

**DSP-Kurs (2):  
Assembler**

G3078 E Juni 2011

(D) € 7,40 • CHF 14,30 • (A, B, L) € 8,20

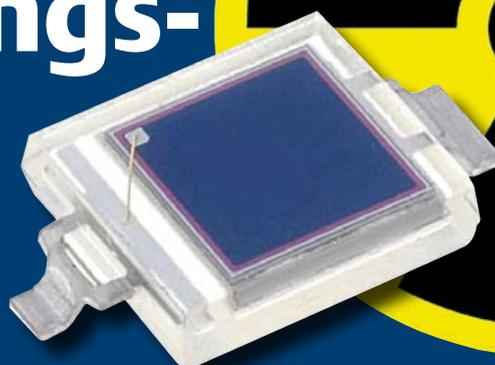
**+ USB-Protokoll unter der Lupe**

# elektor

[www.elektor.de](http://www.elektor.de)

## DIY Strahlungs- Detektor

Fotodiode misst  
Radioaktivität



## Android-Apps selbst entwickeln

Für  $\mu$ C-Board,  
Smartphone & Tablet



- + ElektorBus: Platinen für eigene Anwendungen**
- + ATM18-Projekt: GPS-Tracking per SMS und E-Mail**

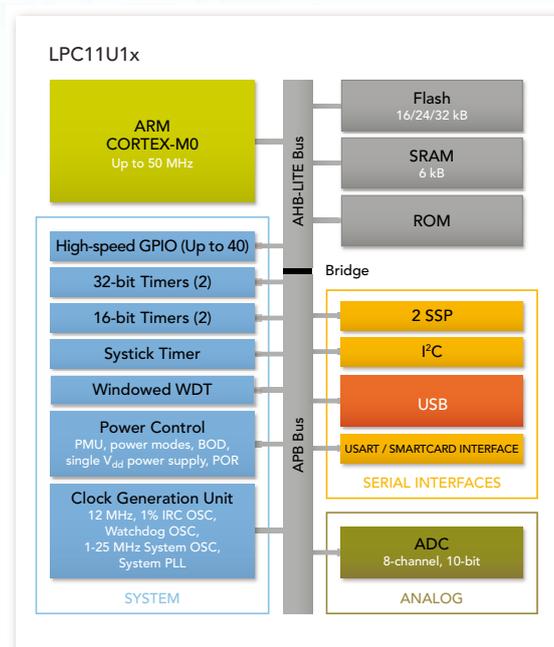


# LPC11U00

## Low cost Cortex-M0 USB solution



- ▶ SmartCard interface
- ▶ Small form-factor mobile and consumer applications
- ▶ Migration path to Cortex-M3
- ▶ Better approach to USB with configurable buffer management
- ▶ Up to 32K Flash and 6K RAM (2K USB SRAM)



**mikroC**  
**PRO for PIC**

**dsPIC**

**PIC32**

**überallhin**

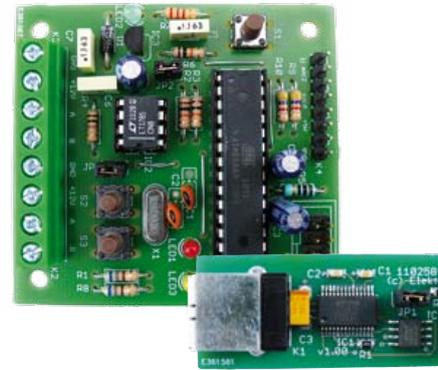
Wenn Sie mehr Rechenleistung benötigen, wechseln Sie einfach von einem PIC oder dsPIC zu einem PIC32. Wir haben genau auf eine Kompatibilität bei über 500 Bibliotheksfunktionen geachtet, so dass Sie den Code buchstäblich mit Copy & Paste übernehmen können. Ein paar kleine Anpassungen noch – und das Programm läuft!

## Trends

Als Elektor-Redakteur ist man ja der erste Mensch auf Erden, der eine neue Ausgabe von vorne bis hinten und Wort für Wort durchliest (natürlich wünschen wir uns, dass uns dabei noch möglichst viele Leser folgen werden!). Beim Bearbeiten der Übersetzungen und dem letzten Korrekturdurchgang fiel mir diesmal auf, dass ein Großteil der Seiten auf den Themenbereich Software entfiel – ohne dass wir das bei der Planung des Heftes dezidiert beabsichtigt hatten. Fünf Artikel beschäftigen sich mit verschiedenen Bus- und Kommunikationsprotokollen, die zur Fernsteuerung, Fernmessung und Positionsbestimmung genutzt werden (elektor-typisch dürfen Sie bei vier dieser Artikel praktische Anwendungen und freie Software erwarten!). Auch interessant: Bei mehreren Projekten verlassen wir in dieser Ausgabe die gewohnte Sphäre der 8-bit-AVR-/PIC-Controller und wenden uns leistungsstärkeren Prozessoren zu. Und noch ein Trend: Bei gleich zwei Artikeln werden diese Prozessoren auf Linux-Basis programmiert (wobei wir nach dem Motto „Learning by Doing“ alles so erklären, dass es auch von Einsteigern nachvollzogen werden kann). Gut zu Elektor passt dabei natürlich auch, dass vom Betriebssystem bis über die Entwicklungs-Tools alles open source und damit gratis ist. Meiner Meinung nach spiegeln wir damit recht gut momentane Entwicklungen in der Elektronik wieder – vor allem im Bereich „Messen, Steuern, Regeln“, der bei unseren Lesern nachweislich besonders beliebt ist. Ob Hausautomation, Modelleisenbahn oder Robotik: Ich könnte mir vorstellen, dass in Zukunft noch viel mehr mit schnelleren und komplexeren Controllern gesteuert wird, auf denen ein Open-source-Betriebssystem (wie Linux oder Android) läuft. Im Zuge dessen wird für Profis wie für Hobbyisten gelten, dass beim Entwickeln eigener Anwendungen noch mehr als bisher programmiert (und noch weniger gelötet) werden wird. Glücklicherweise aber stirbt die „handfeste“ Elektronik nicht aus. Dass ganz traditionelle Bauteile zu erstaunlichen Leistungen fähig sind, beweisen wir mit unserem Selbstbau-Strahlendetektor auf Seite 26!

