt es bei [2]). Auf diese Weise hat man Zugriff auf IPC-gemäße Layout-Daten, die den Standard-Fertigungsrichtlinien entsprechen. In DesignSpark öffnet man ModelSource entweder durch Klick auf den ModelSource-Button oder via „View->ModelSource Bar“. Dann zeigt sich der ModelSource-Bildschirm wie in **Bild 2**.
Wenn man nun etwa den SMD-NPN-Transistor MMBT3904 per parametrischer Suche aufspüren will, klick man auf „CLICK TO CHOOSE“ und loggt sich notfalls ein. Eine Liste der verfügbaren Bipolar-Transistoren erhält man durch „Semiconductors->Discrete Semiconductors->Bipolar Transistors“. Als Resultat werden 740 Transistoren wie in **Bild 3** aufgelistet.
Diese Suche kann man einschränken, indem man ein paar zusätzliche Filter anwendet. Wählt man „NPN“ für „Transistor Type“, „Surface Mount“ für „Mounting Type“, „SOT-23“ für „Package Type“ und „40V“ für „Maximum Collector Emitter Voltage“, ist der zweite aufgelistete Transistor der gesuchte Typ MMBT3904. Nach einem Klick auf „Load Preview“ zeigt ModelSource das Symbol, das Layout und einige wichtige Parameter des Bauteils an (siehe **Bild 4**).
Man kann Bauteile natürlich auch direkt via „Part Number Quick Search“ suchen, wenn man zumindest einen Teil der Bauteilbezeichnung kennt. Da der Transistor aber schon gefunden wurde, klickt man auf „Use Component“, um das Bauteil in die Schaltung einzufügen. Gleichzeitig lädt DesignSpark dieses Bauteil und legt es in einer Library im Verzeichnis „Downloaded Libraries“ ab. Den kompletten Pfad erfährt man per Tab „Folders“ im Library Manager. DesignSpark zeigt den Namen der Library an, in der das Bauteil abgelegt wurde. Man kann den gefundenen Transistor aber auch einfach per Drag & Drop vom ModelSource-Fenster in die eigene Schaltung transferieren oder hierzu „Add Component“ aus der Toolbar nutzen. Doch was wenn selbst ModelSource das gesuchte Bauteil nicht enthält oder wenn Sie etwas an einem Bauteil ändern möchten? Ich würde z.B. beim MMBT3904 ein konventionelleres Symbol mit Emitter-Markierung vorziehen. Hierfür legt man besser eigene (Custom-)Libraries an.

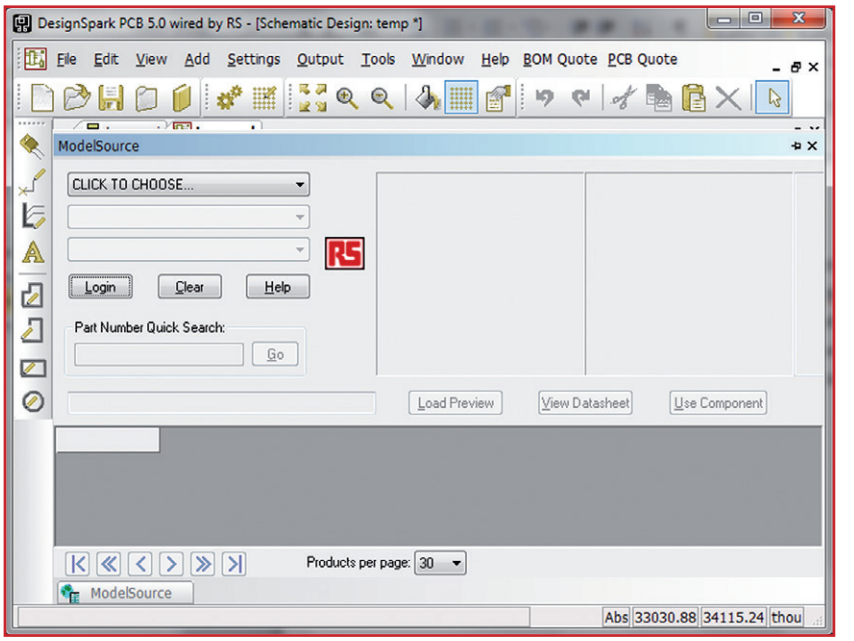


Bild 2.
Der frische Bildschirm von ModelSource.

Custom Libraries

Eigene Libraries bieten Vorteile, sind aber durchaus aufwendig. Ich kopiere gerne Bauteile aus anderen Quellen und modifiziere sie für meine Zwecke. Beim MMBT3904 würde das bedeuten, dass man das Bauteil mit dem Library Manager in eine eigene Library kopiert und anschließend nach Wunsch editiert. Bei dieser Gelegenheit kann man auch gleich kleine Fehler ausmerzen.

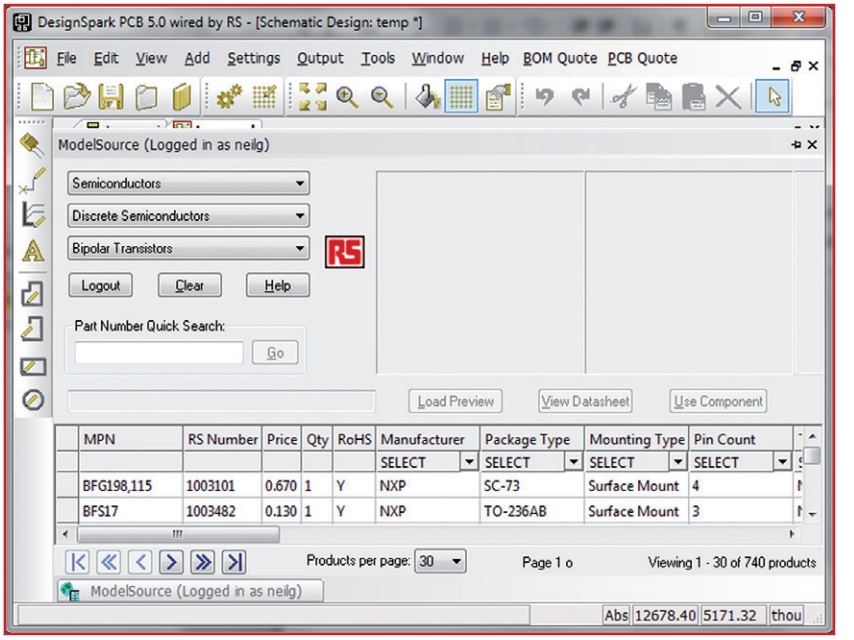


Bild 3.
Ergebnis einer Suche mit ModelSource.

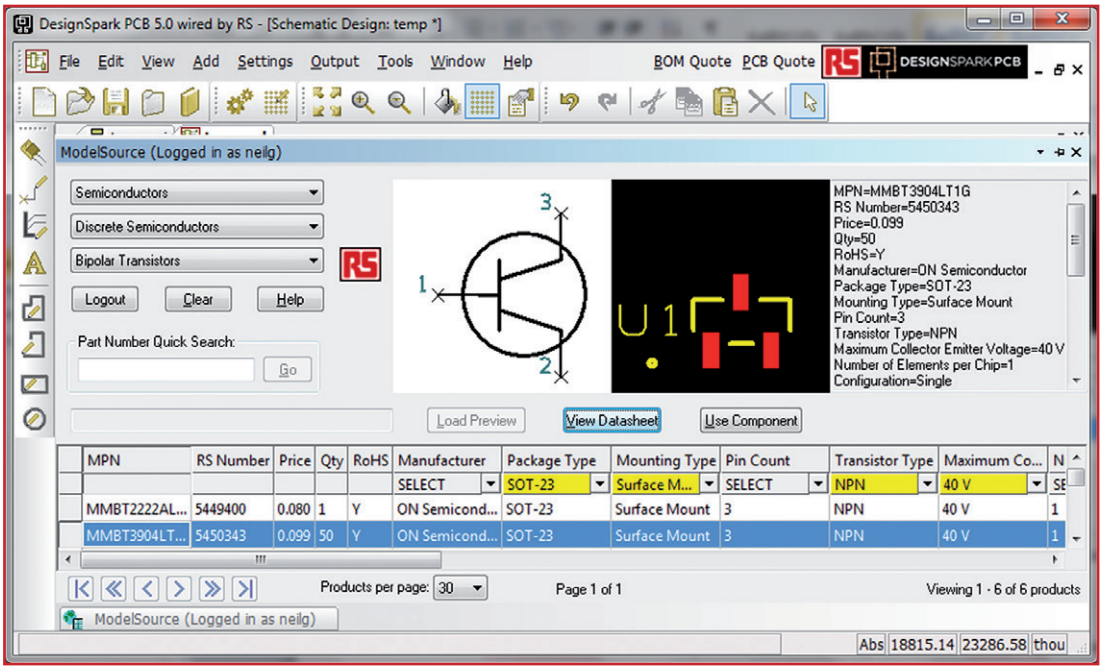


Bild 4.
Detaillierte Infos zu einem Transistor in ModelSource.

Das Wichtigste bei eigenen Libraries ist die Verwendung von Attributen für jedes Bauteil, sodass man problemlos eine Stückliste (BOM = Bill Of Material) erzeugen kann. Ich speichere deshalb alle Herstellerinformationen eines Bauteils in der Component Library. Üblicherweise gibt es mehrere Bauteilbezeichnungen unterschiedlicher Hersteller. Von daher verwende ich die Attribute wie in **Bild 5,** wo drei zusätzliche Bezeichnungen zu sehen sind. Die anderen Attribute sind für die Nutzung der BOM-Funktion von DesignSpark erforderlich.

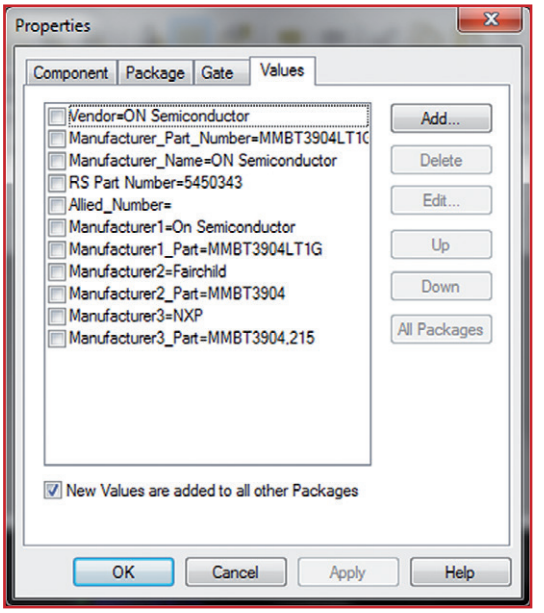


Bild 5.
Nützliche Attribute eines Bauteils.

Fazit

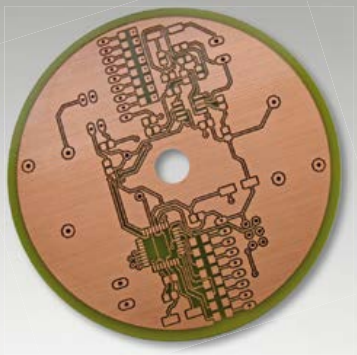
ModelSource bildet zusammen mit den Libraries von DesignSpark eine gute Grundlage, um eigene Libraries zu erstellen, die bei ihrer Nutzung eine Menge Zeit sparen können. Nun könnte man endlich eine Schaltung erstellen. In der nächsten Folge geht es um die Feinheiten beim Editieren einer Schaltung und die BOM-Funktion.

(130207)

Weblinks:

[1] www.designspark.com/tutorial/components-library-structure-library-manager

[2] www.designspark.com/eng/tutorial/components-downloading-from-modelsource-building-up-libraries



(M)ein kleiner Freund im Elektroniklabor

- Kompakt
- Einfach zu bedienen
- 33.000 U/min Spindel
- Ein- und doppelseitige Leiterplatten

LPKF ProtoMat E33 – klein, präzise, wirtschaftlich
Kaum größer als ein DIN A3-Blatt: LPKF Qualität zum Einstiegspreis zum Fräsen, Bohren und Trennen von Leiterplatten und Gravieren von Frontplatten. **www.lpkf.de/prototyping**

LPKF Laser & Electronics AG Tel. +49 (0) 5131-7095-0

LPKF
Laser & Electronics

NEU

LabVIEW 2

Arrays und serielle Daten

Der zweite Band der LabVIEW-Lehrbuchreihe beschäftigt sich u. a. mit Arrays, Cluster und den seriellen VISA-Funktionen. Als Erstes werden vier neue zusammengesetzte Datentypen (Enum, Ring, Array, Cluster) vorgestellt und deren Verwendung wird anhand zahlreicher praktischer Beispiele und Übungen erläutert.

Danach wird es praktisch: Ein 8051er-Mikrocontrollersystem dient als Datenquelle und -senke für verschiedene LabVIEW-VIs, die einerseits beliebige Messdaten empfangen, auswerten und die Ergebnisse numerisch oder graphisch auf dem Rechner-Monitor darstellen. Anderseits können sie auch Steueranweisungen an das Mikrocontrollersystem ausgeben.
Der gesamte Datentransfer zwischen Mikrocontroller und LabVIEW-VI läuft dabei über serielle COM-Schnittstellen oder mittels USB/Seriell-Adapter über USB-Ports.

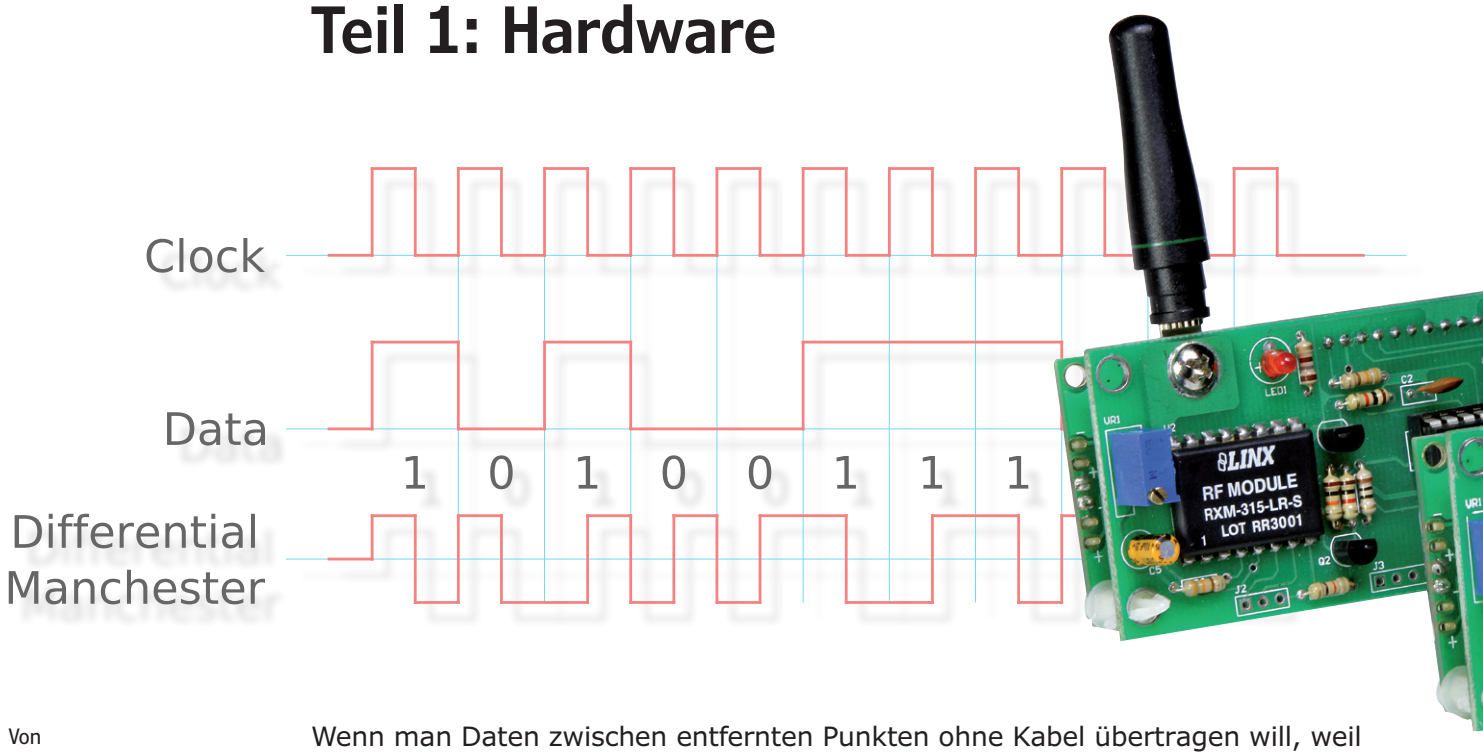
Da alle VIs universell aufgebaut und ausführlich erklärt werden, lassen sie sich problemlos mit anderen Mikrocontrollerboards (AVR-, PIC-, ARM-, Arduino-, Raspberry PI-Systeme, E-blocks) kombinieren, sofern diese über eine serielle Schnittstelle verfügen.

248 Seiten (kart.) · Format 17 x 23,5 cm · ISBN 978-3-89576-274-1
€ 34,80 · CHF 43,20

elektor **Weitere Infos & Bestellung unter www.elektor.de/labview2**

Modulare Funkverbindung mit Manchester-Code (1)

Teil 1: Hardware



Von
Marcelo Maggi (USA)

Wenn man Daten zwischen entfernten Punkten ohne Kabel übertragen will, weil entweder die Infrastruktur keine Verkabelung bietet, weil mindestens eines der beiden Geräte portabel sein soll oder aber weil schlicht Funk präferiert wird, gibt es mehrere Lösungen, die alle spezifische Vor- und Nachteile mit sich bringen. Also untersucht man die Sache, löst die Probleme und wirft den Lötkolben an.

In diesem Artikel geht es um ein Verfahren, Daten per Funk in den ISM-Bändern 315 MHz oder 433 MHz mit einer Datenrate von 5.000 bit/s zu übertragen und dabei preiswerte aber zuverlässige Komponenten zu verwenden. Es wird dabei mit dem Manchester-Code ein HF-taugliches Protokoll implementiert, mit dem zuverlässig Entfernungen bis zu 200 m überbrückt werden können. Wir verwenden mit einem Sender (TX) und einem Empfänger (RX) zwei Funkmodule, die sich für beliebige Anwendungen eignen. Man muss lediglich den Code ihrer Mikrocontroller entsprechend anpassen.

TX, RX und Antenne

Eine Funkverbindung erfordert Hardware und Software bzw. Firmware sowohl auf TX- als auch auf RX-Seite. Dieser erste Teil dieses Artikels beschäftigt sich mit der Hardware.

Die Entwicklung von HF-Schaltungen mit diskreten Bauteilen ist keine einfache Sache. Oft sind die Ergebnisse eher enttäuschend bis schlecht. Die Firma Linx Technologies bietet glücklicherweise mit kompletten, in einem Hybrid-Gehäuse gekapselten Funkmodulen einen guten Umweg. Solche Module gibt es für viele Frequenzen, doch hier geht es um die Bänder mit 315, 418 und 433 MHz, denn diese sind (je nach Land) oftmals frei.

Bild 1 zeigt das TX-Modul und seine Anschlussbelegung laut Datenblatt [1]. **Bild 2** zeigt mit dem RX-Modul das Gegenstück. Es gibt nur wenig aktive Pins. Der Empfänger hat zwar mehr Pins, doch die meisten sind nicht angeschlossen (NC).