Jens Nickel

- 6 Impressum**  
Who is who bei Elektor
- 8 Mailbox & News**  
Leserbriefe, Nachrichten und Termine
- 12 Elektor OSPV1:**  
**Open Source Projekt auf zwei Rädern**  
Der ElektorWheelie bekommt einen kleinen Bruder: Der leichter gebaute, zusammenlegbare OSPV1 wurde für den Indoor-Betrieb entwickelt, Konstruktion und Software sind open source.
- 14 Hier kommt der Bus (6)**  
Mit ersten Elektor-Platinen (Experimental-Knoten, USB/RS485-Konverter) rufen wir alle Leser zum Mit-Entwickeln auf!
- 20 Apps für Android entwickeln**  
Wir zeigen, dass die Entwicklung mit frei erhältlicher Software nahezu kinderleicht sein kann.
- 26 Fotodiode misst Gammastrahlung**  
Bei unserem Halbleiter-Detektor lässt sich sogar die Energie der einzelnen Teilchen erkennen.
- 32 LED-Fahrradlampe**  
Unsere Hochleistungs-LED-Lampe ist natürlich nicht nur für Drahtesel geeignet!
- 36 Positionsbestimmung mit ATM18**  
Das Modem-Modul GM862-GPS von Telit enthält einen GPS-Receiver. Kombiniert mit einem Elektor-ATM18-Modul hat man ein GPS-Tracking-Gerät, das man zum Beispiel ins Auto einbauen kann. Der Eigentümer wird dann jederzeit via E-Mail oder SMS über den aktuellen Aufenthaltsort des Fahrzeugs informiert.
- 43 Labcenter**
  - Platinenlayout: Kleinheiten und Feinheiten!
  - Scirocco im Kleinformat
  - Zurück zur Erde
  - Radmitnehmerschrauben beim ElektorWheelie



# INHALT

**42. Jahrgang**  
**Juni 2011**  
**Nr. 486**

## 14 Hier kommt der Bus (6) Elektor Experimental-Platine und mehr!

Mit ersten Elektor-Platinen zu Entwicklungszwecken erreichen wir nun die nächste Station unseres Busses. Zur nachbausicheren Hardware gehören ein Experimental-Knoten mit analogen und digitalen Eingängen sowie ein kompakter USB/RS485-Konverter. Darüber hinaus stellen wir ein einfaches System vor, das eine effiziente und sichere Bus-Kommunikation gewährleistet.



## 20 Apps für Android entwickeln Für PC, BeagleBoard, Handy oder Tablet

Ende 2008 erschien das erste Smartphone mit dem Betriebssystem Android – inzwischen kommen jedes Jahr geschätzt 350.000 Android-Handys hinzu. Ein Grund für diesen phänomenalen Erfolg könnte sein, dass Android „open source“ ist. Hier wollen wir zeigen, wie man an dieser Erfolgsgeschichte mit einer eigenen App teilhaben kann. Dabei beschränken wir uns aber nicht auf Smartphones, sondern stellen auch ein Mikrocontrollerboard vor, auf dem das Betriebssystem laufen kann.



## 26 Fotodiode misst Gammastrahlen Strahlungsdetektor mit BPW34

Wenn es um die Messung von Radioaktivität geht, denkt man zuerst an Geiger-Müller-Zähler. Aber Zählrohre sind rar und teuer geworden, und wenn man eines bekommt, muss man auch noch für eine Betriebsspannung von einigen 100 V sorgen. Weniger bekannt ist, dass eine ganz normale Fotodiode wie die BPW34 auch Röntgen- und Gammastrahlen detektiert.



## 32 LED-Fahrradlampe Hightech für Feldwege

Käufliche und in der Straßenverkehrsordnung zugelassene Fahrradlampen verwenden ein Halogenbirnchen mit maximal 2,4 W. Auf dunklen Wegen fernab der Zivilisation kann man damit nicht wirklich etwas sehen. Die hier vorgestellte Lampe ist in Deutschland zwar nicht für den Straßenverkehr zulässig, dafür bringt sie es aber mit vier Hochleistungs-LEDs auf etwa 600 Lumen und lässt sich außerdem auf Knopfdruck dimmen.



### 47 Brutkasten mit Mikrocontroller

Die Nachahmung eines brütenden Huhns ist nicht ganz trivial!

### 50 Audio-DSP-Kurs (2): Programmierung

Der DSP56374 von Freescale wird vorzugsweise in Assembler-Sprache programmiert, die leicht zu erlernen ist.

### 56 Inside USB: Wie das Protokoll funktioniert

Für die Kommunikation zwischen PC und angeschlossenen Geräten ist der Universal Serial Bus der unumstrittene Standard. Wir stellen die Grundzüge des Protokolls vor.

### 60 Entwicklung mit dem XPort Pro Modul

Lantronix hat einen Mikrocontroller in eine Ethernet-RJ-Buchse integriert. Anhand von drei Anwendungen demonstrieren wir, was in dem Modul steckt.

### 66 Mini-Projekt: Blaulicht für Modellautos

Nur wenige Bauteile genügen für ein blaues Wechsellicht!

### 68 Review: PC-Oszilloskop der Mittelklasse

Die Reihe PicoScope 3000 des Herstellers Pico Technology umfasst leistungsstarke, am USB-Bus betriebene Oszilloskope.

### 72 E-blocks: RC5 in Flowcode

Mit den E-blocks kann man eigene Projekte dazu bringen, RC5-codierte Infrarot-Signale zu verstehen oder zu generieren.

### 76 Hexadoku Sudoku für Elektroniker

### 78 Retronik Die bemerkenswerte Vorgeschichte der LED

### 84 Vorschau Nächsten Monat in Elektor

Eine multimediale und interaktive Plattform für jeden Elektroniker - das bietet Elektor International Media. Ob Anfänger oder Fortgeschrittener, ob Student oder Professor, ob engagierter Profi oder leidenschaftlicher Hobbyist: Hier finden Sie wertvolle Informationen, Inspiration für die eigenen Entwicklungen, Unterstützung bei der Ausbildung und nicht zuletzt eine gute Portion Unterhaltung. Gedruckt und im Web. Analog und digital. In Theorie und Praxis.



**ANALOG • DIGITAL  
EMBEDDED • MIKROCONTROLLER  
AUDIO • MESSTECHNIK**

**IMPRESSUM**

42. Jahrgang, Nr. 486 Juni 2010  
Erscheinungsweise: 11 x jährlich (inkl. Doppelheft Juli/August)

Elektor möchte Menschen anregen, sich die Elektronik zu Eigen zu machen – durch die Präsentation von Projekten und das Aufzeigen von Entwicklungen in der Elektronik und technischen Informatik.

Elektor erscheint auch in Englisch, Französisch, Niederländisch, Spanisch und weiteren Sprachen. ELEKTOR ist in über 50 Ländern erhältlich.

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](mailto:redaktion@elektor.de) zu richten.

Internationale Chefredaktion **Wisse Hettinga**

Redaktion Elektor Deutschland  
**Ernst Krempelsauer** (Chefredakteur, v.i.S.d.P.)  
**Jens Nickel**  
(E-Mail: [redaktion@elektor.de](mailto:redaktion@elektor.de))

Internationale Redaktion  
**Harry Baggen, Thijs Beckers, Jan Buiting, Eduardo Corral, Clemens Valens**

Redaktionssekretariat **Hedwig Hennekens**

Labor/Technische Redaktion  
**Christian Vossen (Ltg.), Thijs Beckers, Ton Giesberts, Luc Lemmens, Jan Visser**

Grafische Gestaltung und Layout  
**Giel Dols, Mart Schroyen**



# Elektor Proton Robot



Offene Roboter-Plattform  
für Schule, Ausbildung & Freizeit

**NEU!**

Der Elektor Proton Robot ist eine neue vielseitige Lehr- und Experimentierplattform für den Bau von Robotern in Schule, Ausbildung und Freizeit. Dabei kann man zwischen vielen Sensoren und Aktoren sowie auch zwischen den beiden Mikrocontrollern PIC16F887 und ATmega32 wählen. Mit einer Experimentierplatine und etwas Lötarbeit

kann man auch auf einen anderen Mikrocontroller zurückgreifen, wenn er ein Dutzend I/O-Leitungen, einige analoge Eingänge, einen I<sup>2</sup>C-Bus und eine RS232-Schnittstelle zur Verfügung stellt.

Der Proton Robot lädt geradezu zur Weiterentwicklung mit eigenen Ideen ein. Auf der Hauptplatine befinden sich diverse freie I/O-, A/D-, RS232, SPI- und I<sup>2</sup>C-Steckverbinder, und viel Platz für Einbauten gibt es ebenfalls. Steuern Sie den Proton Robot mit Ihrem Handy, über WLAN oder Bluetooth, lassen Sie mehrere Roboter miteinander kommunizieren, montieren Sie Kameras und Mikrofone für die Bild- und Spracherkennung! Der Fantasie sind keine Grenzen gesetzt...

#### Features:

- Ultraschall-Entfernungsmesser
- Keramischer Schallwandler
- 3 Infrarot-Sensoren
- Motortreiber-Modul
- 3 Lichtsensoren zur Linienverfolgung
- 2 Fototransistoren
- 2 Servomotoren
- LCD (4 x 20 Zeichen)
- 2 Drucktaster
- Audio-Modul
- Greifer
- 10 RGB-LEDs (für Augen)
- 8 LEDs (für Mund)

**Erhältlich als Bausatz  
oder Fertigmodul!**

- **Komplettbausatz inkl. Rumpf, Kopf, Greifer, Audio-Modul und Controller (nach Wahl) für 1.249,00 €\***
- **Komplett aufgebaute und getestete Proton Robot inkl. Rumpf, Kopf, Greifer, Audio-Modul und Controller (nach Wahl) für 1.699,00 €\***

\* Preis inkl. 19% MwSt., zzgl. 20,00 € Versandkosten innerhalb der EU (einschließlich Schweiz)

Weitere Infos & Bestellung unter

[www.elektor.de/proton](http://www.elektor.de/proton)

Geschäftsführer/Herausgeber: Paul Snackers

Marketing/Vertrieb (Leitung): Carlo van Nistelrooy

Anzeigen (verantwortlich): Irmgard Ditzgen  
ID Medienservice

Tel. 05 11/61 65 95-0 - Fax 05 11/61 65 95-55

E-Mail: [service@id-medianservice.de](mailto:service@id-medianservice.de)

Es gilt die Anzeigenpreisliste Nr. 40 ab 01.01.2010

Vertriebsgesellschaft: IPS Pressevertrieb GmbH

Postfach 12 11, 53334 Meckenheim

Tel. 0 22 25/88 01-0 - Fax 0 22 25/88 01-199

E-Mail: [elektor@ips-pressevertrieb.de](mailto:elektor@ips-pressevertrieb.de)  
Internet: [www.ips-pressevertrieb.de](http://www.ips-pressevertrieb.de)

Vertrieb Österreich

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

Der Herausgeber ist nicht verpflichtet, unverlangt eingesandte Manuskripte oder Geräte zurückzusenden. Auch wird für diese Gegenstände keine Haftung übernommen. Nimmt der Herausgeber einen Beitrag zur Veröffentlichung an, so erwirbt er gleichzeitig das Nachdruckrecht für alle ausländischen Ausgaben inklusive Lizenzen. Die in dieser Zeitschrift veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen einschließlich Platinen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Die veröffentlichten Schaltungen können unter Patent-

oder Gebrauchsmusterschutz stehen. Herstellen, Feilhalten, Inverkehrbringen und gewerblicher Gebrauch der Beiträge sind nur mit Zustimmung des Verlages und ggf. des Schutzrechtsinhabers zulässig. Nur der private Gebrauch ist frei. Bei den benutzten Warenbezeichnungen kann es sich um geschützte Warenzeichen handeln, die nur mit Zustimmung ihrer Inhaber warenzeichengemäß benutzt werden dürfen. Die geltenden gesetzlichen Bestimmungen hinsichtlich Bau, Erwerb und Betrieb von Send- 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.

© 2011 elektor international media b.v.

Druck: Senefelder Misset, Doetinchem (NL)



ISSN 0932-5468

**Zeitnah**

Mailbox-Beitrag zum „Blauen Globus“ (Website des Zeitschriftenhandels), Elektor 05/2011, S. 8

In der Mailbox meiner heute aus der Zukunft (Mai) zu mir gekommenen Elektor-Ausgabe findet sich der Satz „... die Lücken sollen zeitnah geschlossen werden“. Es gibt wunderbar viele deutsche Worte, die Ereignisse der Zukunft zuordnen sollen: bald, sofort, gleich, rasch, so schnell wie möglich, nächste Woche und so weiter.

Zeitnah ist ein meiner Meinung nach ein guter Kandidat für das Unwort des Jahres! Und um bei der Zeit zu bleiben: Können Sie mir erklären, warum die aktuelle Ausgabe,

die wahrscheinlich im Februar und März geschrieben, Anfang April gedruckt wurde und am 19. April schon beim Leser war, als Mai-Ausgabe bezeichnet wird? Auch das ist ein zeitgeistiger Unsinn!