Kompatibel zu Lochrasterplatinen
5.000 bit/s über 200 m

Außer der Stromversorgung braucht man zum grundlegenden Betrieb lediglich eine Antenne. Die Antenne ist für gute Funkverbindungen essentiell. Auch hier hilft Linx Technologies durch eine Lösung zur elektrischen Verlängerung kurzer Antennen. Von daher werden kurze Stabantennen mit ¼-Wellenlänge eingesetzt. **Bild 3** zeigt die 315-MHz-Antenne.

Die Frequenz auf die die Antenne abgestimmt ist, kann man einem Farb-Code entnehmen: Ein grüner Ring an der Antennenbasis steht für 315 MHz; ein blauer Ring für 418 MHz und rot für 433 MHz.

TX: Der Sender

Das TX-Modul sendet schlicht die Signale am DATA-Pin. Ein paar Einschränkungen gibt es: Da das Modul ohne eigene Intelligenz ist, gibt es keine Daten-Synchronisation, keinen Code-Schutz etc. Diese Feinheiten sind Sache des Anwenders. **Bild 4** zeigt die Schaltung des Senders. Sie ist recht einfach, denn es gibt nur eine Verbindung zwischen den beiden wichtigen ICs: Pin B0 (RB0/INT, Pin 6) des Mikrocontrollers ist mit dem DATA-Eingang des TX-Moduls verbunden. Die restlichen Bauteile sind für den korrekten Betrieb der beiden Haupt-ICs notwendig, spielen aber für die Datenübertragung keine weitere Rolle. Bei C1, C2, C3, C7 und C8 handelt es sich beispielsweise um keramische 100-nF-Kondensatoren zur Entkopplung. R5 hält den Reset-Pin des Mikrocontrollers (Pin 4) auf „high“. Als Oszillator ist ein normaler 20-MHz-Quarz vorgesehen. Das TX-Modul wird mit 3 V versorgt. Die externe 5-V-Versorgung wird hierfür mit dem Spannungsregler LP2950-30LPR auf 3,0 V gebracht. C1, C5, C6 und R4 dienen laut Datenblatt der Stabilität. Das TX-Modul ist noch mit zwei weiteren Widerständen beschaltet. R3 zieht den PD-Pin auf die 3-V-Schiene und sorgt so für ein „high“. Bei „low“ würde der Sender in den Stromspar-Modus gehen, in dem nicht gesendet werden kann. Der Wert von R1 bestimmt die Sendeleistung. Mit 0 Ω ist die Leistung am größten. Sollten nur geringere Leistungen zulässig sein, kann man diese anhand von **Bild 5** durch andere Werte von R1 auf die gewünschten Sendepegel reduzieren. Da der Sender mit 3 V betrieben wird, dürfen auch am DATA-Eingang keine 5-V-Pegel vorkommen. R8 und R9 bilden daher einen Spannungsteiler,

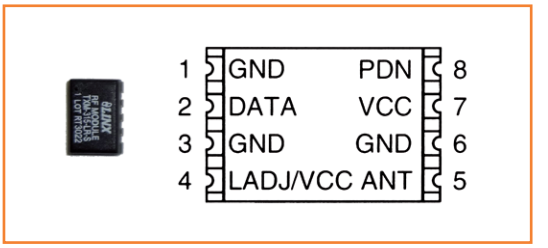


Bild 1.
Das TX-Modul von Linx Technologies samt Anschlussbelegung. Nur wenige Anschlüsse sind notwendig.

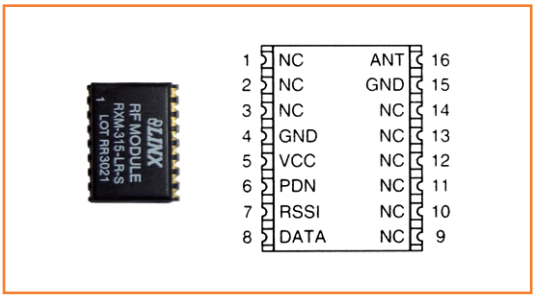


Bild 2.
Das RX-Modul von Linx Technologies samt Anschlussbelegung. Viele Pins sind nicht angeschlossen.

Bild 3.
Die Antenne ist ein (verlängerter) λ-¼-Strahler.

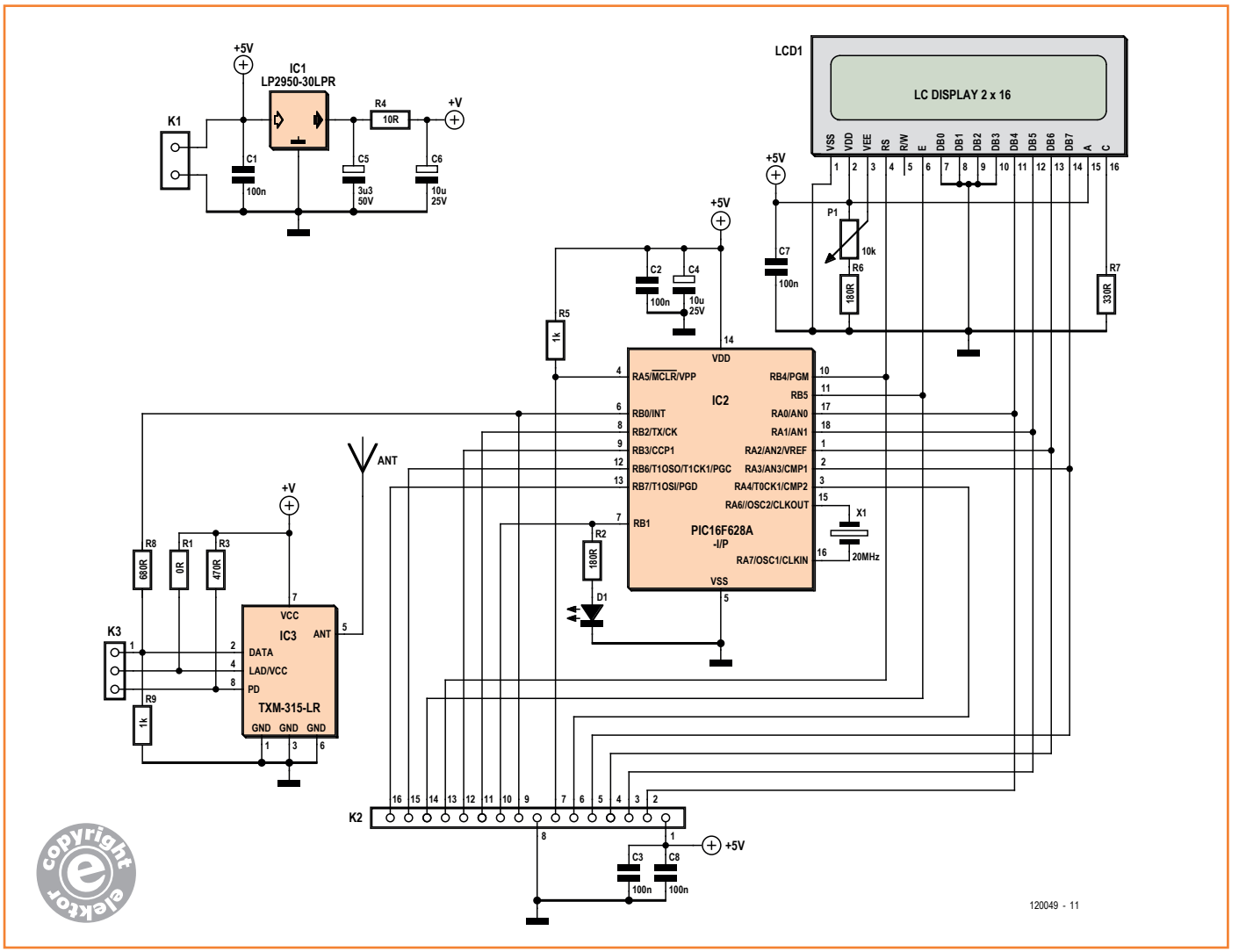


Bild 4.
Die Schaltung des Senders. Ein PIC steuert gleichzeitig das LCD und die Kommunikation.

der die 5-V-Pegel vom Mikrocontroller auf die gewünschten 3 V bringt.

Man könnte viele unterschiedliche Mikrocontroller einsetzen. Beim hier vorgesehenen PIC der Firma Microchip handelt es sich um einen Mikrocontroller mit 18 Pins. Das Modell PIC16F628A mit 3,5 KB Speicher reicht aus. Mit einem PIC16F648A hat man nicht nur 7 KB Flash, sondern auch mehr SRAM und EEPROM, wodurch größere Programme möglich werden. Sogar der altbekannte Typ PIC16F84A eignet sich, was sehr praktisch ist, wenn man sich schon gut damit auskennt.

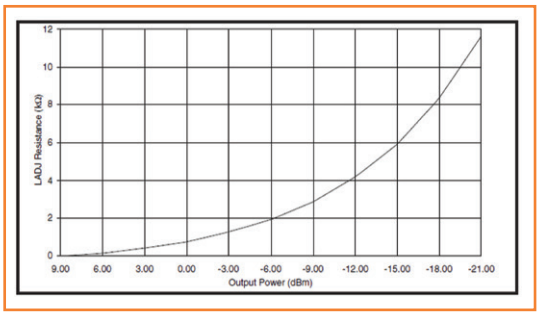
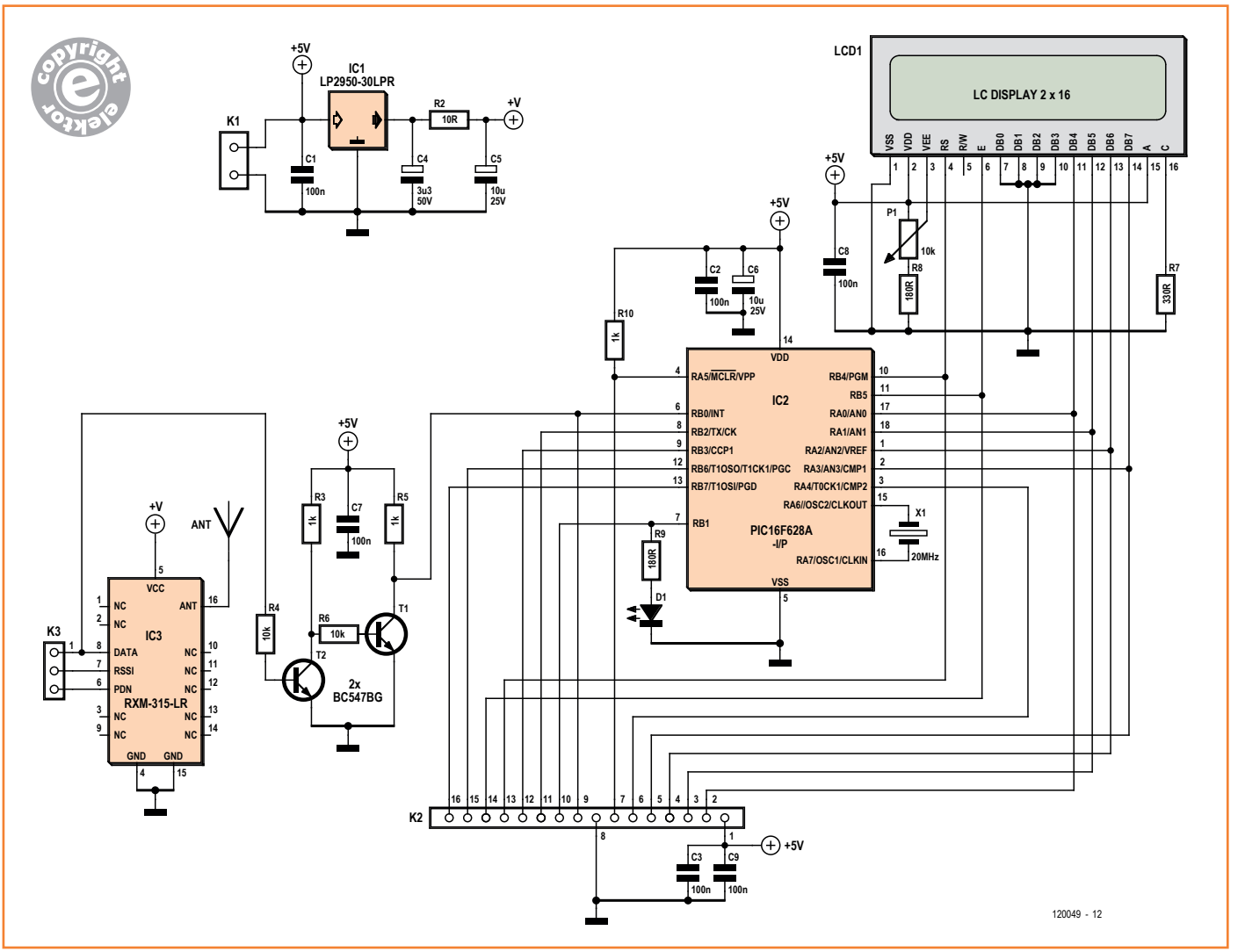


Bild 5.
Die Sendeleistung hängt vom Wert von R3 ab.

Im Schaltbild kann man sehen, dass die relevanten Signale des TX-Moduls via K3 und die des Mikrocontrollers via K2 extern zugänglich sind. Man hat also volle Kontrolle von außen über die Elektronik und kann notfalls sogar den Mikrocontroller zurücksetzen. Beim TX-Modul sind die zu sendenden Daten, der PD-Pin und die Sendeleistung extern zugänglich. Man kann also einfach den PIC aus seiner Fassung ziehen und über K3 einen beliebigen anderen Mikrocontroller die Steuerung übernehmen lassen. Man sollte dann aber nicht vergessen, die anliegenden Pegel auf maximal 3 V zu begrenzen. Letztlich muss man nur die



externe 5-V-Versorgung anlegen und schon hat man einen zuverlässigen Datensender in Betrieb. Nachtrag: Wer Erfahrung mit HF-Elektronik hat, wird es vermeiden, Schaltungen mit Frequenzen jenseits von 10 MHz auf einer Lochrasterplatine zu testen, da sich so keine stabilen Verhältnisse ergeben. Das Modul auf Platine aber ist auf jeden Fall kompatibel zu Lochrasteraufbauten, da die komplette HF ja gekapselt im TX-Modul steckt.

RX: der Empfänger

Ein Blick auf die Schaltung des Empfängers in **Bild 6** bringt viele Ähnlichkeiten zum Sender ans Licht. Es gibt mit C2, C3, C7, C8 und C9 etliche Kondensatoren zur Entkopplung, den gleichen 3-V-Spannungsregler (auch das RX-Modul benötigt 3 V), die gleiche Entstörung durch C6 wie beim Sender, mit R10 einen Pull-up-Widerstand am Reset-Pin des Mikrocontrollers und einen 20-MHz-Quarz als Taktgenerator. Wenn im zweiten Teil die Software an die Reihe kommt, wird man sehen, wie wichtig es ist, dass die Mikrocontroller von Sender und Empfänger den gleichen Takt haben. Auch die Steckverbinder für den Kontakt zur Außenwelt sind ähnlich belegt. Der einzige Unterschied ist, dass das RX-Modul anstelle des Pins für den Sendepegel einen Ausgang hat, der am RSSI-Pin (Received Signal Strength Indicator) die Empfangsfeldstärke ausgibt. Das analoge RSSI-Signal liegt daher mit der Mitte von K3 auf der gleichen Position wie das LAD/VCC-Signal des Senders, das es ja bei einem Empfänger nicht geben kann.

Die von Pin 7 (RB1) von IC2 über R9 angesteuerte LED D1 signalisiert, ob eine Funkverbindung vorhanden ist oder nicht. Mit einem einfachen

Bild 6.
Die Empfänger-Schaltung ähnelt der des Senders.

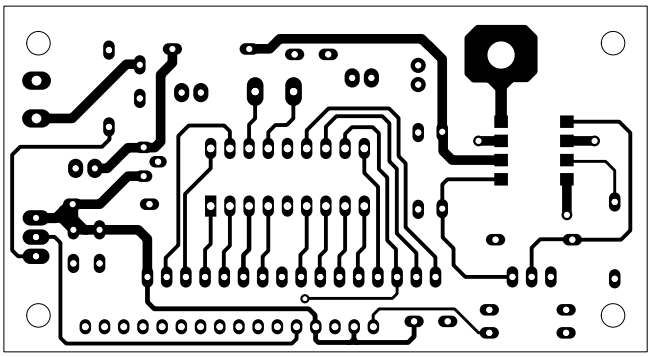


Bild 7. Bestückungsseite der Sender-Platine.

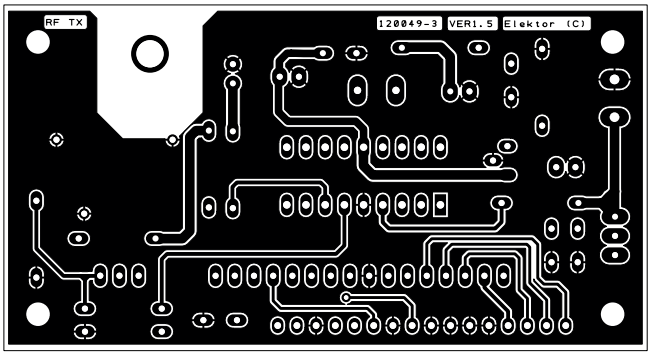


Bild 8. Lötseite der Sender-Platine.

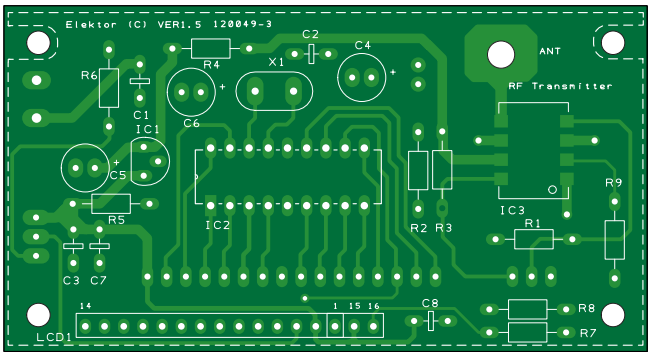


Bild 9. Bestückungsplan der Sender-Platine.

Programm kann man den Sender einen Befehl zur Aktivierung von Pin RB7 des Empfänger-Mikrocontrollers senden lassen. Wenn dieser Befehl richtig abgeschickt und empfangen wurde, leuchtet D1. Dieser Punkt wird bei der Beschreibung der Software noch einmal genauer erläutert.
Die zweifach invertierenden Transistoren T1 und T2 bilden in Kombination einen nichtinvertierenden Pegel-Konverter, der die 3 V aus dem DATA-Ausgang des RX-Moduls auf die vom Mikrocontroller-Eingang benötigten 5 V hievt.
Zwei Hinweise an dieser Stelle:

- Ja, der Mikrocontroller hätte auch mit 3 V betrieben werden können. Doch um die Schaltung universal einsetzbar zu machen, ist sie für 5-V-Mikrocontroller ausgelegt. Es können also auch ältere PICs verwendet werden, die nur mit 5 V betrieben werden können.
- Mit einem invertierenden Pegel-Konverter hätte man einen Transistor und einen Widerstand einsparen können. Dann aber hätte man die Invertierung in der Software kompensieren müssen. Da die Sache einfach

Stückliste Sender

Widerstände:
R1 = 0 Ω
R2,R6 = 180 Ω
R3 = 470 Ω
R4 = 10 Ω
R5,R9 = 1 k
R7 = 330 Ω
R8 = 680 Ω
P1 = 10 k, Mehrgang-Trimmpoti

Kondensatoren:
C1..C3,C7,C8 = 100 n, keramisch
C4,C6 = 10 µ/25 V, Elko
C5 = 3µ3/50 V, Elko

Halbleiter:
D1 = LED, rot
IC1 = LP2950-30LPR
IC2 = PIC16F628A-I/P
IC3 = TXM-315-LR, Linx Technologies (418 oder 433 MHz)

Außerdem:
ANT = ANT-315-PW-LP, Linx Technologies
K1 = 2-pol. Schraubklemme für Platinenmontage, RM 5 mm
K2 = 16-pol. Pfostenstecker, RM 0,1''
K3 = 3-pol. Pfostenstecker, RM 0,1''
LCD1 = 2x16 Zeichen, DEM16217, Elektor #120061-71
X1 = Quarz, 20 MHz
Platine #120049-3

gehalten werden soll, wurde der (kleine) Aufwand eben in der Hardware getrieben. Bei einer Massenproduktion würde man aber die paar Cents dafür sicher auch noch einsparen.