Christoph Resele



*Ich gebe Ihnen zeitnah Recht: Diese neuzeitliche Wortschöpfung wird zur Bereicherung der deutschen Sprache nicht unbedingt gebraucht. Aber Sprache lebt, und dazu gehören auch solche Modewörter, die kommen und gehen und manchmal auch bleibenden Eingang in unseren Wortschatz finden.*

*Als Kandidat für das „Unwort des Jahres“ erscheint mir das Wörtchen „zeitnah“ trotz des ihm innewohnenden Unsinn als weniger aussichtsreich, da vergleichsweise harmlos. Aber darüber müsste die Jury der „Sprachkritischen Aktion Unwort des Jahres“ entscheiden, vorausgesetzt, Sie senden Ihren Vorschlag zeitnah an die E-Mail-Adresse [vorschlaege@unwort-desjahres.net](mailto:vorschlaege@unwort-desjahres.net).*

*Ich möchte zu unserer Ehrenrettung noch dar-*

*auf hinweisen, dass dieses Wort in einem als Zitat wiedergegebenen Text enthalten war – die Elektor-Redaktion würde so etwas natürlich niemals schreiben. ;-)*

*Die Vordatierung der Monatsangabe auf Zeitschriften ist dem Zeitgeist des vorigen Jahrhunderts zu verdanken. Ich kann mich daran erinnern, dass mir diese Praxis schon in den 1960er-Jahren bei US-amerikanischen Zeitschriften aufgefallen ist. Es dauerte dann nicht sehr lange, bis auch in Europa Verlage damit begonnen haben, den Erscheinungstermin vorzuverlegen, um mit der „neueren“ Ausgabe am Kiosk zu punkten. Und wenn einer damit anfängt, dauert es auch nicht lange, bis andere nachziehen, um keinen Nachteil gegenüber der vorgepreschten Konkurrenz zu riskieren. Mit dem Ergebnis, dass Monatszeitschriften schon lange nicht mehr zum Monatsanfang, sondern fast zur Mitte des Vormonats erscheinen.*

Ernst Krempelsauer

**Und ewig leuchtet die Nixie-Uhr...**

Kleine Nixie-Kunde, Elektor 05/2011, S. 32 (090518)

Das Lesen des Artikels über Nixieröhren hat mich dazu veranlasst, mal über mein Projekt von 1974 zu schreiben.

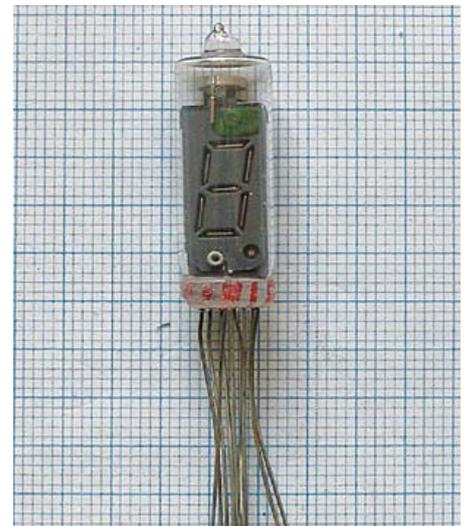
In den 70er Jahren gab es in Elektor ein Uhrenprojekt mit dem IC MM5314. Das wurde von mir weiterentwickelt und mit 7-Segment-Nixieröhren bestückt, einmal mit vier und einmal mit sechs Röhren.

Das Uhrchen läuft seither ununterbrochen, also schon mehr als 35 Jahre. Kein Ausfall bisher, selbst die Transistoren haben durchgehalten. Es waren sogar welche aus einem Sonderangebot und selektiert nach Sperrspannung. Der Bau der Uhr wurde noch dadurch erschwert, dass das IC MM5314 die Ausgänge multiplext. Das bedeutet, dass auch die Anoden der Nixieröhren geschaltet werden mussten. Das Besondere ist, dass es keine normalen Nixieröhren mit 10 Eingängen sind - es sind 7-Segment-Nixieröhren!

Erstaunlicherweise ist die Anzeige nach dieser langen Betriebszeit nicht dunkler geworden, während eine mit LEDs bestückte Uhr schon so dunkel geworden ist, dass man sie nicht mehr gebrauchen kann.

Es gibt bei mir auch eine Uhr mit einer 7-Segment-Glühfadenanzeige, die immer noch läuft. Auch diese Anzeige ist im Laufe der Zeit nicht dunkler geworden – im Gegensatz zu einer Uhr mit einer 7-Segment-Minitron-Anzeige. Das ist zwar auch eine Glühfadenanzeige, die aber sehr schnell unbrauchbar geworden ist.

Uwe Kardel (Leser seit 1971)



*Die Lebensdauer so mancher Projekte aus den ersten 10 Elektor-Jahren (ab 1970) ist immer wieder faszinierend – besonders, wenn man die Elkos sieht...*



*Was auch beeindruckt: Elektor-Projekte wurden offenbar schon immer als Anregung für eigene Entwicklungen verwendet.*

*Die unterschiedliche Lebensdauer der Glühfaden-Displays ist wohl auslegungsbedingt. Die Zahl der Betriebsstunden hängt bei Glühfäden immer stark von der Betriebsspannung ab. Eine 230-V-Glühbirne ist an 115 V zwar nicht besonders hell, sie hält aber fast ewig.*

**MailBox**

In dieser Rubrik veröffentlichen wir Kritik, Meinungen, Anregungen, Wünsche oder Fragen unserer Leser. Die Redaktion trifft die Auswahl und behält sich Kürzungen vor. Bitte geben Sie immer an, auf welchen Artikel und welche Ausgabe (Monat/Jahr) sich Ihr Schreiben oder Ihre Mail bezieht.

Sie erreichen uns per E-Mail [redaktion@elektor.de](mailto:redaktion@elektor.de), per Fax (02 41/88 909-77) oder unter der Anschrift: Redaktion Elektor Susterfeldstr. 25 – 52072 Aachen

## JUNI

**7. - 9.** Nürnberg  
**Sensor + Test**

Sensoren sowie Stand-Alone- und PC-basierte Mess- und Prüftechnik für alle denkbaren Anwendungen sind auf diesem Messe-Klassiker zu sehen.

[www.sensor-test.de](http://www.sensor-test.de)

**8.** Holzwickede (bei Dortmund)  
**ESD Seminar**

Kostenloses Seminar von EBV Elektronik: Wie erstellt man eine Schaltung in Hinsicht auf den ESD-Schutz? Wie lässt sich der ESD-Schutz von Produkten optimieren?