Wie beim Sender kann man auch beim Empfänger einfach den PIC aus seinem Sockel entfernen und stattdessen einen beliebigen anderen Mikrocontroller über K3 anschließen. Auch die Empfänger-Platine ist kompatibel zu Aufbauten mit Lochrasterplatinen. Der Empfänger kann sehr einfach an andere Elektronik angeschlossen werden.

Bau des Senders

Linx Technologies stellt pin-kompatible TX-Module mit drei unterschiedlichen Frequenzen her. Man kann sie also sehr leicht austauschen. Wie schon erwähnt benötigen HF-Schaltungen besondere Umsicht, damit das Ergebnis nicht enttäuscht. Auch wenn die Module von Linx Technologies die Sache sehr vereinfachen und so relativ leicht stabile Verbindungen möglich sind, sollte man sich doch an die Empfehlungen bezüglich des Platinen-Layouts halten, um gute Leistungen zu erzielen. Hier-

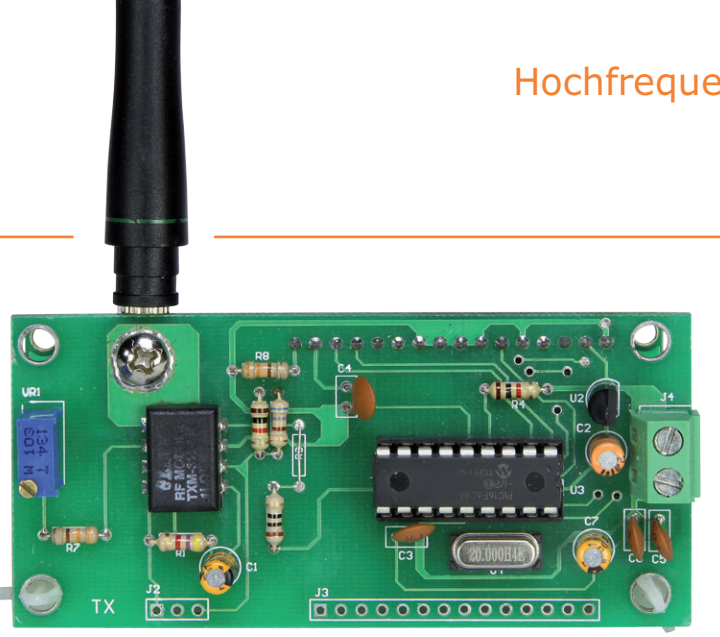


Bild 10. Der erste Sender-Prototyp benötigte noch ein paar Änderungen.

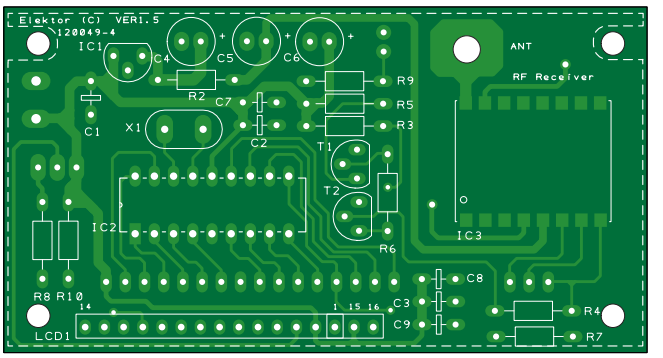


Bild 11. Bestückungsplan der Empfänger-Platine.

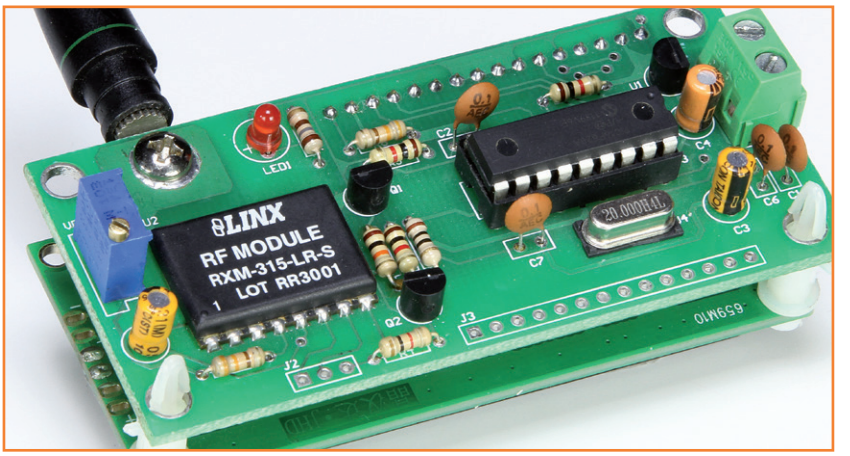


Bild 12. Das LCD passt genau auf die Rückseite des Empfänger-Prototypen.

Stückliste Empfänger

Widerstände:

R1 = nicht bestückt
R2 = 10 Ω
R3,R5,R10 = 1 k
R4,R6 = 10 k
R7 = 330 Ω
R8,R9 = 180 Ω
P1 = 10 k, Mehrgang-Trimmpoti

Kondensatoren:

C1..C3,C7,C8,C9 = 100 n
C4 = 3µ3/50 V, Elko
C5,C6 = 10 µ/25 V, Elko

Halbleiter:

D1 = LED, rot
IC1 = LP2950-30LPR
IC2 = PIC16F628A-I/P
IC3 = RXM-315-LR, Linx Technologies (418 oder 433 MHz)
T1,T2 = BC547B

Außerdem:

ANT = ANT-315-PW-LP, Linx Technologies
K1 = 2-pol. Schraubklemme für Platinenmontage, RM 5 mm
K2 = 16-pol. Pfostenstecker, RM 0,1''
K3 = 3-pol. Pfostenstecker, RM 0,1''
X1 = Quarz, 20 MHz
LCD1 = 2x16 Zeichen, DEM16217, Elektor #120061-71
Platine #120049-4

für gibt es drei wichtige Regeln zu befolgen:

- Die (doppelseitige) Platine muss auf der dem Modul gegenüberliegenden Seite eine Massefläche haben.
- Es sollten keine anderen Leiterbahnen unter dem Modul hindurch führen. Im Abstand von 3,8 mm von den Seiten und von der Oberfläche des Moduls dürfen sich keine leitfähigen Gegenstände (Bauteile) befinden.
- Der Anschluss der Antenne sollte so kurz wie möglich sein.

Mit diesen Vorschriften im Hinterkopf haben wir eine Platine rund um das TX-Modul entworfen und die restlichen Bauteile passend angeordnet. **Bild 7** zeigt das Layout der Sender-Platine von der Bestückungsseite. In **Bild 8** ist die Lötseite zu sehen. Man beachte die isolierte Positionierung des TX-Moduls, die Massefläche auf der Lötseite und die Tatsache, dass nur wenige Leiterbahnen auf der Bestückungsseite zu sehen sind. Wie empfohlen ist die Leiterbahn zur Antenne sehr kurz geraten. Die Antenne wird direkt auf die Platine geschraubt. Etwas Lötzinn sorgt gleichzeitig für mechanische Stabilität und eine gute elektrische Verbindung. Das Rastermaß von K2 und K3 ist 0,1'' (2,54 mm), so dass die Platine gut zu einer Lochrasterplatine oder einem Steckbrett passt. **Bild 9** zeigt den Bestückungsdruck des Senders. Die Platine ist mit 85 × 46 mm recht klein. In **Bild 10** ist der erste Prototyp der Schaltung zu sehen. Für den Mikrocontroller ist ein Sockel vorhanden, das macht einen Austausch einfach.

Bau des Empfängers

Alles was zum Bau des Senders gesagt wurde, gilt auch für den Empfänger: Die Massefläche muss vorhanden sein und das RX-Modul muss ausreichend Abstand zu anderen Bauteilen haben. Genauso die Antenne: Ihre Verbindung sollte kurz ausfallen und nach ihrer Verschraubung mit der Platine sollte noch etwas Lötzinn spendiert werden.

Die Layout-Dateien für Sender und Empfänger können kostenlos von der Webseite zu diesem Artikel [2] herunter geladen werden. Der Bestückungsplan des Empfängers ist in **Bild 11** zu sehen. Die Maße der Platinen von Sender und Empfänger sind identisch.

Auch beim Rastermaß von 0,1'' für K2 und K3 und deren Positionen auf den Platinen gibt es keinen Unterschied zwischen Sender und Empfänger. Wie beim Sender hat Linx Technologies pin-kompatible RX-Module mit den drei unterschiedlichen ISM-Frequenzen im UHF-Bereich im Lieferumfang.

Bild 12 zeigt den ersten Prototypen des Empfängers. Auch hier ist der Mikrocontroller gesockelt. Auch wenn es eigentlich selbstverständlich ist, sei nochmals darauf hingewiesen: Für eine gute HF-Verbindung müssen alle HF-Bauteile wie TX- und RX-Modul plus TX- und RX-Antenne auf die gleiche Frequenz abgestimmt sein. Die Antenne kann nicht zwischen gesendeten oder empfangenen Wellen unterscheiden und deshalb lässt sich das gleiche Modell sowohl für den Sender als auch für den Empfänger nutzen. Auch wenn die verwendeten Antennen sehr gut sind, so sind sie dennoch nicht perfekt. Um staatliche Vorschriften einzuhalten kann es nötig sein, die Sendeleistung oder gar den Gewinn der Sendeantenne zu begrenzen. Auf der Empfängerseite hingegen kann man gar nicht genug Antennengewinn haben. Von daher können sich Experimente mit diversen Antennentypen durchaus lohnen.

Hiermit ist der erste Teil des Artikels abgeschlossen. Nächsten Monat geht es im zweiten Teil weiter mit der Software, wodurch die Hardware erst zu einer breit einsetzbaren Funkstrecke wird.

(120049)

Weblinks

[1] www.linxtechnologies.com

[2] www.elektor.de/120049

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

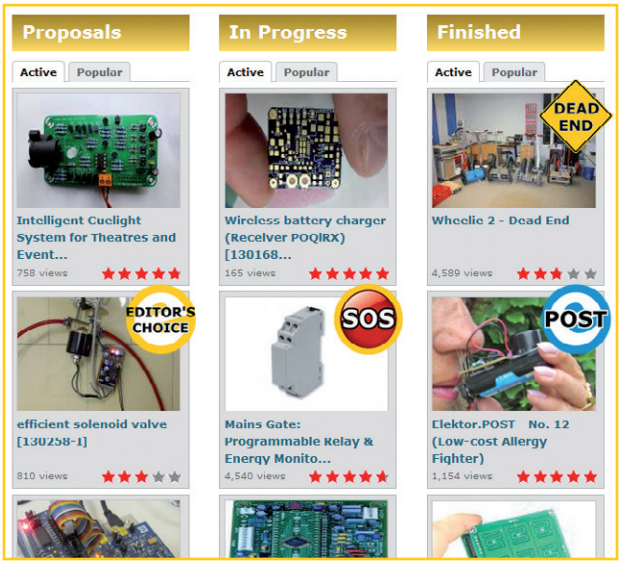
elektor

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

Tipps & Tricks

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

Der Sommer ist vorbei, Ihre Projekte sind abgeschlossen. Nun ist es Zeit, sie auf Elektor.Labs zu dokumentieren. Hier einige Tipps und Tricks, um mehr aus Ihrer Online-Publikation zu machen.



Elektor.Labs schmückt sich mit neuen Icons.

Icons

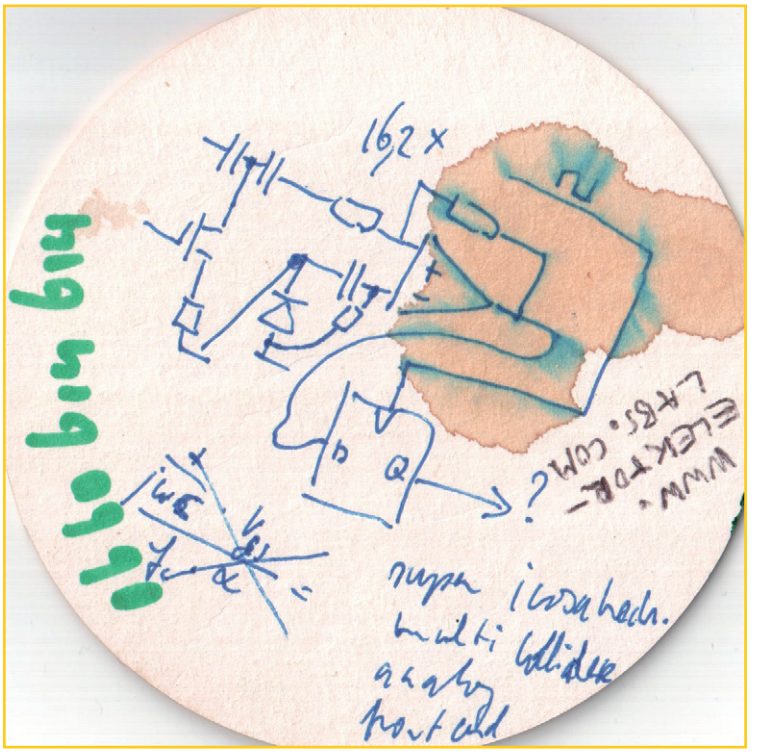
Die Elektor.Labs-Webseite wird durch neue Features permanent verbessert. Das ist ein evolutionärer Prozess und neue Funktionen kommen hinzu, wenn wir merken, dass wir sie brauchen. Eine der neuesten Ergänzungen sind die Icons, die Sie bestimmt schon auf der Webseite bemerkt haben. Momentan gibt es vier Icons, welche die Aufmerksamkeit auf Projekte lenken. Zwei davon: das blaue Post-Icon und das gelbe Icon „Editor's choice" werden von Elektor vergeben. „Dead-end" und „SOS" können von allen Postern eingesetzt werden. Man kann damit zeigen, dass man gerne Unterstützung hätte oder dass das Projekt in einer Sackgasse steckt.

Beachten Sie bitte, dass Projekte mit dem „Dead-end"-Icon in die Spalte „Finished" verschoben werden können, weshalb man damit vorsichtig sein sollte.

So aktiviert man ein Icon für ein Projekt.

Projekt-Sichtbarkeit

Auch wenn Illustrationen für den Projekt-Header optional sind, ist es doch empfehlenswert, dafür ein Foto oder eine Zeichnung hochzuladen. Schön illustrierte Projekte erhalten gemeinhin eine höhere Priorität und landen somit weiter oben in der Liste. Bei jedem Update Ihres Projekts (beim Klick auf den Save-Knopf) wird es ganz oben in die Liste einsortiert - *wenn es ein Projekt-Header-Foto hat*. Das Default-Bild eines Projekts ist der nette Bierdeckel mit einer Skizze drauf.



Ersetzen Sie das Default-Projekt-Header-Foto durch ein schönes „richtiges“ Foto Ihres Projekts, und verbessern Sie so dessen Sichtbarkeit.

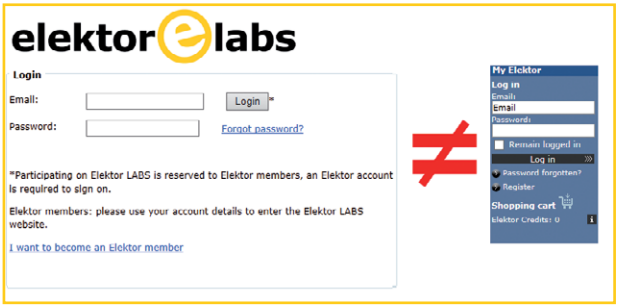
www.elektor-labs.com

Aktive User

Es gibt sehr aktive User auf Elektor.Labs und wir finden das toll. Deshalb wurde ein Scoring-Mechanismus eingebaut, der aktive User auszeichnet. Hin und wieder haben wir nämlich Hardware, Bücher oder andere Goodies aus dem Elektor-Bestand abzugeben und aktive User stehen natürlich an erster Stelle bei den Empfängern dieser Geschenke. Das Scoring funktioniert einfach: Das Einstellen eines neuen Projekts bringt vier Punkte, ein Beitrag ist zwei Punkte wert und jeder Kommentar wird mit einem Punkt honoriert. Nun muss man die Scores natürlich auch noch sehen können. Daran arbeiten wir gerade...
P.S. Das Scoring ist nicht komplett automatisiert. Es sind auch Menschen involviert. Von daher bringt es nichts, unsinnige Projekte einzustellen oder entsprechende Beiträge oder Kommentare zu verfassen. Das wird als Spam betrachtet. Und Moderatoren verhindern, dass es da jemand zu weit treibt.

Passwörter und E-Mail-Adressen

Ein schwieriges Thema - und deshalb bekommen wir immer wieder Fragen dazu. Die wichtigste Info ist, dass man es als Elektor-Mitglied aus historischen Gründen mit zwei Login-Systemen zu tun hat. Wir arbeiten zwar schon an einer Vereinheitlichung, doch das braucht leider noch etwas Zeit. Ihr Account für Elektor.de, Elektor.Post und Elektor.Store ist nicht der gleiche wie der für Elektor.Labs und Elektor.Magazine. Man kann sich die Sache selbstverständlich auch einfach machen und für beide Systeme die gleichen Namen und Passwörter verwenden, aber man muss das nicht.
Wenn Sie Ihre E-Mail-Adresse ändern wollen, dann kontaktieren Sie uns und schicken Sie uns Ihre alte und die neue E-Mail-Adresse an service@elektor.de oder an labs@elektor.com.



Es gibt zwei Accounts und zwei getrennte Login-Systeme.