[www.ebv.com/en/events.html](http://www.ebv.com/en/events.html)

**8. - 10.** München  
**Intersolar**

Diese bekannte Messe hat sowohl die Solarthermie als auch die Photovoltaik zum Thema.

[www.intersolar.de](http://www.intersolar.de)

**16.** Dortmund  
**LabVIEW Hands-On**

Kostenloses Praxisseminar, in dem man erste Schritte mit der grafischen Entwicklungsumgebung LabVIEW unternehmen kann. Plus weitere Termine, siehe Website!

[www.ni.com/seminars/europe.htm](http://www.ni.com/seminars/europe.htm)

**20.** Dortmund  
**Elektor-Seminar  
„FPGA - Der Logikbaukasten für Jedermann“**

Ein schneller und umfassender Einstieg in die FPGA-Thematik.

Neben der Theorie kommt auch die Praxis nicht zu kurz.

[www.elektor.de/fpga-seminar](http://www.elektor.de/fpga-seminar)

**24. - 26.** Friedrichshafen  
**Ham Radio / Hamtronic**  
Messe-Paradies für  
Amateurfunker!

[www.hamradio-friedrichshafen.de](http://www.hamradio-friedrichshafen.de)

**28. - 30.** Frankfurt am Main  
**Large-area, Organic & Printed  
Electronics Convention**

Konferenz mit begleitender Ausstellung. Chancen, Produkte und Entwicklungen in der organischen und gedruckten Elektronik sind das Thema.

[www.lope-c.com](http://www.lope-c.com)

**29.** Zürich  
**Elektor-Seminar  
„Eagle PCB und Design“**  
Vom Schaltungsdesign bis zur  
Platinenfertigung reicht die  
Themenpalette. Dazu gibt's  
Tipps und Tricks zum Um-  
gang mit dem bekannten  
Elektronik-CAD-Programm.

[www.elektor.de/eagle-seminar](http://www.elektor.de/eagle-seminar)

## JULI

**5. - 6.** Ludwigsburg  
**Entwicklerforum Embedded-  
System-Entwicklung**

Das zweitägige Forum wird von der Zeitschrift Design & Elektronik veranstaltet. Am ersten Tag geht's um Hardware und Sicherheit, Tag zwei widmet sich der Embedded Software.

[www.embedded-entwicklerforum.de](http://www.embedded-entwicklerforum.de)

**13. - 14.** München  
**Hightech  
auf dem Olympiaturm**

Das von Meilhaus initiierte Messtechnik-Event mit Vorträgen und einer Ausstellung findet jedes Jahr statt. Kostenlos nach Anmeldung!

[www.olyturm.com](http://www.olyturm.com)

**18. - 19.** Stuttgart  
**Technische Dokumentation  
mit Microsoft Word**

Das zweitägige Seminar für Entwickler beschäftigt sich mit Format- und Dokumentvorlagen,

Inhaltsverzeichnissen, referenzierten Bildern und vielem mehr.

[www.vdi-wissensforum.de/fileadmin/pdf/02SE024030.pdf](http://www.vdi-wissensforum.de/fileadmin/pdf/02SE024030.pdf)

**19. - 20.** München  
**Batterien im Fahrzeug**

Funktionsprinzipien elektrochemischer Speichersysteme, Batteriemangement, Messverfahren und Sicherheit sind nur einige der Themen dieses VDI-Seminars.

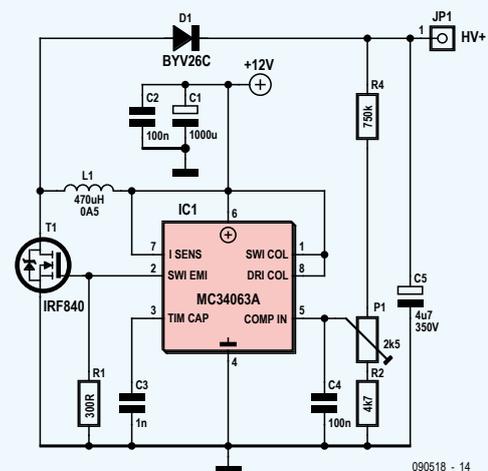
[www.vdi-wissensforum.de/fileadmin/pdf/01SE09006.pdf](http://www.vdi-wissensforum.de/fileadmin/pdf/01SE09006.pdf)

## Updates und Ergänzungen

**Kleine Nixie-Kunde –  
Aufbau und Ansteuerung von Nixie-Röhren**

Elektor 05/2011, S. 32 (090518)

In der Schaltung des Boost-Converters (Bild 7 auf S. 36) ist beim Reinzeichnen die Freilaufdiode D1 (BYV26C) verschwunden. Hier der berichtigte Schaltplan:



090518 - 14

Anzeige

**EURO**  
CIRCUITS

**Europas Leiterplatten-Referenz für  
Prototypen & Kleinserien**

[www.eurocircuits.de](http://www.eurocircuits.de)

Von Phil Knurhahn

### Verträgt Elektronik radioaktive Strahlung?

Dieser Frage sind die Redakteure der Zeitschrift „IEEE Spectrum“ im Zusammenhang mit der Katastrophe im Kernkraftwerk Fukushima Dai-ichi nachgegangen. Sie haben sich mit einem Spezialisten der Vanderbilt University in Nashville (Tennessee) unterhalten: Professor Dan Fleetwood leitet dort eine Arbeitsgruppe auf dem Gebiet „Strahlungseffekte“ mit Schwerpunkt in der Elektronik. Seine Äußerungen zusammenfassend kann man sagen, dass hochenergetische Strahlung und auch Neutronen Atome ionisieren können und damit die Kristallstruktur des Siliziums zerstören. Die fertigen Chips selbst aber werden eher deshalb ausfallen, weil Leckströme auftreten, die durch die Aufladung der Isolation zwischen den leitenden Strukturen der Chips entstehen. In den klassischen CMOS-Strukturen schießen Gammastrahlen und Röntgenstrahlen die Elektronen aus den Atomen der Isolatoren und bilden dabei Elektronen-Löcher-Paare. Dadurch entsteht eine positive Ladung, die die Arbeitsbedingungen für den Chip verändert. Bis zu einem gewissen Grad kann man die Isolatoren in den Chips „härten“, damit sie weniger Defekte bei Strahlungseinflüssen zeigen. Grundsätzlich aber verträgt Elektronik eine viel höhere Strahlenbelastung als der Mensch – problemlos bis in den Bereich um die 5 Sievert pro Jahr (zum Vergleich: für den Menschen gelten Werte von 100 Millisievert pro Jahr bereits als kritisch). Spezielle Bausteine, die für hohe Strahlenbelastung ausgelegt sind, vertragen aber noch höhere Dosen – bis zu 1000 Sievert.

<http://eecs.vanderbilt.edu/researchgroups/radiationeffects>

### Unsinnigste Maschine

Der jährlich verliehene Rube-Goldberg-Preis motiviert Studentengruppen, sich vielleicht ein einziges Mal in ihrem Leben mit einer unsinnigen Maschine zu beschäftigen – Hauptsache, sie funktioniert. Der Wettbewerb verlangt die Entwicklung einer Maschine, die „möglichst effektiv hohe Komplexität und Ineffizienz miteinander verbindet“. Dafür sind mindestens 20 Einzelschritte erforderlich und die Aufgabe muss in höchstens zwei Minuten erfüllt werden – ohne Eingriffe von außen. Den diesjährigen Preis gewann ein Team

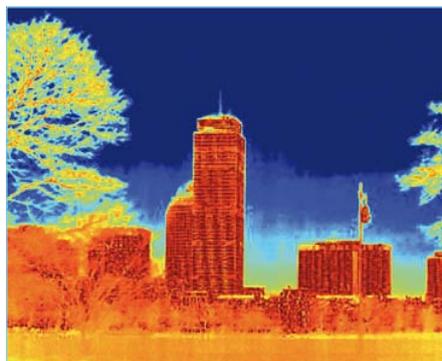


der University of Wisconsin-Stout unter Andrew Behnke (Bild: Purdue University, Mark Simons). Die Aufgabe war, ein Gerät zu bauen, das eine Pflanze gießt – und dafür möglichst viele Einzelschritte benötigt. Die Sieger-Maschine brauchte 135 Einzelschritte.

[http://en.wikipedia.org/wiki/Rube\\_Goldberg\\_Machine\\_Contest](http://en.wikipedia.org/wiki/Rube_Goldberg_Machine_Contest)

### Schnellanalyse fürs Energiesparen

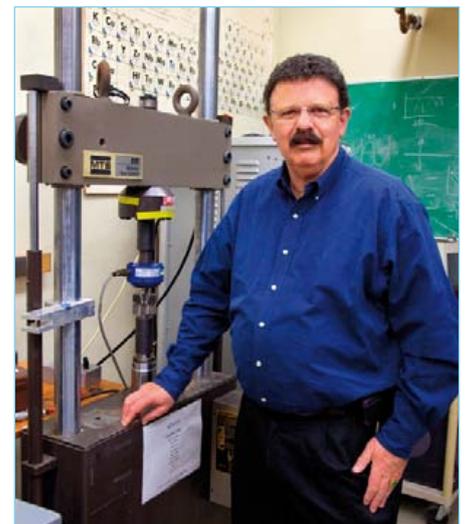
Die bisher genutzten Verfahren zur Überprüfung der Wärmeverluste an Häusern sind zeitaufwändig, wenngleich aber meist lohnend. Am MIT in Cambridge (Massachusetts, USA) wurde jetzt ein mobiles Verfahren erarbeitet, das – ähnlich wie bei Googles „Streetview“ – ganze Straßen abfährt und dabei die Häuser reihenweise erfasst. Das liefert natürlich zunächst einmal nur Frontbilder – aber gibt damit auch erste Hinweise, wo eine detaillierte Analyse sinnvoll ist (Bild: Long N. Phan, MIT). Die Forscher Long Phan und Jonathan Jesneck, die zum Lehrstuhl von Professor Sanjay Sarma (Field Intelligence Laboratory) gehören, präsentierten ihre Ergebnisse auf der MIT Energy Conference. Innerhalb weniger Monate haben sie die gesamte Stadt Cambridge erfasst (ein einminütiges Video dazu kann man auf der Website sehen).



Die vordringlich zu sanierenden Gebäude bzw. Gebäudeteile (wie z.B. Dächer) fallen durch ihre Farbgebung auf – von gelb über rot bis zu violett. Dies Scanverfahren kann eine gründliche Infrarotanalyse nicht ersetzen, aber doch Prioritäten setzen. Entscheidend war, dass man statt der üblicherweise genutzten hochauflösenden Infrarotkameras – sie kosten bis zu 40.000 US-\$ – eine neue patentierte Lösung mit vielen und preiswerteren kleinen IR-Kameras geschaffen hat, die das Bild mosaikartig zusammensetzen. Die billigeren Kameras sind schon für weniger als 1.000 \$ zu bekommen und liefern dann ein hoch auflösendes Gesamtbild. Genutzt wird das schon in der Raumfahrt bei Robotern auf dem Mars.

<http://web.mit.edu/newsoffice/2011/ir-scanning-energy-0316.html>

### Metallisches Glas



Professor Robert Ritchie, der an der Berkeley University Materialwissenschaften lehrt, hat einen neuen Werkstoff erfunden: Metallisches Glas. Er und seine Mitarbeiter glauben, dass dies mit Abstand das stabilste Material ist, was heute zur Verfügung stehen würde – und dass man damit zum Beispiel Flugzeuge bauen könnte oder sogar lange Brücken. Ein Video zeigt, wie man am Institut versucht, das Material zum Brechen zu bringen. Auf dem Foto ist Robert Ritchie im Labor zu sehen (Foto: Roy Kaltschmidt, Berkeley Lab Public Affairs).

[www.smartplanet.com/people/video/metallic-glass-the-strongest-toughest-material-yet/6196423/?tag=video](http://www.smartplanet.com/people/video/metallic-glass-the-strongest-toughest-material-yet/6196423/?tag=video)  
<http://newscenter.lbl.gov/news-releases/2011/01/10/new-glass-tops-steel/>

## Günstige 8-bit-PICs mit Segment-LCD-Ansteuerung

Microchip erweitert seine 8-bit-PIC16-Mikrocontroller-Familie um fünf weitere Mitglieder. Die MCUs mit der Typenbezeichnung PIC16LF190x sind in der herstellereigenen XLP-Technologie (eXtreme Low-Power) ausgeführt und eignen sich insbesondere für die Implementation von niedrigpreisigen und energiesparenden LCD-Anwendungen. Hierzu gehören Smartcards, medizinische Geräte, Haushaltsgeräte, Schlüsselanhänger und andere Anwendungen mit segmentierten LCDs. Aufgrund der XLP-Technologie liegt der Ruhestrom nahe bei 20 nA und der typische Arbeitsstrom beträgt 35 Mikroampere pro MHz.