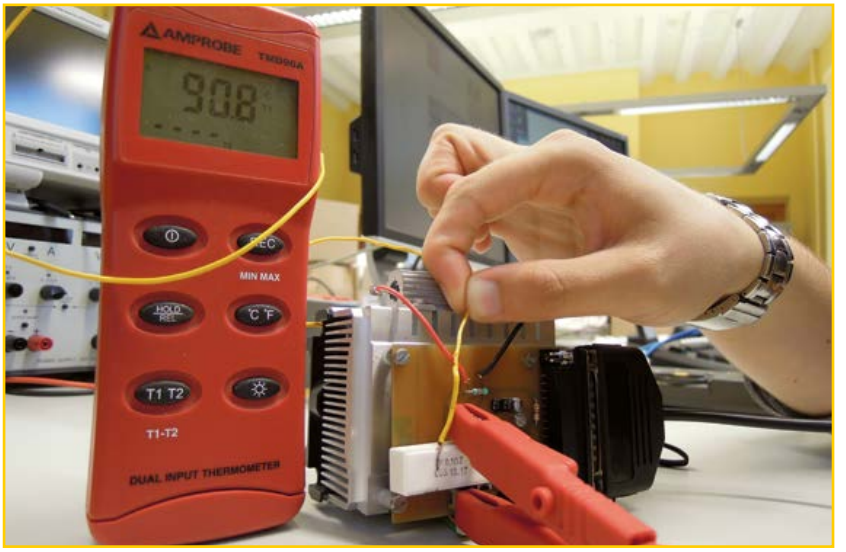
90° und steigend

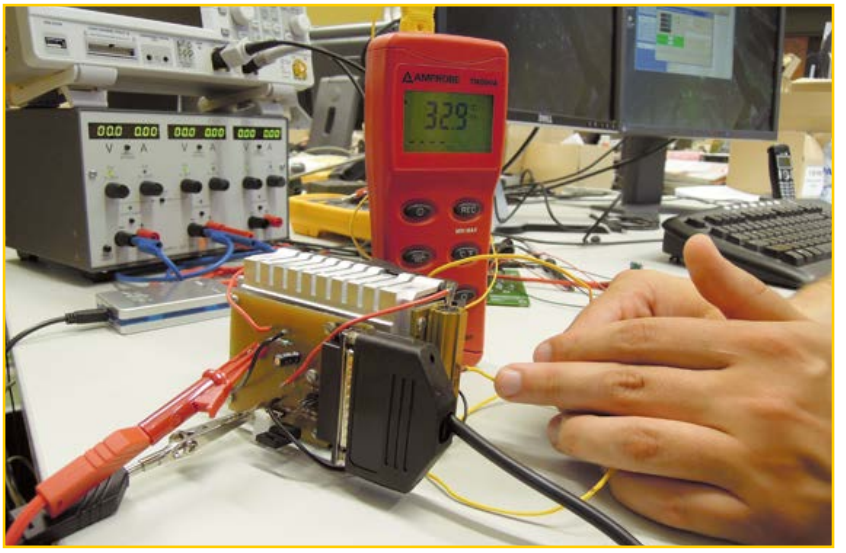
Von **Thijs Beckers**
(Elektor-Redaktion)

Diesmal waren nicht (nur) die sommerlichen Temperaturen schuld. Beim Ausprobieren eines Batterietesters, der bald in Elektor veröffentlicht werden soll, war Kollege Tim Uiterwijk aus dem Elektor-Labor überrascht von der Temperatur eines 7-W-Serienwiederstands (der große weiße im Bild, an den ein Temperatursensor gehalten wird), denn er war gut über 90 °C heiß!
Das hatte er nicht erwartet, denn die berechnete Verlustleistung (P_{diss}) lag deutlich unter den 7 W, die dieser Drahtwiderstand laut Datenblatt vertragen konnte. Die Verlustleistung berechnet sich ja bekanntlich zu:

$P_{diss} = I^2 \times R$.

Laut Formel wäre der Maximalstrom durch den 0,1-Ω-Widerstand also 8 A. Während dieses Tests aber wurde der Strom auf 4,5 A beschränkt. Mit I = 4,5 A und R = 0,1 Ω ergeben sich (4,5 A)^2 × 0,1 Ω = 2 W. Dafür wurde dieser große Widerstand aber sehr schnell sehr heiß. Zu heiß, denn unsere Labor-Daumenregel besagt, dass Bauteile außen möglichst nicht über 80 °C warm werden sollen. Das Problem wartete also auf eine Lösung.





Wie man auf den Bildern sehen kann, kam ein Kühlkörper zum Einsatz. Dabei handelt es sich um einen gewöhnlichen CPU-Kühlkörper (mit Lüfter), der für einen Intel Pentium 4 gedacht war. Wegen des stetigen Luftstroms haben diese Art Kühlkörper einen besonders niedrigen Wärmewiderstand in der Größenordnung von 0,4 K/W. Sie eignen sich also prima dafür, große Wärmemengen abzuführen. Man denke nur an die alten Pentium-CPUs, die es locker auf bis zu 125 W an Verlustwärme brachten und trotzdem ausreichend gekühlt wurden. Es ist zwar bereits ein Shunt-Widerstand auf dem Kühlkörper montiert (oben im Bild), doch reicht es noch locker für einen zweiten.
Tim tauschte zunächst den keramischen Lastwiderstand durch einen Drahtwiderstand im Alu-Gehäuse aus, der für 50 W Leistung ausgelegt war. Beide Lastwiderstände montierte er an den Seiten des Kühlkörpers an den Stellen, wo der Luftstrom maximal war (siehe Foto unten). Diese Lösung funktionierte gut, denn der zuvor heiße Widerstand erreichte nur noch 33 °C. Mit einem durch die Software begrenzten Maximalstrom von 8 A erwärmte sich der Widerstand nur noch auf 50 °C – die „Nicht-heißer-als-80°-Regel“ war erfüllt. Es war sogar noch etwas Reserve vorhanden. Theoretisch sollte der 50-W-Widerstand 22 A und Temperaturen bis zu 250 °C aushalten können, doch diese Extreme werden in der Schaltung bei weitem nicht erreicht.
Problem gelöst. Nun bleibt Ihnen nur noch, auf das Erscheinen des Artikels mit der kompletten Schaltung zu warten. Bald ist es soweit.

(130055)

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

NEU

Ein Muss für jeden Elektor-Leser!

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

elektor

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

Bewegungsgesteuerter Fernauslöser mit Arduino

Von **Rolf Blijleven** (NL)

Ein preiswertes Nachtlicht mit Bewegungssensor, eine Infrarot-LED, nur wenige passive Bauelemente und dazu ein passend programmiertes Arduino-Board: Daraus entsteht ein System, das bei Bewegungen eines Objekts drahtlos eine Fotokamera auslöst.

Arduino erinnert mich an Lego: Der spielerische Umgang ist lehrreich und spannend, und nicht selten entsteht etwas Sinnvolles und Nützliches. Eine Weile freue mich über das Erschaffene, dann nehme ich es auseinander und beginne ein neues Projekt. Für meine Spiegelreflexkamera D80 von Nikon hatte ich mit Arduino einen drahtlosen Fernauslöser konstruiert. Einen solchen Fernauslöser hätte ich als Zubehör im Fotohandel erstehen können, doch Arduino macht nicht nur Spaß, auch zusätzliche Optionen sind umsetzbar. Außer dem Arduino-Board benötigte ich eine Infrarot-LED, einen Widerstand und ein passendes Arduino-Programm. Das Programm fand ich im Internet.
Nachdem der Fernauslöser realisiert war, hatte ich die Idee, einen Bewegungssensor vorzuschalten. Die käuflichen Typen sind leider fast ausnahmslos für den Anschluss an das Stromnetz konzipiert. Dem stand entgegen, dass ich meinen bewegungsgesteuerten Fernauslöser unbedingt schnurlos betreiben wollte. In einem Elektromarkt entdeckte ich ein batteriebetriebenes Nachtlicht, in dem ein Bewegungssensor bereits eingebaut war (**Bild 1**). Das Preisschild wies aus, dass dieses Nachtlicht nur 2,65 Euro kosten sollte. Ein wahres Schnäppchen!

Das Nachtlicht

Vom Innenleben meiner Errungenschaft war ich überrascht: Kein auf die Platine gepappter Chip und keine SMDs, stattdessen ein IC mit Beinchen, einige konventionelle Widerstände und Konden-

satoren, ein PIR-Sensor und eine Fotodiode.
Auf dem IC war „TL0001“ aufgedruckt, das sagte mir erst einmal nichts. Also war Google gefragt, und siehe da, ein Datenblatt existierte, allerdings in chinesischer Sprache. Dank Google war auch die Sprache keine unüberwindliche Hürde. Der chinesische Text wurde einfach in die Zwischenablage übernommen und anschließend von Google Translate übersetzt. Ein Gewinn für die niederländische Sprache war die Übersetzung zwar nicht, doch immerhin ließ sie erahnen, worum es ging. Ein Schaltbeispiel war dabei, das mit meinem Nachtlicht nicht ganz übereinstimmte, aber die Verwandtschaft war unverkennbar.
Das Nachtlicht hatte drei Eigenschaften, die ich nicht brauchen konnte: Funktionsfähig war es nur im Dunkeln, die Einschaltperiode war mehrere Minuten lang (der Fernauslöser schaltete mit einem kurzen Impuls), und wenn der Bewegungssensor ein Objekt erkannte, traten drei lichtstarke LEDs in Aktion.
Das Aufleuchten der LEDs ließ sich schnell abstellen. Die LED-Ströme flossen über einen gemeinsamen Strombegrenzungswiderstand. Zwei LEDs habe ich demontiert, und der Widerstand erhielt einen höheren Wert (2,2 kΩ, „A“ in **Bild 2**). Die dritte LED dient jetzt zur Funktionskontrolle.
Die nächste Aktion galt der unerwünschten Deaktivierung bei Tageslicht. Im Schaltbeispiel (**Bild 3**) ist R3 ein LDR. Statt eines LDR fand ich eine Fotodiode vor, ein Bauelement, das ebenfalls mit zunehmend heller Beleuchtung seinen Widerstand verringert. Die Fotodiode tauschte ich gegen einen hochohmigen Widerstand aus (220 kΩ, „B“ in Bild 2). Das hatte Erfolg, jetzt reagierte das Nachtlicht auch in hellster Umge-

Bild 1.
Es gibt sie preiswert in Baumärkten und Tiefpreisläden: Nachtlichter mit eingebautem Bewegungsmelder.

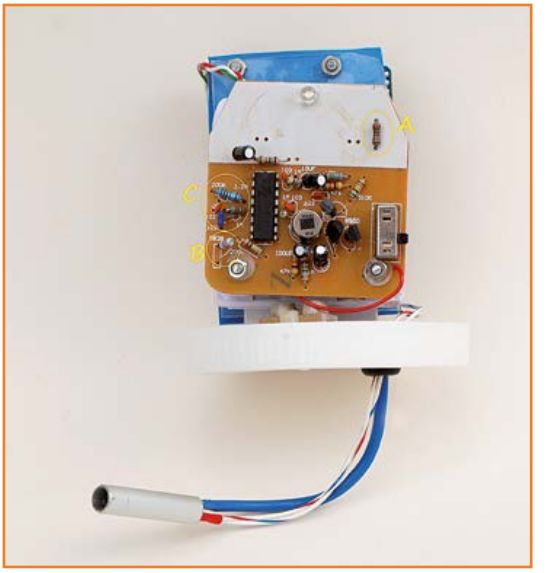


Bild 2.
Das Innenleben des Nachtlichts, hier mit den Modifikationen.

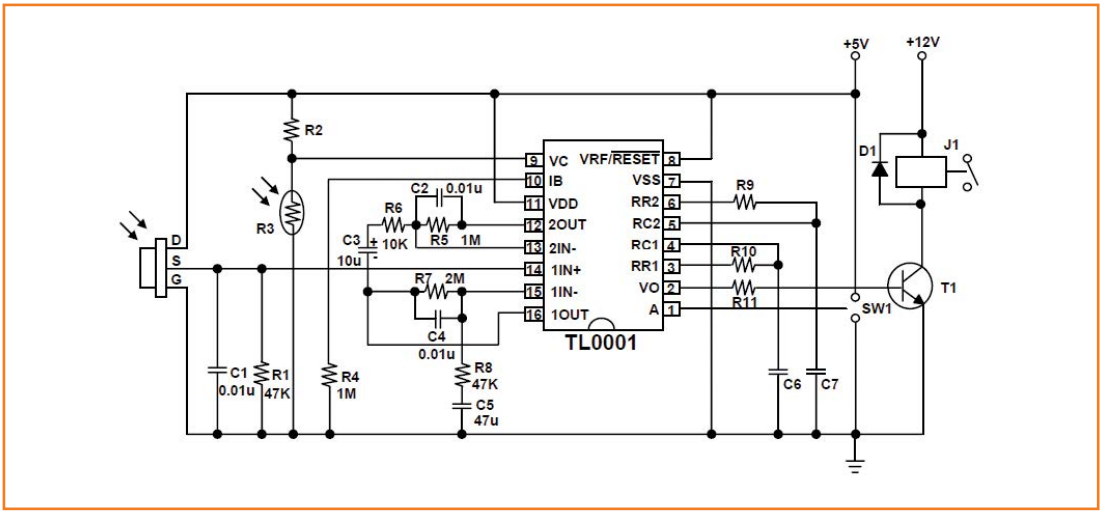


Bild 3.
Schaltbeispiel für den TL0001 vom chinesischen Chip-Spezialisten Treasure Link Technology. Die Steuerung des Nachtlichts ist ähnlich, nur in einigen Details weicht sie ab.

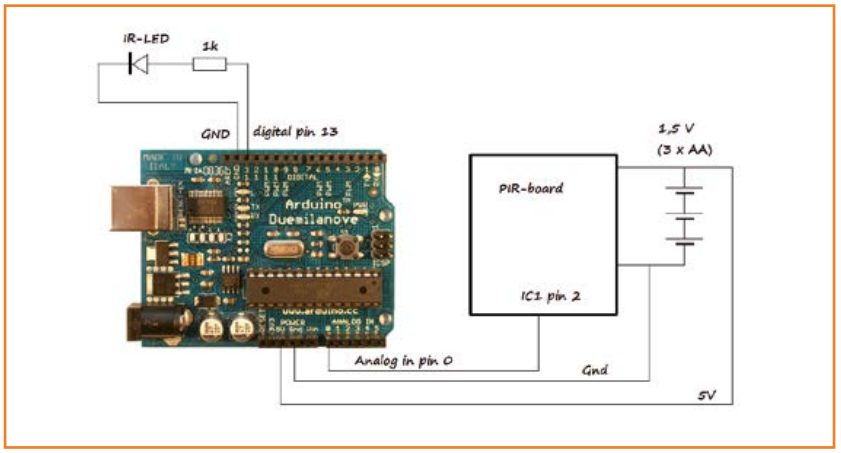


Bild 4.
So werden der Arduino, die Infrarot-LED und das PIR-Board (aus dem Nachtlicht) miteinander verbunden.

bung auf bewegte Objekte.
Nun musste ich nur noch die Einschaltdauer verkürzen. Ich vermutete, dass die Zeit von einer RC-Kombination abhing, doch welches R und welches C waren zuständig? Mir war aufgefallen, dass sich das Nachtlicht nach dem Abschalten nicht unmittelbar wieder einschalten ließ. Folglich musste eine zweite RC-Kombination vorhanden sein, die für eine kurze Einschaltverzögerung sorgte. Das Datenblatt war gefragt, und tatsächlich, darin stand, von Google maschinell aus dem Chinesischen übersetzt:
„Tx-Ausgang Zeit Verzögerung von Außenseite nach Größe R9 und C7, mit Wert Tx ≈ 24576xR9C7 anpassen; verursacht durch externes Blockieren Zeit Ti. R10 und C6 von Umfang von Korrektur Wert von Ti ≈ 24xR10C6." Deutlicher ging es eigentlich kaum noch, oder was meinen Sie? Jetzt war Reverse Engineering angesagt: Nicht eine Platine für ein Schaltbild entwerfen, sondern umgekehrt, ein Schaltbild anhand einer Platine zeichnen. Das fällt leichter, wenn die Layouts beider Platinenseiten nebeneinander liegen. Eine Fotokopie der Kupferseite leistete die gewünschte Hilfestellung.
Mit etwas Zeichen-, Rechen- und Lötarbeit hatte ich R9/C7 und R10/C6 identifiziert und durch „bessere" Werte ersetzt. Doch das erwies sich als Trugschluss, das Nachtlicht versagte nun vollständig den Dienst. Während ich versuchte, die Ursache zu ergründen, fiel mein Blick auf einen Satz im Datenblatt: *„BISS0001 Chip ist voll kompatibel mit Chip."* Das Datenblatt des BISS0001 brachte Licht ins Dunkel, denn dort waren die Formeln anders geschrieben: *Tx ≈ 24576xR10C6* und *Ti ≈ 24xR9C7*. Im Datenblatt des TL0001 waren die Kombinationen R10/C6 und R9/C7 miteinander vertauscht! Mit R10 · C6 = 1 kΩ · 100 nF und R9 · R7 = 270 kΩ · 1 nF ergaben sich für Tx ≈ 24 ms und für Ti ≈ 0,5 s („C" in Bild 2). Kondensator C7 konnte an seinem Platz bleiben.
Der Versuch, die Empfindlichkeit des PIR-Sensors zu steigern, hatte nur mäßigen Erfolg. Der Verstärker im TL0001 bestand offensichtlich aus zwei Stufen. Die Verstärkung des internen Opamps mit den Anschlüssen 1IN+, 1IN- und 1OUT (siehe Datenblatt) beträgt ungefähr R7/R8. Mit 1 MΩ und 12 kΩ liegt sie bei 84, während sie mit den ursprünglichen Werten 2 MΩ und 47 kΩ nur bei rund 40 lag. An der Empfindlichkeit des PIR-Sensors ändert dies nichts, es werden lediglich kleine Signale höher verstärkt, während große Signale an die Betriebsspannung stoßen. Die zweite Stufe, ebenfalls ein Opamp, hat die Verstärkung R6/R5. Ursprünglich betrug sie 100, nun lag sie mit R6 = 15 kΩ bei 67. Damit wurden auch große Signale nicht mehr begrenzt.
Nach dieser Modifikation erkannte der Sensor auch etwas weiter entfernte Bewegungen, doch das war nur bedingt ein Gewinn. In ihrem Erfassungsbereich reagieren PIR-Sensoren auf Änderungen von Wärmedifferenzen. Eine Katze, die im Winter an einer vereisten Mauer entlang streift, wird zuverlässiger erkannt als die gleiche Katze, die sich an einem tropisch heißen Sommertag vor der gleichen Mauer bewegt.
Es ist nicht unwahrscheinlich, dass an anderen Orten andere Versionen von Nachtlichtern mit eingebautem Bewegungssensor im Handel sind. Auch dann, so hoffe ich, wird meine Version eine Orientierungshilfe für die Modifikation der Schaltung und den Einsatz zusammen mit dem Fernauslöser sein.

Arbeitspferd Arduino

Mein zum PIR-Board umfunktioniertes Nachtlicht habe ich an einen analogen Eingang des Arduino angeschlossen. Ein digitaler Ausgang steuert über einen strombegrenzenden Widerstand eine Infrarot-LED, ihre Signale lösen die Kamera aus. Die drei Batterien vom Typ AA versorgen auch das Arduino-Board. Aus **Bild 4** geht hervor, wie das System zusammengeschaltet ist.
Das Timing des Infrarot-Signals habe ich den Quellen [2] und [3] entnommen. Bewegte Objekte in einigen Metern Entfernung vom PIR-Sensor lösen das Infrarot-Signal aus, das seinerseits ebenfalls einige Meter bis zum Kamerastandort überbrückt. Auch Fensterscheiben durchdringt das Infrarot-Signal, so dass beispielsweise die Kamera im Haus stehen kann, während sich der

elektor ACADEMY
the school of electronics

3-tägiges Seminar 'Embedded Linux in Theorie und Praxis – ein Crashkurs'

Der Einstieg in ein so mächtiges Werkzeug wie Linux ist nicht trivial! Ziel des Kurses ist es, Ihnen grundlegende Embedded-Linux-Konzepte sowie die Handhabung von Linux zu vermitteln. Was sind z. B. Vor- und Nachteile? Sie werden Ihren eigenen Bootloader und Kernel cross-kompilieren, diverse Programme auf einem PC erstellen/cross-kompilieren und auf einem eingebetteten System ausführen und debuggen. Eine Kombination aus Theorie und praktischen Übungen wird es Ihnen ermöglichen, das neu erworbene Wissen bei Eigenentwicklungen einzusetzen. Nach dem Kurs sind Sie wahrscheinlich noch kein Embedded-Linux-Experte, aber hoffentlich in der Lage sein, sich selbständig zurechtzufinden.