Zum Funktionsumfang der PIC16LF190x-



Familie gehören bis zu 14 KB Flash-Programmspeicher, bis zu 512 Bytes RAM und bis zu 14 ADC-Kanäle mit 10 bit Auflösung sowie auch serielle Kommunikation und die Ansteuerung von bis zu 116 LCD-Segmenten. Die MCUs verfügen zur Einhaltung der Quarzgenauigkeit und zur Unterstützung der Low-Power-Echtzeituhr über einen integrierten Temperaturfühler. Weiterhin sind sie mit einer Unterspannungserkennung mittels internem ADC und einer Spannungsreferenz ausgestattet. Zur Evaluierung der neuen MCUs stehen Eval-Boards und Debugger zur Ver-

Nachrichten aus Forschung und Technik, interessante Produkt-Neuheiten und vieles mehr findet man aktuell unter [www.elektor.de](http://www.elektor.de)!

fügung. Die Controller sind in SMD- und DIP-Gehäusen erhältlich.

[www.microchip.com/lcd](http://www.microchip.com/lcd)

## Power-Module mit bis zu 10 A

National Semiconductor stellt zwölf neue „Simple-Switcher“-Power-Module vor, die für einen Ausgangsstrom von bis zu 10 A ausgelegt sind. Die Module kombinieren die Effizienz eines synchronen Schaltreglers mit der Einfachheit eines Linearreglers. Sie kommen ohne externe Induktivität aus und sind hinsichtlich des Layouts und der Platzierung der Bauelemente nicht so komplex wie es für Schaltregler-Designs typisch ist. Die neuen Power-Module können die Lastströme durch Parallelschaltung gleichmäßig untereinander verteilen und sind synchronisierbar, es lassen sich somit Ausgangsströme bis zu 60 A realisieren. Durch die Synchronisation mehrerer Module mit derselben Taktfrequenz ist es möglich, in sensiblen Systemen die Schaltstörungen auf ein Minimum zu begrenzen.

Die neuen Module sind 1:1 pin-kompatibel zu anderen Mitgliedern der Familie. Sechs der zwölf neuen Module können an Eingangsspannungen zwischen 6 V und 36 V betrieben werden und erzeugen eine einstellbare, präzise geregelte Ausgangsspannung zwi-

schen 0,8 und 6 V. Die zweite, ebenfalls sechs Produkte umfassende Gruppe ist für Eingangsspannungen von 6 bis 20 V konzipiert. Muster und ein Evaluation Board lassen sich unter der unten angegebenen Webseite bestellen.

[www.national.com/switcher\\_power\\_modules](http://www.national.com/switcher_power_modules)

## Neuer Katalog von Reichelt

Der bekannte Distributor Reichelt Elektronik hat einen neuen Katalog herausgebracht. Auf 1084 Seiten werden mehr als 30.000 Artikel aus den Bereichen Elektronik und IT angeboten.

Titelangebot ist eine Steckdosenleiste mit integriertem Webserver, mit der Geräte über das Internet oder ein eigenes PC-Netzwerk ferngeschaltet werden können (Preis 99,95 Euro). Darüber hinaus überwacht die Steckdosenleiste LAN NETIO-230B den korrekten Betriebszustand pingfähiger Geräte; bei Bedarf werden diese neu gestartet. Speziell Elektroniker dürften sich auch für das programmierbare Labornetzgerät interessieren, das für 155 Euro angeboten wird. Die Spannung kann zwischen 1 und 36 Volt geregelt werden (Strom 0..10 A). Das Gerät besitzt drei programmierbare Speicher, weitere 20 Presets lassen sich über den USB-Port mit einem Computer programmieren. Der neue Katalog ist im Webshop kostenlos erhältlich. Dort ist auch ein elektronischer „Blätterkatalog“ mit direkter Verlinkung zu den Produkten des Shops zu finden.

[www.reichelt.de](http://www.reichelt.de)



**SIMPLE SWITCHER® Power Modules**

- Current Sharing up to 60A
- CISPR 22 (Class B) EMI Standard Compliant
- Frequency Synchronization

National Semiconductor

**PCBs Muuuuch Cheaper...**

**16,94 EURO\***

5 pcs, 100 mm x 100 mm

\*per piece, incl. VAT (21%)

+ shipping costs e. g. Germany 10,71 EURO

**Jackaltac.com**

www.jackaltac.com

No-frills policy

# NEU

# ELEKTOR OSPV<sup>1</sup>

## Open Source Projekt auf zwei Rädern

### Ein Indoor-Roller

Letztlich entscheiden Sie (und Verkehrsregeln Ihres Landes), wo Sie mit dem OSPV<sup>1</sup> unterwegs sind. Konstruiert wurde er für den Einsatz in Gebäuden, wegen der zwei Zentimeter Bodenfreiheit sind unebenes Pflaster, Bordsteinschwellen und Schlaglöcher nicht sein Metier. Der Abstand zwischen den Rädern ist mit 14 cm so bemessen, dass Sie das Gleichgewicht spielend halten können, solange der Untergrund eben ist. In Fabrikgebäuden, Sporthallen, Schulen und natürlich in unseren Labor- und Verlagsräumen huscht der OSPV<sup>1</sup> wie ein Wiesel dahin. Die Füße des rollenden Fahrers stehen neben den Rädern. Das Aufsteigen und Losfahren ist leichter als beim Fahrrad, der niedrige Schwerpunkt des OSPV<sup>1</sup> vermittelt ein sicheres Gefühl.

### Leicht steuerbar

Sobald Ihre Füße auf den Trittflächen stehen, müssen Sie sich leicht nach vorn oder hinten neigen, damit der OSPV<sup>1</sup> anfährt. Beim ElektorWheelie wird zum Lenken die Haltestange seitwärts bewegt, der OSPV<sup>1</sup> wird dagegen mit einem kurzen Steuerknüppel gelenkt. Auch der OSPV<sup>1</sup> reagiert sofort und auf der Stelle, Sie können sogar tänzerische Figuren wie enge Kreise und Pirouetten drehen. Auf dem Tanzparkett, da sind wir sicher, würden Sie mit dem OSPV<sup>1</sup> eine gute Figur machen! Wenn Sie den OSPV<sup>1</sup> einschalten, protokolliert er seinen Neigungswinkel und benutzt diesen Wert als Referenz. Bitte achten Sie darauf, dass die Trittflächen in diesem Moment waagrecht stehen.