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

Eagle PCB und Design

In diesem Kurs werden Sie lernen, wie man mit dem Programm Eagle der Firma Cadsoft GmbH Leiterplatten entflechten kann. Begonnen wird mit dem Zeichnen von Schaltplänen unter Verwendung von Standard-Eagle-Bibliotheken. Sie lernen, wie man Schaltpläne über mehrere Seiten hinweg zeichnet und wie man eigene Bibliotheken und Bauteile erstellt. Anschließend werden Sie lernen, wie man aus dem Schaltplan eine Leiterplatte definiert und diese dann entflechtet (layoutet). Dabei werden auch Spezialkenntnisse zum Layouten von HF-Schaltungen, wie Leitungswellenwiderstand, vermittelt. Nach erfolgreichem Layout werden Produktionsdaten erzeugt, die man benötigt, wenn man die Platine fertigen lassen möchte. Dabei wird auch auf die verschiedenen Produktionsarten wie fräsen und ätzen eingegangen. Zum Abschluss gibt es Tipps und Tricks zum Umgang mit EAGLE.

Referent: Prof. Dr.-Ing. Francesco P. Volpe - Teilnahmegebühr: € 449,00

50% Sonderrabatt für Elektor-Abomitglieder: 224,50 € (inkl.MwSt)

Altium – Leiterplattenentwicklung: Bibliotheken intelligent aufbauen

Im Seminar werden verschiedene Aspekte erarbeitet, die bei der Anlage von Bibliotheken zu beachten sind. Das fängt ganz klassisch bei der Struktur an, die besagt welche Komponenten wo abgelegt werden, damit der Entwickler später eine Chance hat, die Bauteile zu finden. Aber auch die Belange der Fertigung und des Einkaufs werden in einer guten Datensammlung berücksichtigt. Heute sind diese sicher keine Optionen, sondern feste Anforderungen. Nicht zuletzt werden aber auch die Möglichkeiten eines aktuellen EDA Tools beleuchtet. Hier geht es sowohl um generelle Bibliothekskonzepte, aber auch um die Realisierungsmöglichkeiten.

Referent: Dipl. Ing. Christian Keller Teilnahmegebühr: € 299,00 (inkl. MwSt).

50% Sonderrabatt für Elektor-Abomitglieder: 149,50 € (inkl.MwSt)

elektor ACADEMY
the school of electronics

Workshops * Seminare * Kurse * Weiterbildungen

Top-Fachleute aus der Branche referieren über ein faszinierendes Thema!

Altium – Leiterplattenentwicklung: Bibliotheken intelligent aufbauen
Hanau 21.11.2013

Arduino-Programmierung und Projektentwicklung
Dortmund: 05.09.2013
Zürich (CH): 07.09.2013

Kommerzielle Verwendung von Open-Source-Software
München: 12.09.2013
Dortmund: 19.09.2013

Rapid SMD-Prototyping mit PCB-POOL und TARGET 3001!
München: 11.09.2013

Embedded Linux
Hanau: 16.09. bis 18.09.2013
München: 09.12. bis 11.12.2013

Linux Debugging
Dortmund: 19.09. + 20.09.2013
München: 12.12. + 13.12.2013

Eagle PCB und Design
München: 12.09.2013
Braunau am Inn: 13.09.2013
Dortmund: 19.09.2013
Hanau: 17.10.2013

Weitere Infos & Anmeldung: www.elektor.de/events

Fernauslöser im Freien befindet.
Meine Kamera, eine D80 von Nikon, hat noch folgende Eigenschaft: Wenn die D80 im „IR remote mode“ arbeitet, wartet sie eine bestimmte Zeit auf ein Infrarot-Signal. Bleibt das Signal aus, kehrt sie selbständig in den Normalbetrieb zurück. Für meine Ambition, die Naturfotografie, ist dieses Verhalten recht hinderlich. An der Kamera lässt sich das Warteintervall auf 15 Minuten einstellen. Deshalb generiert die Arduino-Firmware nach etwa 14 Minuten ein Infrarot-Signal, das die Kamera gewissermaßen wach hält. Die Kamera ist damit tagelang schussbereit, auf Wunsch so lange, bis beispielsweise ein exotisches Lebewesen des Weges kommt.
Die Länge des Aktivierungsintervalls ist anpassbar, so dass ohne PIR-Board auch Zeitraffer-Clips von wachsenden oder aufblühenden Pflanzen gedreht werden können. Für das Aktivierungsintervall ist Timer 1 zuständig, ein 16-bit-Timer, der von 0 bis 65536 zählt. Wenn der Timer mit dem `timerPreload` = 3036 beginnt, zählt er 65536 - 3036 = 62500 Zähltakte und generiert dann einen Interrupt. Der Arduino Duemillenove ist mit 16 MHz getaktet, ein Vorteiler durch 1024 leitet daraus die Frequenz 15625 Hz ab. Die Timer-Interrupts ereignen sich folglich nach 62500 / 15625 s = 4 s. Im Programmcode sieht dies wie folgt aus:

```
#define four_sec 1
#define twelve_sec 3 * four_sec
#define minute 5 * twelve_sec
#define quarter 14 * minute
```

Ein `quarter` ist hier auf 14 Minuten festgelegt, die Wartezeit 15 Minuten war für meine Kamera etwas zu lang. Mit `timeCounter` wird die Zeit hochgezählt. Die Interruptroutine setzt den Startwert `timerPreload` auf 3036 und inkrementiert den `timeCounter`. Der Zählerstand multipliziert mit 4 ist die abgelaufene Zeit in Sekunden:

```
ISR(TIMER1_OVF_vect) {
  TCNT1 = timerPreload;
  timeCounter +=1;
}
```

Im Hauptzweig des Programms wird das Infrarot-Signal gesendet, das die Kamera auslöst. Das geschieht entweder, wenn das PIR-Board ein bewegtes Objekt erkannt hat oder ein `quarter` (14 Minuten) verstrichen ist:

```
if (val > 200 || timeCounter == quarter ) {
      timeCounter = 0;
      takePicture();
      delay(500);
}
```

Der Quellcode der Firmware steht auf der Elektor-Website [5] zum freien Download bereit. Da der Umfang der binären Datei nur etwa 4 KB beträgt, ist in einem Arduino mit 32 KB Flashspeicher noch viel Platz für Erweiterungen frei. Wie wäre es beispielsweise mit einem drahtlosen Fernauslöser, der auf *akustische* Ereignisse reagiert?

(130265)gd

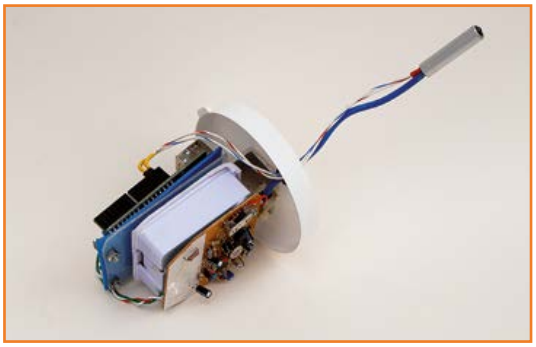


Bild 5.
Arduino und PIR-Board mit Batteriegehäuse, Rückseite an Rückseite auf einer isolierenden Profilplatte montiert.

Bild 6.
Ein spritzwasserdichtes Kunststoffgehäuse schützt die Konstruktion vor Staub und Nässe.

Weblinks

[1] www.treaslink.com/UploadFiles/2010531152721141.pdf
[2] www.e-ele.net/DataSheet/BISS0001.pdf
[3] www.bigmike.it/ircontrol
[4] http://luckylarry.co.uk/arduino-projects/arduino-ir-remote
[5] www.elektor.de/130265

ECD 7

Bauteilbibliothek mit über 75.000 Komponenten

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

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

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

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

elektor

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

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: 34,90 €
zzgl. Ust./Versand

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

elektor PCB Service

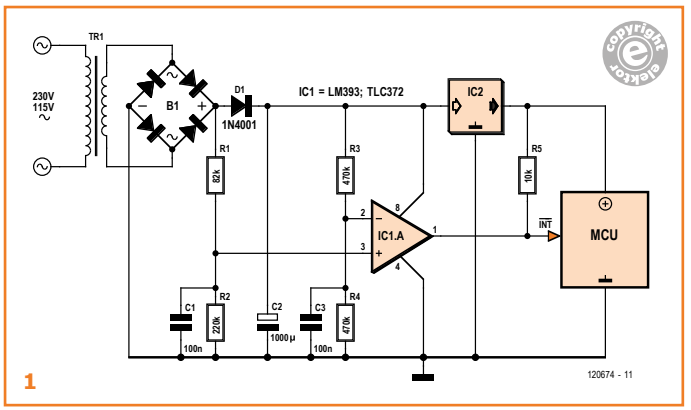
powered by Eurocircuits

25% Rabatt auf alle neuen Elektor-Platinen!

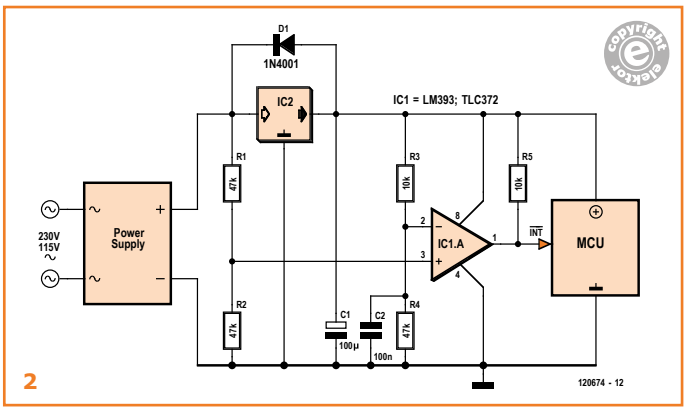
Der Elektor-PCB-Service gewährt auf jede neue Elektor-Platine (ab Erscheinungsdatum) 90 Tage lang 25% Rabatt!

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

Schnell noch speichern 2.0



1



2

Von **Jürgen Werner** (D)

Beim Ausschalten oder sonstigen Spannungsausfällen ist es bei manchen Schaltungen mit Mikrocontroller sinnvoll, den aktuellen Status oder bestimmte Werte in dessen EEPROM abzulegen, damit diese Informationen beim erneuten Einschalten automatisch wieder zur Verfügung stehen. Zu diesem Zweck wurde schon eine von Rainer Reusch entwickelte Schaltung (siehe **Bild 1**) als Entwicklungstipp in Elektor veröffentlicht [1]. Das Prinzip dahinter ist, dass die Spannung vor D1 mit der durch C2 gepufferten über einen Komparator verglichen wird, der dann ein Signal an den Mikrocontroller liefert, wenn die Eingangsspannung ausfällt. Dank D1 und C2 fällt die Spannung beim Ausschalten am nichtinvertierenden Eingang von IC1.A schneller ab als am invertierenden Eingang. Die Folge ist ein „Low"-Signal am Komparator-Ausgang. Solange daher noch genügend Energie im Puffer-Elko zur Verfügung steht, hat der Mikrocontroller noch Zeit, per Interrupt-Routine die notwendigen Daten zu retten.

Dies funktioniert soweit ganz gut. Jedenfalls in einfachen Situationen. Ein Problem allerdings ist, dass das Beschreiben von EEPROM-Zellen durchaus einige Millisekunden dauert. C2 muss aber größer als nötig dimensioniert werden, da er auch noch den Ruhestrom des Spannungsreglers zu liefern hat. Außerdem ist die Dimensionierung von zum Beispiel C1 aufgrund der hier vorhandenen Welligkeit nicht ganz so einfach. Noch problematischer wird die Sache, wenn man statt eines Trafos mit Brückengleichrichter so ein schönes modernes Steckernetzteil verwendet, das selbst schon stabilisiert oder gar ein Schaltnetzteil ist. Denn dann funktioniert die Sache gar nicht, da die Spannung an R1 wegen des im Netzteil integrierten Puffer-Elkos nicht schnell genug abfällt. Dies veranlasste Jürgen Werner zum Tüfteln; als Resultat präsentiert er mit der Schaltung von **Bild 2** eine verbesserte und sogar noch einfachere Lösung. Der Komparator ist hinter den Spannungsregler verschoben. Verglichen wird nur noch die Eingangsspannung mit der Ausgangsspannung des Spannungsreglers. Eine Diode in Serie mit dem Eingang des Spannungsreglers entfällt. Der Puffer-Elko C1 kann jetzt kleiner ausfallen. Das Beste aber ist, dass die Schaltung nicht mehr davon abhängig ist, wie schnell die Eingangsspannung sinkt. Schaltet man nämlich das Netzteil aus, dann sinkt die Eingangsspannung, während die Ausgangsspannung noch eine Weile durch den Spannungsregler konstant gehalten wird. Bei korrekt dimensionierten Spannungsteilern ist immer irgendwann die Spannung am nichtinvertierenden Eingang des Komparators kleiner als am invertierenden und schon entsteht das „Low"-Signal, das am Mikrocontroller einen Interrupt auslösen kann. Die Schaltung ist für ein Netzteil mit 9-V-Ausgangsspannung und einen Spannungsregler für 5 V dimensioniert. D1 schützt den Spannungsregler vor Stromfluss in der verkehrten Richtung. Mit 100 µF für C1 und einer Last von 5 mA hat der Mikrocontroller noch mindestens 17 ms Zeit, wichtige Zustände im EEPROM abzulegen. Ein flankengetriggerter Interrupt ist hier vorzuziehen. Wenn man stromverbrauchende Controller-Einheiten wie den ADC etc. abschaltet, bleibt noch mehr Zeit zum Speichern.

(120674)

[1] Schnell noch Speichern, Elektor Februar 2012, www.elektor.de/080379

elektorLive! 2013

Seminar- & Ausstellungstag für die Elektronik-Entwicklung und -Anwendung

Am 12. Oktober 2013 findet im Congress Park Hanau die zweite Elektor*Live!*-Veranstaltung statt.

An diesem Tag halten kompetente Elektor-Autoren, -Entwickler und Experten aus der Elektronik-Branche verschiedene Seminare zu populären Elektronik-Themen ab. Geplant sind 3 Seminar-Runden mit bis zu 4 parallel stattfindenden Seminaren.

Begleitet werden die Seminare von einer Ausstellung renommierter Elektronik-Unternehmen, die ihre Produkte und Innovationen vor Ort präsentieren werden. Während der Pausen haben Sie die Möglichkeit, mit diesen Firmen in Kontakt zu treten.

Freuen Sie sich jetzt schon auf interessante Seminare und Workshops mit hochkarätigen Referenten!

Geplante Seminare:
- Einführung in LabVIEW
- Elektronik-Apps mit Android
- Raspberry Pi-Anwendungen
- Röhrenverstärker (mit Menno van der Veen)
- Einstieg in die AVR-Programmierung
- Layout-Design mit EAGLE 6
- ARM Cortex-M3 Mikrocontroller
- Entwicklung eines FPGA-Erweiterungsboards
- Praxisprojekte mit Arduino
- Rapid SMD-Prototyping

Datum:
Samstag, 12. Oktober 2013

Zeit:
09:00 Uhr – 17:00 Uhr

Ort:
Congress Park Hanau

Eintritt:
29,50 € für Elektor-Mitglieder
49,50 € für Nicht-Mitglieder
19,50 € für Schüler/Studenten
Der Eintritt berechtigt zur Teilnahme an 2 Seminaren.

Tagesablauf:
09:00 Uhr – 17:00 Uhr: Ausstellung
09:30 Uhr – 11:30 Uhr: 1. Seminar-Runde
12:30 Uhr – 14:00 Uhr: 2. Seminar-Runde
15:00 Uhr – 16:30 Uhr: 3. Seminar-Runde

Weitere Infos & Anmeldung unter
www.elektor-live.de

Programmierung mit Mono

Für Windows, Linux, Mac OS, Android

Elektroniker müssen zunehmend auch Programme für den PC erstellen. Häufig will oder darf man sich dabei nicht auf ein Betriebssystem festlegen. Mit dem Open-Source-Framework „Mono“ lassen sich Anwendungen für Windows, Linux, Mac OS und sogar für Android und das iPhone entwickeln. Dieser Artikel gibt einen Überblick und zeigt die Programmierung unter Windows und Linux an einem Beispiel.

Von
Dr. Veikko Krypczyk

Der Traum jedes Softwareentwicklers ist es, seine Anwendung nur einmal zu konzipieren, das Design für die Oberfläche einmalig zu entwerfen und dann das fertige Produkt auf allen gängigen Betriebssystemen zur Verfügung zu stellen. Auch für den Elektronikentwickler, der für seine Hardware eine PC-Steuerung vorsieht, ist es von großem Vorteil, wenn das Steuerprogramm sowohl auf Rechnern mit dem Betriebssystem Microsoft Windows, aber auch auf anderen Systemen wie Linux oder Mac OS einsatzfähig ist.

Warum Mono?

Eines gleich vorweg: Eine ideale Variante einer plattformübergreifenden Entwicklung wird es auch in Zukunft nicht geben. Generalisierung geht zu Lasten der Spezialisierung! Es gilt, das richtige Mittelmaß für den individuellen Anwendungszweck zu finden und dieses mit den dafür am besten geeigneten Werkzeugen umzusetzen. Das Framework „Mono“ ist weitgehend kompatibel zum .NET-Framework von Microsoft, aber für weitaus mehr Betriebssysteme einsetzbar. Ein gutes Argument für die Verwendung von Mono ist die (zunehmende) Popularität von C#, respektive der gesamten .NET-Plattform. Besondere Vorteile von .NET sind eine sehr umfassende Klassenbibliothek, die Möglichkeit der freien Wahl der Programmiersprache (C#, Visual Basic, Delphi,…) und eine effektive Entwicklungsumgebung. Diese Vorteile der Programmentwicklung auch auf andere Betriebssysteme zu übertragen, ist das Hauptmotiv für die Entwicklung und Nutzung von Mono. Darüber hinaus ist das Projekt in vielen Bestandteilen open-source, damit möchte man die Abhängigkeit von Microsoft verringern.
Das Mono-Projekt existiert nun schon einige Zeit, unter [1] bekommt einen Überblick zur Historie. Mono umfasst die folgenden Teilprojekte [2]:

- Mono: Hierbei handelt es sich um das Kernelement. Als technisches Vorbild gilt das .NET-Framework. Die Verwendung von Mono für Windows und Linux ist kostenfrei.
- MonoTouch for iOS: Mittels dieses Projektes ist es möglich, iPhone und iPad-Apps auf Basis der Programmiersprache C# zu entwickeln. Bei der Verwendung fallen Lizenzgebühren an. Lediglich das Testen mit Hilfe der Free-Edition (Emulator) ist kostenfrei. Darüber hinaus existieren eine Indie- (299 Dollar), eine Business- (999 Dollar) und eine Enterprise- (1.899 Dollar) Edition.
- Mono for Android: Dient der Entwicklung von Apps für die Android-Plattform. Hinsichtlich der Editionen und der Preisgestaltung gilt das gleiche wie bei MonoTouch for iOS.

Einstieg

Um Erfolge mit Mono zu verbuchen, sind Vorkenntnisse notwendig: Ein sicherer Umgang mit den Betriebssystemen, z.B. Microsoft Windows und Linux, Kenntnisse in der Programmiersprache C#, ein grundsätzliches Verständnis des Aufbaus von Anwendungen mit grafischen Benutzeroberflächen, der Umgang mit einer integrierten Entwicklungsumgebung (IDE) und ein Basisverständnis des .NET-Frameworks. Die betriebssystemübergreifende Programmentwicklung ist kein geeignetes Betätigungsfeld für Einsteiger, für die aber die kostenfreie Entwicklungsumgebung Visual C# 2012 von Microsoft empfehlenswert ist (siehe Elektor Februar und März 2009 [3]). Ist der Anfang dort gelungen, so sind Versuche zur plattformübergreifenden Entwicklung nicht mehr fern. Dann ist Mono eine interessante Option.

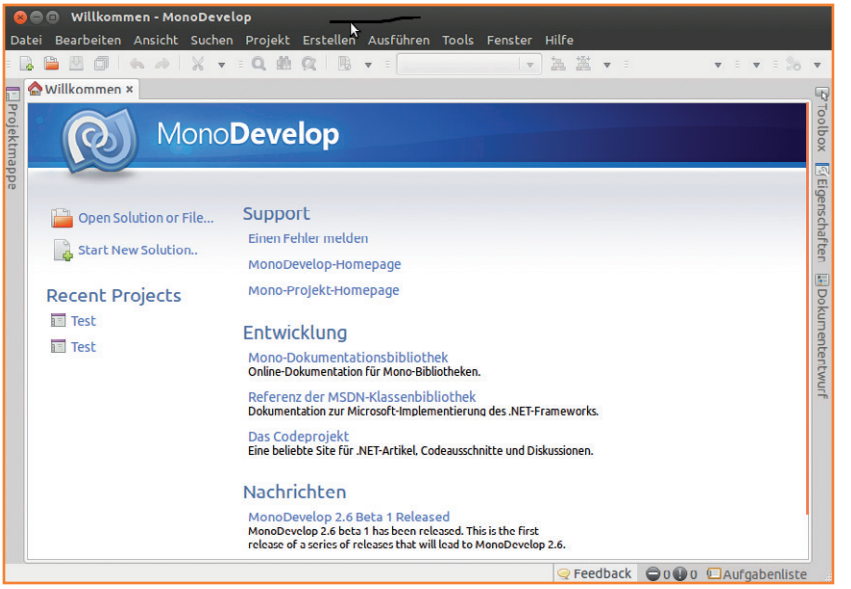


Bild 1. Erster Eindruck von MonoDevelop (Version 2.8) unter Ubuntu-Linux.

Entwicklungsumgebung

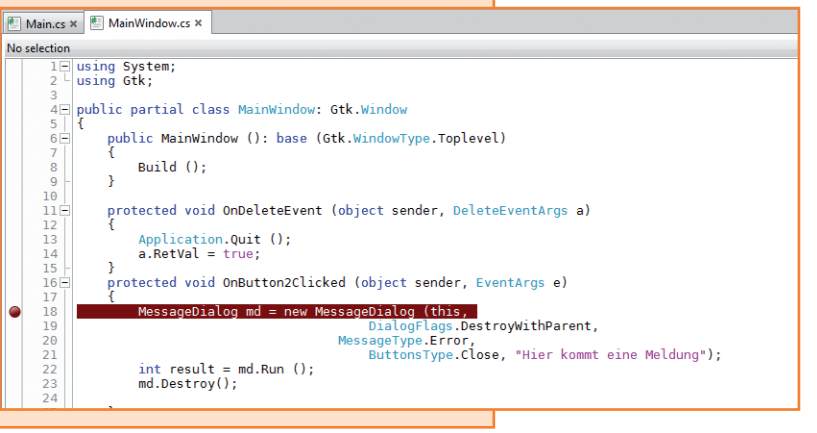
Mono selbst ist zunächst nur das Pendant zur .NET-Laufzeitumgebung. Für die Entwicklung von Programmen wird noch eine Entwicklungsumgebung benötigt. Dazu wird MonoDevelop bereitgestellt, eine vollständig integrierte Entwicklungsumgebung (IDE). Vom Aufbau, der Bedienung

Ein Streifzug durch die Entwicklungsumgebung

Die Entwicklungsumgebung MonoDevelop Version 4 bietet genügend Potenzial für eine umfangreiche Beschäftigung. Der Aufbau der Programmoberfläche entspricht dem typischen Design einer IDE. Wichtige Aspekte sind:

- die Anlage eines neuen Projektes erfolgt mit FILE | NEW SOLUTION…
- unter PROJEKT | OPTIONEN… können Voreinstellungen für das Projekt, zum Beispiel für den Compiler vorgenommen werden
- Unter BUILD bzw. RUN lässt sich ein Build des Projektes/Solution veranlassen bzw. der Start der Anwendung mit oder ohne Debugging durchführen.
- Im Quellcode können Breakpoints gesetzt werden (siehe Screenshot). An dieser Stelle wird das Programm während der laufenden Ausführung unterbrochen und es werden Informationen, beispielsweise zu Variablen oder Objekten, zur Verfügung gestellt.
- VERSION CONTROL: Es lässt sich ein Versionskontrollsystem anbinden und steuern.
- Erweiterungen der IDE können über TOOLS | ADD-IN-MANAGER… installiert, deinstalliert und verwaltet werden.

Bevor man ein ernsthafteres Projekt startet, sollte man sich mit der Bedienung und Arbeitsweise der IDE vertraut machen. Das Erstellen von Oberflächen will geübt werden. Außerdem gilt immer der Grundsatz: Das eigene Projekt regelmäßig speichern!



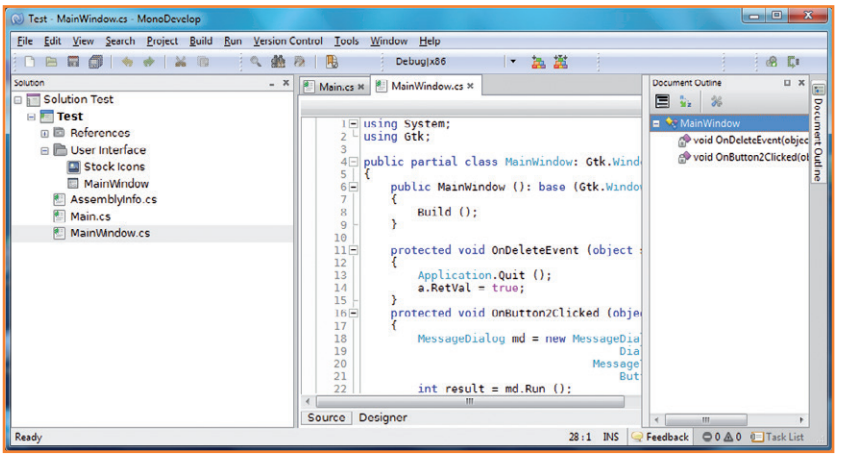


Bild 2. MonoDevelop 3.0 unter Microsoft Windows.

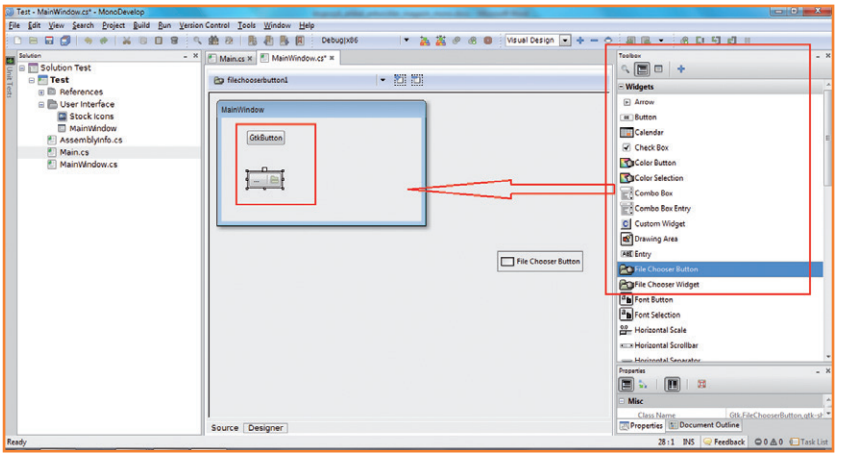


Bild 3. Einsatz des grafischen Designers von MonoDevelop.

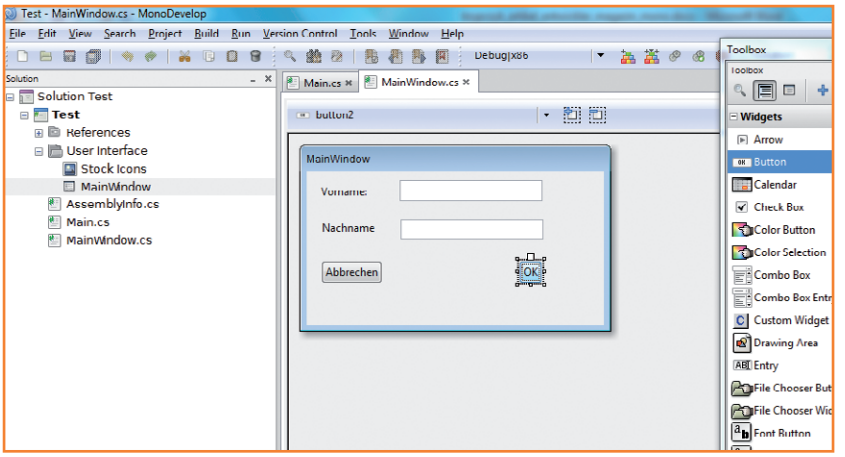


Bild 4. Entwurf der Benutzeroberfläche für das Testprojekt.

und der Konzeption ist diese mit Visual Studio zu vergleichen (siehe Textkasten). Die aktuellsten Informationen zu MonoDevelop findet man unter [4].

Installation unter Linux …

Beginnen wir mit der Arbeit unter Linux. Als Betriebssystem nutzen wir Ubuntu in der Version 12.04 LTS, andere Distributionen sind natürlich möglich. Zuerst gilt es, Mono und MonoDevelop unter Linux zu installieren. Dazu wechselt man in das Ubuntu-Softwarecenter und wählt (über die Suchfunktion) MonoDevelop als zu installierendes Paket. Ubuntu bezieht die notwendigen Ressourcen automatisch aus dem Internet und die Installation läuft ohne weitere Eingriffe selbstständig ab. Zu beachten ist, dass die aktuellste Version von MonoDevelop noch nicht für alle unterstützten Betriebssysteme als vorcompiliertes Paket zur Verfügung steht, so wie auch unter Ubuntu. Hier wird auf die Version 2.8 ausgewichen, was wir akzeptieren. Will man unmittelbar die neueste Version nutzen, sind die Quellen von der Webseite des Projektes zu laden.

Nach der Installation geht es unmittelbar an einen ersten Versuch. Wir starten MonoDevelop (**Bild 1**). Danach wird ein neues Projekt mittels des Befehls DATEI | NEU | PROJEKTMAPPE... angelegt. Auf dem eingeblendeten Dialogfeld wird in der Rubrik C# ein GTK#2.0-Projekt ausgewählt. Dazu sind ein Name und der Speicherort für das Projekt zu vergeben. Im nächsten Dialogfeld ist die GTK#-Version auszuwählen. Wir entscheiden uns für die aktuellste Version (2.8). Mono kann grundsätzlich mit verschiedenen grafischen Benutzeroberflächen umgehen, u.a. mit GTK, Qyoto oder Qt4Dotnet.

Die Frage ist nun, welche Bibliothek am besten geeignet ist. Auf der Seite des Projektes finden sich dazu einige Hinweise [5]. Für die folgenden Experimente werden wir die GTK#-Bibliothek einsetzen. Das GTK+ (GIMP Toolkit) ist eine universelle Umgebung zur Gestaltung von grafischen Benutzeroberflächen [6]. Es umfasst einen vollständigen Satz von Steuerelementen, beispielsweise zur Gestaltung von Dialogfeldern. GTK+ ist in C++ programmiert und zunächst für eine Verwendung durch die Programmiersprache C++ vorgesehen. Die Anbindung an das Mono-Framework bzw. die Sprache C# wird durch einen Wrapper erreicht. Die Verwendung von GTK durch Mono (C#) wird durch den Zusatz von „#“ deutlich gemacht.

… und Windows

Nun kehren wir zu Microsoft Windows zurück und machen uns dort an die Installation von Mono und MonoDevelop. Zu einer vollständigen Installation gehört auch die aktuelle GTK-Version. Für Windows steht MonoDevelop in der Version 4 als Installationspaket zur Verfügung. Nach dem Start ist man positiv überrascht. Die IDE wirkt modern und leichtgewichtig; sie erinnert (wen verwundert es) an Visual Studio 2010 (**Bild 2**). Auch unter Windows legen wir nun ein erstes Testprojekt an; ausgewählt werden C# und GTK 2.0. Mittels des grafischen Designers (**Bild 3**) lässt sich schnell ein rudimentärer Entwurf der grafischen Benutzerschnittstelle vornehmen.

Ein erster Test

Auf dem Hauptformular werden nun zwei Labels, zwei Textboxen (Entry) und zwei Buttons platziert (**Bild 4**). Zuvor ist jedoch ein entsprechender Layoutcontainer anzugeben. Für eine absolute Platzierung der Steuerelemente wird ein Container vom Typ Fixed ausgewählt. Die Verbindung zwischen der Benutzerschnittstelle und der Programmlogik erfolgt über sogenannte Signale. Dieses entspricht der Ereignisbindung in Windows-Forms. Gebunden wird das Signal Clicked. Beim Doppelklick in der IDE auf dieses Signal wird der zugehörige Code-Rumpf erzeugt. Diesen gilt es mit Leben zu füllen. Im Beispiel soll beim Klick auf den OK-Button ein Meldungsfenster (MessageBox) angezeigt werden. **Listing 1** zeigt den zugehörigen Quellcode. Eine Besonderheit bei GTK: Die Zerstörung des Objektes muss manuell erfolgen. Informationen zur Syntax der GTK#-Bibliothek kann man unter [7] nachlesen. Nach dem Compilieren lässt sich die Anwendung wie geplant starten (**Bild 5**). Die Verzeichnisstruktur entspricht im Wesentlichen derjenigen, wie sie von Visual Studio erzeugt wird; unter anderem wird eine ausführbare Datei in Form eines exe-Files erzeugt.

Unser Mini-Projekt wird nun auch unter Linux getestet. Dazu erfolgt ein Transfer der Anwendung auf das Ubuntu-System. Es muss lediglich die ausführbare Datei gestartet werden (**Bild 6**). Dieses geschieht über das Kontextmenü (rechte Maustaste) durch Aufruf des Eintrages AUSFÜHREN MIT MONO-RUNTIME.

Portierung von Anwendungen

Werden Anwendungen vollständig neu entwickelt und ist ein Einsatz auf unterschiedlichen Betriebssystemen geplant, so kann dieses bei der Programmierung berücksichtigt werden. Oft wird jedoch die Anwendung bereits für .NET (also für das Betriebssystem Microsoft Windows) vorliegen und man muss überlegen, mit welchem Aufwand eine Portierung möglich ist. Dazu sind letztendlich die Klassen des .NET-Frameworks durch ihre Äquivalente in Mono zu ersetzen. Um den Aufwand und die Erfolgsaussichten einzuschätzen, steht das Tool Mono Migration Analyzer (MoMA) zur Verfügung. Nach der Auswahl

Listing 1: Quellcode der Test-Anwendung

```
protected void OnButton2Clicked(object sender, EventArgs e)
{
MessageDialog md = new MessageDialog (this,
DialogFlags.DestroyWithParent,
MessageType.Error,
ButtonsType.Close,
“Hier kommt eine Meldung”);
int result = md.Run();
md.Destroy();
}
```

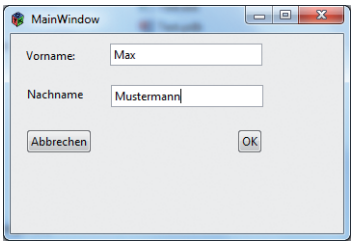


Bild 5.
Die Testanwendung unter Windows 7.

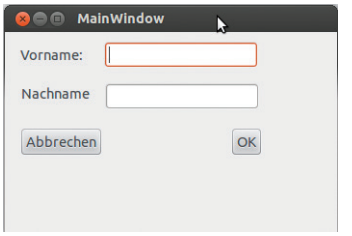


Bild 6.
Die Testanwendung unter Linux.

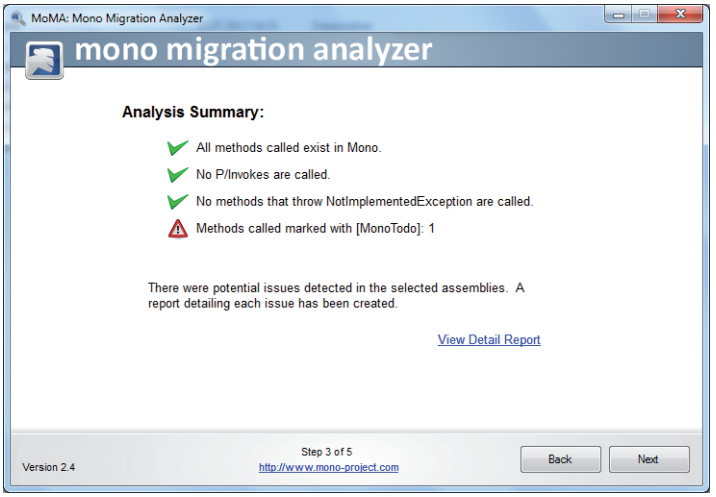


Bild 7.
Der Mono Migration Analyzer (MoMA) als Hilfe bei der Portierung von Anwendungen auf Mono.

einer exe- oder dll-Datei wird der Analysevorgang gestartet (**Bild 7**).
Wurden Schwierigkeiten festgestellt, ist es die Aufgabe des Entwicklers, die spezifischen Programmabschnitte (z.B. direkte API-Aufrufe) durch entsprechende Methoden in Mono zu ersetzen. Dann folgt der endgültige Portierungsprozess.

Mono für weitere Systeme

Hat man sich in Mono eingearbeitet, bietet das Framework noch eine Menge mehr. Mono kann auch dazu verwendet werden, Apps für die mobilen Plattformen Android und iOS bereitzustellen. Bevor man jedoch die dazu notwendigen kostenpflichtigen Lizenzen (siehe oben) erwirbt, sollte man sich mit Mono umfassend vertraut machen. Dazu ist die Open-Source-Implementierung für den PC bestens geeignet. Gerade dies kann eine interessante Option für den Elektroniker darstellen, zum Beispiel zur Steuerung externer Hardware.

(120490)

Weblinks

[1] http://de.wikipedia.org/wiki/Mono-Projekt
[2] www.mono-project.com/Main_Page
[3] www.elektor.de/080668
[4] http://monodevelop.com/
[5] www.mono-project.com/Gui_Toolkits
[6] www.gtk.org/overview.php
[7] http://buttle.shangorilla.com/1.1/handlers/monodoc.ashx?link=T%3AGtk.MessageDialog

Über den Autor

Dr. Veikko Krypczyk studierte Betriebswirtschaftslehre mit dem Schwerpunkt Wirtschaftsinformatik und promovierte zum Thema Algorithmenentwicklung für Tourenplanungsprobleme.
Er arbeitet als Fachautor, berät Unternehmen in Fragen der Softwareentwicklung und ist begeisterter Programmierer. In seiner Freizeit heizt er gelegentlich den Lötkolben an und „werkelt“ an dem einen oder anderen Projekt.

Für Fragen und Anregungen senden Sie bitte eine E-Mail an info@it-fachartikel.de.
Ergänzende Informationen findet man unter http://it-fachartikel.de.

Lesen Sie jetzt Elektor auf Ihrem Tablet!
Einzelheft: nur 7,99 €
Doppelheft: nur 14,99 €
Elektor-Mitglieder erhalten jede App-Ausgabe kostenlos!
Elektor-App herunterladen:
• www.elektor.de/ios-app
• www.elektor.de/android-app

Philips PR9150/PR9151 Oberflächen-Rauheitstester

Kratzen an der Oberfläche der 50er Jahre

Von **Jan Buiting,**
Elektor-Redaktion

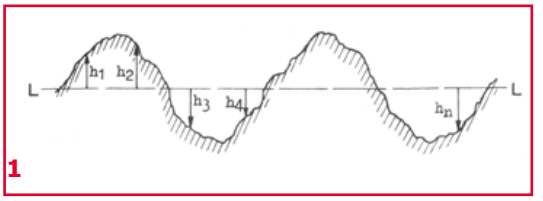
Fast alle meine Bekannten hassen die mechanische Bearbeitung von Werkstoffen und insbesondere von Metallen. Ich weiß, das wirft kein gutes Bild auf die Branche der Elektroniker. Die meisten meiner e- und mech-Freunde sind aber mit CAD & CAM aufgewachsen und begeistern sich zurzeit eher für so heiße Innovationen wie das 3D-Printing.

In der Mitte der 1950er Jahre war die Firma Philips in den Niederlanden berühmt für die hervorragende Qualität der mechanischen Baugruppen in ihren Test-, Mess- und Kommunikations-Geräten. Dies muss in erster Linie an den wissenschaftlichen Grundlagen der renommierten Philips-Labors (NatLab) gelegen haben, wo einigen der brillantesten Physikern und Maschinenbauern völlige Freiheit gegeben wurde, ihre Kunst zu perfektionieren, ohne dass dabei jeder Gulden zweimal umgedreht wurde.
Zu dieser Zeit war Philips auch ein angesehener Hersteller von wissenschaftlichen Prüf- und Messgeräten, die in Labors und Werkstätten rund um den Globus eingesetzt wurden. Philips „PR"-Reihe war natürlich elektronisch, aber nicht auf elektronische Anwendungen beschränkt. Beispiele hierfür sind Säure-Tester und Messgeräte für die Leitfähigkeit von Flüssigkeiten, den Schalldruck, alle Arten von Gasen, Vibration, Röntgenstrahlen und so weiter und so fort.

Im Jahr 1952 hat sich der Philips-Physiker G.W. van Santen mit dem Kuddelmuddel um die Definitionen der Mechaniker zum Thema Oberflächenrauheit (SR) von Metallen beschäftigt. An einem Ende dieses Spektrums standen sehr teure Instrumente, die ein paar finanzkräftige Labore ihr eigen nennen konnten, auf der anderen alte Hasen auf dem Gebiet der Metallverarbeitung, die mit geschlossenen Augen ihre Fingernägel benutzten, um die Glätte einer Oberfläche nach Schleifen, Läppen, Honen und Polieren zu beurteilen. Immerhin kann man mit etwas Übung eine Oberflächenrauheit bis auf etwa 40 µm (der Dicke eines menschlichen Haares) ertasten.

Rauheit verwirrt und definiert

Die Definition der Rauheit ist eine des (G)rauens und nicht unähnlich dem Turmbau zu Babel. Begeben wir uns auf eine Zeitreise ins Jahr 1952! **Bild 1** zeigt den (überhöht dargestellten) Verlauf einer bearbeiteten Fläche. Die Oberflächenrauheit sollte seinerzeit mit international normierten



1

Größen und Einheiten eindeutig beziffert werden. Das Gegenteil war der Fall!

Einige Kostproben zum Thema Oberflächenrauheit:

- **Rautiefe R_z**, der Höhenunterschied zwischen höchstem Punkt und tiefstem Tal (meist) in Mikron. Dies bildet die Grundlage der veralteten DIN 4762.

- **Maximale Einzelrautiefe R_{max}**, die größte Abweichung auf einer gegebenen Strecke von einer mittleren Oberflächenlinie.

- **Quadratischer Mittelrauwert R_q (h_{rms}, h_{eff})**, die Quadratwurzel des arithmetischen Mittels der quadrierten Abweichungen von der Grundlinie über eine definierte Länge gemessen.

$$h_{eff} = \sqrt{\frac{h_1^2 + h_2^2 + h_3^2 + ... h_n^2}{n}}$$

Die US-amerikanische Norm ASA B46 benutzt durchgehend h_{rms} in der Einheit Mikro-Zoll.

- **Mittlere Rauheit R_a (h_{avg}),** das arithmetische Mittel der Absolutwerte aller Rautiefen.

$$h_{avg} = \frac{h_1 + h_2 + h_3 + ... h_n}{n}$$

Der British Standard BS 1134 definiert die durchschnittliche Rautiefe als Norm, in Mikro-Zoll ausgedrückt.
Die niederländische Normierungskommission empfiehlt ebenfalls die durchschnittliche Rautiefe als Standard.

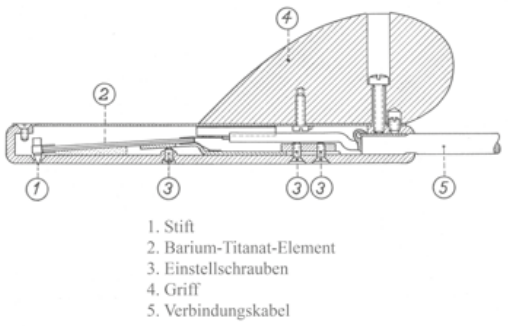
Einheiten der Rauheit

Für eine glatte (Metall-)Oberfläche ist Mikron (Mikrometer µm; 10^{-6} m) ein ziemlich grobes Maß, das wahrscheinlich nur Fans kleiner Zahlen begeistern würde. Und Milli-Mikron ist leider zu klein und würde zu unpraktisch großen Zahlen führen. Mikro-Zoll dagegen wäre ein guter Kompromiss zwischen groß und klein, da die meisten Oberflächen Rauheiten im Bereich 1...500 Mikro-Zoll besitzen. Zum Beispiel:

Glatte Oberfläche: 0,05 µm = 2 µ'' = 50 mµ
Raue Oberfläche: 6,3 µm = 250 µ'' = 6300 mµ

Da in zivilisierten Ländern aber natürlich nur met-

2



3

rische ISO-Einheiten in Betracht kommen, hat man einen Trick angewandt und die Einheit „ru“ ins Leben gerufen. In vielen Ländern erinnert „ru“ irgendwie an Rauheit (**Rau**heit, **r**oughness **u**nit, **ru**gosité, **ru**wheid in Deutsch, Englisch, Französisch, Niederländisch).
Eine Oberfläche weist eine Rauheit von 1 ru auf, wenn die mittlere Rautiefe R_{avg} = 1/40 Mikron (≈ 1 micro-inch) beträgt.
Das Problem war aus dem Weg geräumt und es konnten die Rauheitsklassen R1 bis R6 festgelegt werden, zusammen mit einer Reihe von Symbolen in Konstruktionszeichnungen, die von Arbeitern an Dreh- und Fräsmaschinen leicht erlernt und erkannt werden konnten.

Oberflächenmessung mit PR9150/ PR9151

Der einzige Unterschied zwischen den beiden Geräten liegt in den Röhrentypen. PR9151 ist mit amerikanischen Röhren ausgestattet (zum Beispiel 6X4 statt AZ41). Leider verfüge ich über keinen Schaltplan, den ich hier abbilden könnte, aber ich erkunde stets das Innere von solchen Geräten (**Bild 2**). Werfen wir einen Blick in die seit 60 Jahren unberührten Eingeweide: Die Elektronik ist mit einem einstellbaren Eingangsabschwächer (für die Rauheitsklassen R2 bis R5), einem Verstärker, einem Gleichrichter und einem Drehspulinstrument aufgebaut.

Tabelle 1. Versionen des PR9150/9151

Typ	Skalierung	Messbereiche	Rauheit-Referenz
PR9150/01 PR9151/01	Skala /01 (Bild. 5a) für Vergleich; Rot= gültig, Grün = verworfen Multiplikation (mit angezeigtem SR-Faktor in Bezug auf Referenzstück)	1 – 300 ru, unterteilt in 4 Bereiche	Referenzstück oder PR9180/00 (Ra Kalibrierungswerte: 125 – 32 – 8 und 2 ru (3,2 – 0,8 – 0,2 und 0,05 µ))
PR9150/02 PR9151/02	Skala /02 (Bild 5b) kalibriert für Ra in ru (=C.L.A. in µ’’); Skala zeigt SR-Werte wie von der Norm empfohlen	I: 50-280 ru II: 10-70 ru III: 3-16 ru IV: 1-4 ru	PR9180/00 (siehe oben)
PR9150/03 PR9151/03	Skala /03 (Bild 5c) mit dezimaler Teilung, kalibriert für Ra in ru.	I: 50-250 ru II: 10-60 ru III: 3-16 ru IV: 1-4 ru	PR9180/00 (siehe oben)
PR9150/04 PR9151/04	Skala /04 (Bild 5d) kalibriert für Rt = Hmax in µ	I: 5-25 µ II: 1-6 µ III: 0,3-1,6 µ IV: 0,1-0,4 µ	PR9180/02 (Ra Kalibrierungswerte: 12,5 – 3,2 – 0,8 und 0,2 µ)
PR9150/05 PR9151/05	Skala /05 (Bild 5e), kalibriert für Ra in µ	I: 1-6 µ II: 0,3-1,6 µ III: 0,1-0,4 µ IV: 0,02-0,1 µ	PR9180/00 (siehe oben)

Die Elektronik ist in einem schönen Holzkasten mit Leder-Tragegriff und Klappdeckel untergebracht. Das Ganze wiegt etwa 6 kg. Immer, wenn ich das Gehäuse öffne und andere Menschen sind anwesend, denken diese, ich würde an ihnen eine Strahlungsmessung vornehmen.

Der Tastkopf

Der Höhepunkt des Instruments ist der Tastkopf, dessen Aufbau in **Bild 3** dargestellt ist. Er enthält einen piezoelektrischen Kristall aus Barium-Titanat, der vom Tastkopfgehäuse auf der einen Seite gehalten wird und von einem Stift auf der anderen. Der Stift ist eine synthetische Saphirnadel mit einer Härte von 2.000 HV (Vickers-Härte) und einem Endradius von etwa 60 µm. Dies ermöglicht es, dem Rauheitsprofil veredelter Oberflächen in seine tiefsten Täler mit Steigungen >150° zu folgen. Die Tastkopfspitze ist leicht zu sehen und zu fühlen. Wäre der PR9150/9151 mit einem Lautsprecher ausgestattet, könnte man wahrscheinlich Schallplatten von Joe Cocker oder Janis Joplin in guter Qualität abspielen.
Die Sonde ist so klein, dass sie in Bohrungen bis zu 8 mm Durchmesser eingesetzt werden kann. Sie ist über ein Stück abgeschirmten Kabels und einen massiven, sehr hochwertigen Stecker mit dem Gerät verbunden. Der Tastkopf (**Bild 4**) besitzt einen elegant gestylten Bakelit-Griff.

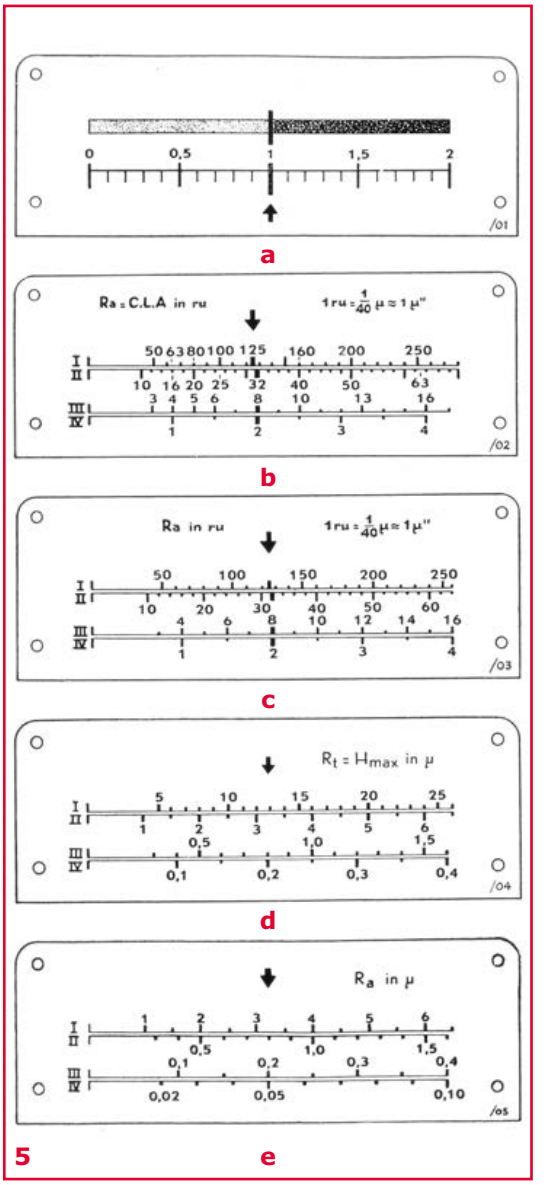
Norm nach Wahl

Offenbar war man bei Philips nicht allzu zuversichtlich, dass sich die neu normierte Einheit **ru** durchsetzen würde (was sie dann ja auch nicht tat). So stellte ich erstaunt und fasziniert fest, dass fünf verschiedene Varianten des PR9150/9151 im Handbuch des ersten Geräts, das ich vor etwa zwei Jahren erhielt, erwähnt wurden. Ich habe die Unterschiede in **Tabelle 1** zusammengefasst. Jede Version besitzt eine andere Skalierung des Instruments, um für die verschiedenen Märkte, Wünsche der Einkäufer und auch Anwendungen vorbereitet zu sein. Die **Bilder 5a-d** zeigen die verschiedenen Skalen (aus dem Handbuch kopiert). Ich besitze mittlerweile die PR9151-Gerätevarianten /01, /02 und /05, auch drei Tastköpfe (eine mit einem gebrochenen Stift) und ein Handbuch. Doch das ist noch nicht alles.

Originale Calibration Box!

PR9150/9151 sind nicht gerade extrem selten, die im Handbuch erwähnte *Calibration Box* ist es

4



5

6

ESTD 2004

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

schon! Ich habe schließlich eine aufgetrieben, sie war im Lieferumfang der dritten PR9151 (klar, das am meisten verdreckte Gerät) enthalten.

Natürlich kann man vergleichende Messungen mit dem PR9150/9151 vornehmen. Die Vertriebs- oder Werbeabteilung kann aber dem Kunden schlecht mitteilen, dass die Oberfläche von Material A dreimal glatter ist als die von Material B, sie benötigt absolute Zahlen! Deshalb muss das Gerät vor den Messungen kalibriert werden. Die *Calibration Box* PR9128/00 in **Bild 6** besteht aus stabilem Bakelit. Achten Sie auf die geschlossenen und offenen Dreieck-Symbole, die vier Klassen der Oberflächenrauheit bezeichnen. Die genauen Werte sind mit der Hand geschrieben.

Ich konnte leicht die Unterschiede in der Rauheit der vier Metall-Referenzstücke sehen und Fräsrillen der beiden „härtesten“ Proben ertasten. Die beiden anderen Stücke lassen sich ein wenig schwieriger auseinanderhalten. Alle Stücke sind dünn verchromt, was aber keinen Einfluss auf ihre Oberflächenrauheit nimmt. Ist da jemand, der eine PR9180/02-Box besitzt?

Anwendung

Nach der Kalibrierung des Geräts mit Hilfe der Referenzstücke der *Calibration Box* soll der Tastkopf in einem Winkel von 90 Grad in Bezug auf die Richtung der Bearbeitung über die Oberfläche bewegt werden. Dazu bewegt man die Sonde zwei bis drei Mal pro Sekunde über eine Strecke von 1,5...2 cm nach oben und unten oder dreht das Werkstück langsam (etwa in einer Drehbank oder Fräse). Die Geschwindigkeit der Sonde zum Objekt sollte mindestens 4 cm/s betragen. Die Tastkopfunterseite muss die Oberfläche über die gesamte Länge berühren. Obwohl der PR9150/9151 das Signal über ein Hochpass-Filter schleust, um die Auswirkungen unkontrollierter Handbewegungen zu kompensieren, ist etwas Erfahrung notwendig, um vertrauenswürdige Ergebnisse zu erzielen. Und ja, das Testen schädigt die Oberfläche, der Tastkopf hinterlässt Kratzer.

In einem recht langen Kapitel der Bedienungsanleitung kommt Philips zu dem Schluss, dass die Ergebnisse einen Fehler von weniger als 20 % aufweisen, was als ganz ausgezeichnet gepriesen wird („outstanding, considering that extremely accurate and costly roughness meters achieve about 15 % precision due to the non-homogeneous structure of ordinarily finished surfaces“).

Heute gibt es Messgeräte zur Oberflächenrauheit in Hülle und Fülle bei Ebay & Co. Sie nutzen deutlich verbesserte Technologien zur Messung von Unebenheiten als wie im Jahr 1955 piezoelektrische Tastköpfe und ein oder zwei Röhren. Dennoch sieht man sie selten in metall-verarbeitenden Betrieben, die mit Honen und Präzisions-Polieren zu tun haben. Ich denke, die gute alte Fingernagel-Methode, gepaart mit einem gerüttelt Maß an Handwerkskunst und Erfahrung kann noch immer ihre Vorteile gegenüber hypermodernen, Tausende von Euros teuren elektronischen Messgeräten geltend machen.

(130057)

CIRCUIT CELLAR

The CC Weekly <Code/> Challenge is underway!

Each week, you'll find a new snippet of source code that contains one error.

>>>

```
1  #include <stdio.h>
2
3  int main (int argc, char *argv[])
4  {
5    char ch;
6    int i;
7
8    for (i = 0; i < argc; ++i) {
9      printf ("Argument #%d: '%s'\n", i, *argv[i]);
10   }
11 }
```

sample code only

If you can find the error, you could be a winner!

Follow *Circuit Cellar* on Facebook and Twitter for information about each week's challenge, prizes, and winners announcements.

@circuitcellar
@editor_cc

circuitcellar

For complete details, visit circuitcellar.com/cc_weekly_code_challenge

SPONSORED BY IAR SYSTEMS

Hexadoku Sudoku für Elektroniker

Es ist nicht wichtig, ob Sie unser bekanntes Rätsel in einem Rutsch lösen können oder ob Sie drei Wochen darüber brüten: Was zählt ist das Ergebnis. Dieses sollten Sie uns – in Form der Ziffern in den grauen Kästchen – gleich zusenden. Hier warten nämlich vier schöne Gutscheine auf die glücklichen Gewinner!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist. Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst - sprich die Zahlen in den grauen Kästchen herausfindet - kann wie jeden Monat einen Hauptpreis oder einen von drei Trostpreisen gewinnen!

Mitmachen und gewinnen!

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

Einsenden

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail, Fax oder Post an:
Elektor – Redaktion – Süsterfeldstr. 25 – 52072 Aachen
Fax: 0241 / 88 909-77 E-Mail: hexadoku@elektor.de
Als Betreff bitte nur die Ziffern der Lösung angeben!
Einsendeschluss ist der 30. September 2013!

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

Die richtige Lösung ist: **F9407**.
Der Eurocircuits/Elektor-PCB-Service-Gutschein im Wert von 100 € geht an: Ciril Zalokar aus Ljubljana (Slowenien).
Einen Elektor-Gutschein über je 50 € haben gewonnen: Arne Jansson, Philippe Monnard und Gerard Yvrau.
Herzlichen Glückwunsch!

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

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

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

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

elektor GOLD membership

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

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

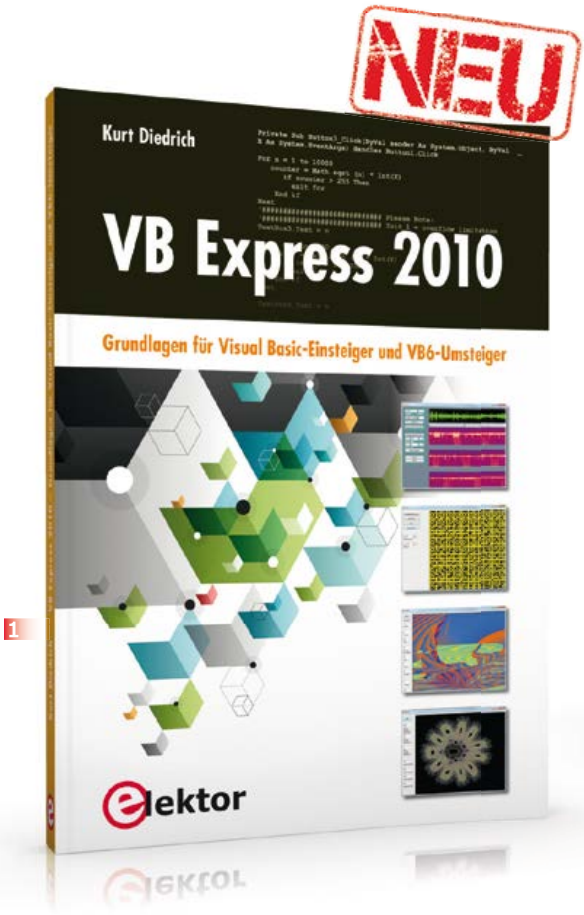
UMWELTSCHONEND – GÜNSTIG – GREEN

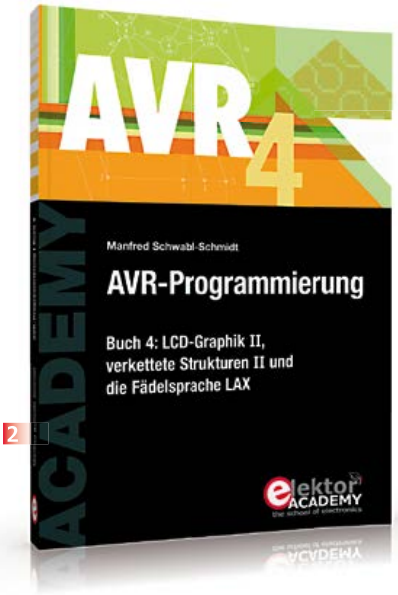
Möchten Sie Elektor lieber im elektronischen Format beziehen? Dann ist die neue GREEN-Mitgliedschaft ideal für Sie!
Die GREEN-Mitgliedschaft bietet (abgesehen von den 10 Printausgaben) alle Leistungen und Vorteile der GOLD-Mitgliedschaft.

elektor GREEN membership

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

Elektor Store





Grundlagen für Visual Basic-Einsteiger und VB6-Umsteiger

1 VB Express 2010

NEU

Dieses Buch unterstützt den Anwender bei den ersten Schritten mit Visual Basic, in dem es sich auf die Werkzeuge der Toolbox und deren Eigenschaften konzentriert, die zum Schreiben praktisch verwertbarer Programme notwendig sind. Zu jedem Thema findet der Leser ausführlich kommentierte Beispielprogramme, die er selbst ausprobieren kann und die sich auf das Mindeste beschränken, was zum Starten der Software notwendig ist.

284 Seiten (kart.) • ISBN 978-3-89576-269-7
€ 34,80 • CHF 43,20

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

2 AVR-Programmierung 4

In diesem neuen vierten Band der erfolgreichen Buchreihe zur Programmierung von AVR-Mikrocontrollern wird die LCD-Graphik aus Band 3 weiterentwickelt. Hinzu kommen das Füllen von Polygonen, die Zuordnung von Pixelkoordinaten zu Graphikobjekten und die Verwendung des Displays als Textfenster. Aufbauend auf der Darstellung der inneren Mechanik von Fädelsprachen im vorigen Band wird außerdem die Fädelsprache LAX vorgestellt und implementiert.

320 Seiten (kart.) • ISBN 978-3-89576-232-1
€ 46,00 • CHF 57,10

Schaltungsprojekte für Profis

3 Arduino

Für den großen Erfolg der Arduino-Plattform lassen sich zwei Ursachen finden. Zum einen wird durch das fertige Board der Einstieg in die Hardware enorm erleichtert; der zweite Erfolgsfaktor ist die kostenlos verfügbare Programmieroberfläche. Unterstützt wird der Arduino-Anwender durch eine Fülle von Software-Bibliotheken. Die täglich wachsende Flut von Libraries stellt den Einsteiger vor erste Probleme. Nach einfachen Einführungsbeispielen ist der weitere Weg nicht mehr klar erkennbar, weil oft detaillierte Projektbeschreibungen fehlen. Hier setzt dieses Buch an. Systematisch werden Projekte vorgestellt, die in verschiedene Themengebiete einführen. Dabei wird neben den erforderlichen theoretischen Grundlagen stets größter Wert auf eine praxisorientierte Ausrichtung gelegt.

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

Apps programmieren – Schritt für Schritt

4 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

4

6

7

5

8

Bestücke und getestete Platine

5 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

30 Projekte in C für Fortgeschrittene

6 ARM-Mikrocontroller 2

Die im Buch beschriebenen Projekte mit dem mbed-Board sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung.

Es bedeutet, dass man keinerlei Software auf seinem PC installieren muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC.

243 Seiten (kart.) • ISBN 978-3-89576-271-0
€ 39,80 • CHF 49,40

Kompletter Elektor-Jahrgang 2012 auf DVD

7 Elektor-DVD 2012

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

ISBN 978-90-5381-273-0
€ 27,50 • CHF 34,10

Taschenbuch für Elektronik und Elektrotechnik

8 Formelsammlung

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

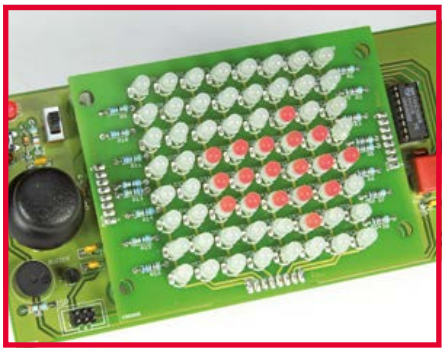
271 Seiten (kart.) • ISBN 978-3-89576-251-2
€ 29,80 • CHF 37,00

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

www.elektor.de/shop

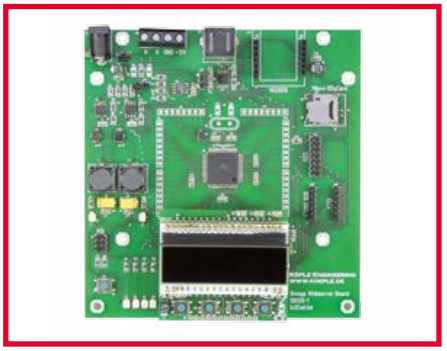
Elektor-Verlag GmbH
Süsterfeldstr. 25
52072 Aachen
Tel. +49 (0)241 88 909-0
Fax +49 (0)241 88 909-77
E-Mail: bestellung@elektor.de

Nächsten Monat in Elektor



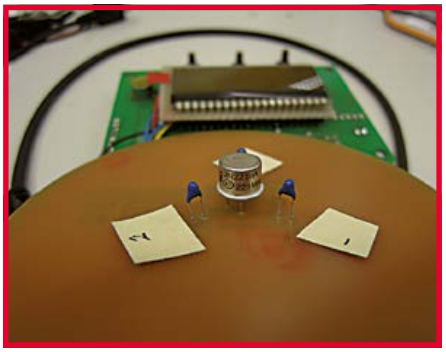
Zweifarben-LED-Matrix 8 x 8 für ATmega328P

Mit der LED-Matrix aus zweifarbigen, quadratisch angeordneten LEDs setzen wir das Programmieren von Mikrocontrollern aus dem Hause Atmel anschaulich in Szene. Schon allein der visuelle Eindruck ist sicher einen Blick wert. Nebenbei können wir das Bitshift-Verfahren erläutern, mit dem die 64 LEDs gesteuert werden. Die Unterhaltung kommt nicht zu kurz, denn zu unseren Programmen gehört ein Spiel, das auf der LED-Matrix läuft. Hilfreich sind hier Grundkenntnisse der Programmiersprache C oder C++.



Xmega-Webserver-Board

Es war schon angekündigt, doch es hat sich verspätet: Ein neues Mikrocontroller-Board mit einem leistungsstarken Mikrocontroller aus der AVR-Familie von Atmel. Die Verbindungen zur Außenwelt laufen über serielle RS485- und UART/TTL-Schnittstellen, dort kann zum Beispiel der USB-TTL-Umsetzer „BOB“ angeschlossen werden. Über den „Embedded Extension Connector“ ist das Board vielfältig erweiterbar. Für ein TCP/IP-Modul ist Platz reserviert, so dass Webserver- und andere Netzwerkanwendungen realisiert werden können. Vorhanden sind ferner ein Steckplatz für eine Mikro-SD-Speicherkarte, ein abnehmbares Display, vier LEDs und vier Taster.



Windmesser

Die meisten Windmesser, auch Anemometer genannt, messen Windgeschwindigkeiten mit Schalen- oder Flügelrädern. Die Windrichtungen werden von Windfahnen erfasst, die sich im Wind drehen. Unser Windmesser folgt einem anderen Prinzip, das ohne Mechanik auskommt. Wir haben es vom thermischen Flussmesser übernommen: Ein Heizelement erwärmt die strömende Luft. Abhängig von Geschwindigkeit und Richtung wird die Luft zu Sensoren geleitet, die um das Heizelement gruppiert sind. Die Software interpretiert die Sensor-Signale und stellt das Ergebnis auf einem Display dar.

Änderungen vorbehalten! Elektor Oktober 2013 erscheint am 25. September 2013. Verkaufsstellen findet man unter: www.pressekaufen.de.

Rund um die Uhr und sieben Tage die Woche

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

Second Step

NEU!

Sie haben den „First Step“-Kurs erfolgreich durchgearbeitet und haben Lust auf mehr bekommen?
Dann machen Sie doch einfach den nächsten Schritt in die große Welt der Mikrocontroller!
Aufbauend auf Ihrem bisherigen Wissen beschäftigen Sie sich beim Fortsetzungskurs „Second Step“ mit weiteren interessanten und wichtigen Mikrocontroller-Anwendungen:
Steuern Sie zum Beispiel ein alphanumerisches LCD an, oder erweitern Sie Ihr System um einen Uhrenbaustein (Real Time Clock), oder lernen Sie eine doppelte UART-Schnittstelle und einen Watch Dog kennen und nehmen diese Bausteine in Betrieb!

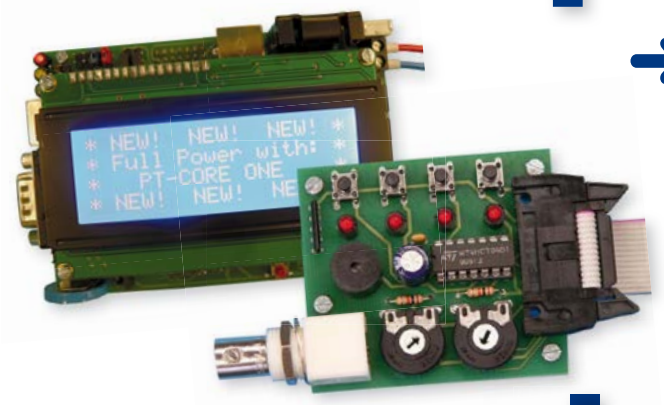
Bestandteile des „Second Step“-Pakets:

elektor ACADEMY
the school of electronics

3 Arbeitshefte

(inkl. passendem DIN A4-Ringbuch)

- Beschreibung der neuen Hardware
- Selbst geschriebene Funktionen, lokale und globale Variablen und Header-Dateien
- Externe parallele Peripherie-Einheiten: RTC, LCD und DUART
- Strings und Arrays
- Watch Dog und CS\-Logik
- Serielle Datenübertragung
- Umfang aller Arbeitshefte: 450 Seiten



1 PT-CORE-Mikrocontroller-Board + 1 PT-DAA-Zusteckkarte

- 8051er-Mikrocontroller AT89C51CC03
- 2,5-V-Referenzspannungsgeber für A/D-Wandler LT1009
- TTL/RS232-Pegelwandler MAX232
- CAN-Buskoppelstufe PCA82C250
- Watch Dog MAX807
- Uhrenbaustein mit Batteriepufferung RTC72421
- Alphanumerisches LC-Display (4x20 Zeichen)
- Doppelter UART-Baustein SCC2681
- Chip-Select-Dekodierung inkl. 4 freier CS\-Signale GAL16V8
- Datenlatch 74HCT573
- Karten-Format: 117 x 83 mm²
- Spannungsversorgung 9 V DC, Verpolungsschutzdiode und Miniatursicherung
- PT-DAA-Zusteckkarte enthält den kompletten optischen, akustischen und analogen Funktionsumfang des „First Step“-Boards

1 CD-ROM mit Zusatzinfos

- Datenblätter
- Systemdokumentation
- Entwicklungsumgebung
- Beispielprogramme

Das gesamte „Second Step“-Paket kostet nur 249,00 Euro.

Weitere Infos und Bestellung unter
www.elektor.de/second-step

Unsere Shopsprachen:

reichelt.de elektronik

- ✓ über 40 Jahre Erfahrung
- ✓ über 45.000 Produkte am Lager
- ✓ schneller 24-Std.-Versand
- ✓ kein Mindermengenaufschlag

Kundenbewertungen:

97.79% zufriedene Kunden

4.89 / 5.00

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

* Quelle: Shopauskunft.de (24.06.2013)

Profi-Löttechnik und Zubehör

Professionelle Qualität zu attraktiven Preisen!

UNVERZICHTBARE LÖT-HELFER

Druckluft 67

CRC

Zur Schmutz- und Staubentfernung an Feinmechanik, Optik, Elektronik usw. Keine Entzündungs- und Explosionsgefahr bei bestimmungsgemäßer Anwendung. Sehr ergiebig.

KONTAKT 334 14,20 400 ml

ESD-Armband

SPROTEK

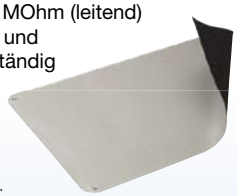
Antistatisches Kletband mit Ableit-Spiralkabel für sicheres Arbeiten an empfindlichen Bauteilen.

Bestellnummer:
ESD BAND 4,95

Tisch-Arbeitsmatten

Stat-X

- Elektrische Leitfähigkeit
- · Oben: 10 - 100 MOhm (ableitend)
- · Unten: 1 MOhm (leitend)
- Lötzinn- und hitzebeständig



Bestellnummer:
ESD ARBEITSMATTE 29,95 600 x 900
ESD MATTE 1200 34,50 600 x 1200

Löt- und Entlötstation

ZHONGDI ELECTRONIC TOOLS

- ✓ schnelle Aufwärmzeit
- ✓ hervorragende Hitze - Energie Umsetzung
- ✓ schnelle & präzise Temperaturregelung

- ESD-geschützt
- keramisches Heizelement
- in 30 Sek. von Raumtemperatur auf 350 °C
- energiesparende Sleep-Funktion
- 24V/60W Löt-, 80W Enlötkolben
- Vacuum : 600 mmHG

149,95
STATION ZD-987

Lötdampfabsauger

XYTRONIC

- inkl. 3 einfach zu wechselnde Ersatz-Carbonfilter
- Luftdurchsatz: ca. 120 m³/h
- Maße: 162 x 200 x 120 mm
- Filtergröße: 130 mm

Schützen Sie sich!

Bei Lötarbeiten entstehen Dämpfe, die als gesundheitsschädlich eingestuft werden.

Bestellnummer:
FILTER 426-DLX 28,95

Lötspitzen Reiniger

- Gebilde aus Messingschnitzeln
- Kein Wasser notwendig
- Kein Temperaturtropfen wie bei herkömmlichen Schwämmen

Bestellnummer:
CLEANER 460A 4,95 Reiniger inkl. Ablage
CLEANER 460A E 2,95 Ersatzreiniger

Löt-Hilfsmittel-Set, 6-teilig

Set mit unendbehrlichen Helfern bei filigranen Arbeiten.

Bestellnummer:
SOLDER AID TOOLS 4,95

SPROTEK

Premium Werkstatt-Lupenleuchte

Jetzt wird´s chic am Arbeitsplatz - Weiße Premium Werkstatt-Lupenleuchte mit 152 mm Echtglas-Linse im neuen edlen Design mit superflachem Kopf und neuer Optik.

- 90 ringförmig angeordnete Ultra-Efficiency-LED mit Diffusor
- Schattenfreie Ausleuchtung
- Innenliegende Feder- und Kabelführung
- Befestigungsklemme mit 5 - 60 mm Öffnung
- Arbeitsradius: ca. 1 m

90 LEDs

49,95
LL ZD-6015 LED

Jetzt bestellen! **www.reichelt.de**

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

Aktueller Katalog!
Kostenlos – Jetzt anfordern!

reichelt elektronik
Ihr kompetenter Partner für
Bauelemente • Stromversorgung • Messtechnik • Werkstattbedarf
Haus- & Sicherheitstechnik • Netzwerk- & PC-Technik • Sat-/TV-Technik

Für Verbraucher: Es gelten die gesetzlichen Widerrufsregelungen. Alle angegebenen Preise in € inklusive der gesetzlichen MwSt., ab Lager Sande, zzgl. Versandspesen für den gesamten Warenkorb. Es gelten ausschließlich unsere AGB (unter www.reichelt.de/agb, im Katalog oder auf Anforderung). Zwischenverkauf vorbehalten. Alle Produktnamen und Logos sind Eigentum der jeweiligen Hersteller. Abbildungen ähnlich. Druckfehler, Irrtümer und Preisänderungen vorbehalten. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (HRA 200654 Oldenburg) • Preisstand: 01.08.2013 • Tagesaktuelle Preise: www.reichelt.de