### Schnell zusammengelegt

Der OSPV<sup>1</sup> lässt sich schnell und bequem zerlegen, so dass er auf der Auto-Rückbank oder im Kofferraum transportiert werden kann. Er ist leicht, nur 25 kg bringt er auf die Waage, auch das „schwache“ Geschlecht kann ihn mühelos anheben und verstauen. Die Haltestange ist teilbar, das Lösen der Flügelmutter genügt, um den Platzbedarf zu reduzieren. Wir sind sicher, dass nicht nur dieses Detail die Kreativität der rollernden Community herausfordern wird. Vielleicht haben Sie eine Idee für eine noch einfachere oder intelligentere Konstruktion? Der OSPV<sup>1</sup> ist für innovative Ideen wie geschaffen!

Mit unserem ElektorWheelie sind inzwischen viele Menschen unterwegs, die Technik des balancierenden Fortbewegungsmittels fasziniert Jung und Alt. Hinter dem neuen OSPV<sup>1</sup> stehen die gleichen Ideen, Unterschiede gibt es jedoch in der Umsetzung. Der leichter gebaute, zusammenlegbare OSPV<sup>1</sup> wurde für den Indoor-Betrieb entwickelt, Konstruktion und Software sind open source. Auch diesem innovativen Einachser sind staunende Blicke sicher!



### Wichtige Daten

Gewicht:	25 kg	Bodenfreiheit Trittflächen:	2 cm
Höhe:	120 cm	Höhe Trittflächen:	5,6 cm
Breite:	47 cm	Breite zwischen Trittflächen:	29,5 cm
Tiefe:	47 cm	Reichweite:	8 km
Belastung:	90 kg	Wendekreis:	0 m

Wer ist nicht auf der Suche nach Balance im Leben? Fortschrittliche Elektronik macht das mühelos möglich. Vor zwei Jahren startete der ElektorWheelie, unser erster balancierender Elektroroller. Jetzt stellen wir unseren jüngsten Spross vor. Er bekam den klangvollen Namen *Open Source People Vehicle* <sup>1</sup>, kurz OSPV<sup>1</sup>.



## Alles Open Source

Nichts steht Modifikationen und Weiterentwicklungen im Weg, Mechanik, Hard- und Software, alles ist Open Source. Die Elektronik stimmt mit dem ElektorWheelie überein, so dass Programm-Anpassungen ein zu eins übertragbar sind. Da der Elektor-Wheelie schon eine Weile unterwegs ist, konnten Wheelie-Fahrer schon viele gute Ideen in die Tat umsetzen. Die Schaltungen, die Platinen und der Quellcode des Mikrocontroller-Programms warten darauf, von der Elektor-Website [1] heruntergeladen zu werden. Der OSPV<sup>1</sup> kommt einschließlich leistungsstarker Akkus auf dem Versandweg zu Ihnen ins Haus.

## Er kann sich sehen lassen

Was sagen die Menschen, an denen unser innovativer Elektro-Flitzer vorüberrollt? Wir sind nur staunenden und zustimmenden Reaktionen begegnet. „Ein tolles Ding!“, das hörten wir immer wieder. Der OSPV<sup>1</sup> ist ein ebenso ungewöhnliches wie attraktives Zweirad, nicht nur sein Kern, sondern auch die Schale!

## Und er rollt und rollt

Der OSPV<sup>1</sup> wurde ersonnen, um Personen zu befördern, doch dabei muss es nicht bleiben. Wenn ein Aktenstapel in einen fernen Gebäudeteil verbracht werden muss, ist er als elektrische Schubkarre nutzbar. Einen Karton mit Büchern transportiert er ebenso wie einen schweren Müllsack. Der Grundsatz des Open Source gibt Ihnen die Freiheit, individuelle Modifikationen vorzunehmen und den OSPV<sup>1</sup> als Ihren persönlichen Diener zu engagieren. Unser neuer Elektro-Roller fühlt sich überall dort wohl, wo der Boden eben ist. Fabrikhallen, Flughafen-Terminals, Verwaltungsgebäude, Schulen, Universitäten oder die eigene vielzimmerige Villa, das alles kommt ihm gelegen. Darf der OSPV<sup>1</sup> auch auf Straßen und Bürgersteigen fahren? Das hängt ganz von den örtlichen Vorschriften und Gesetzen ab. Natürlich darf der OSPV<sup>1</sup> nicht überladen werden, doch wenn Sie nicht schwerer als ungefähr 85 kg sind, wird er Sie mit Leichtigkeit tragen.

## Sie können ihn haben!

Wir liefern den OSPV<sup>1</sup> exklusiv über den Elektor-Webshop aus. Wie auch Sie stolzer OSPV<sup>1</sup>-Fahrer werden, darüber geben wir Ihnen gern auf unserer Website, [www.elektor.de/ospv1](http://www.elektor.de/ospv1), Auskunft.

Geschwindigkeit:	15 km/h	Antrieb:	Zahnriemen
Räder:	PU, Ø 14 cm	Ladegerät:	multinorm
Stromquelle:	Zwei Bleigel-Akkus 12 V, 9 Ah	Ladezeit:	ca. 2,5 Stunden
Motoren:	DC, 2 · 250 W		

# Hier kommt der Bus (6)

## Elektor Experimental-Platinen und mehr!

Mit ersten Elektor-Platinen zu Entwicklungszwecken erreichen wir nun die nächste Station unseres Busses. Zur nachbausicheren Hardware gehören ein Experimental-Knoten mit analogen und digitalen Eingängen sowie ein kompakter USB/RS485-Konverter. Darüber hinaus stellen wir ein einfaches System vor, das eine effiziente und sichere Bus-Kommunikation gewährleistet.

Von Jens Nickel

Von ersten Erfolgen angespornt, setzen wir nun nach: In diesem Teil der Serie werden nachbausichere Elektor-Platinen vorgestellt, die sich zum Entwickeln von Bus-Anwendungen eignen.

Grundlage unseres „Experimental-Knotens“ ist die Schaltung des schon im letzten Teil vorgestellten „Test-Knotens“, der jedoch nur über eine „Test-LED“ und einen „Test-Button“ verfügte. Das genügte, um fest-

zustellen, ob Messages erfolgreich über den Bus geschickt wurden. So richtig spannend war es aber nicht, und eine sinnvolle Anwendung konnte man damit auch nicht entwickeln.

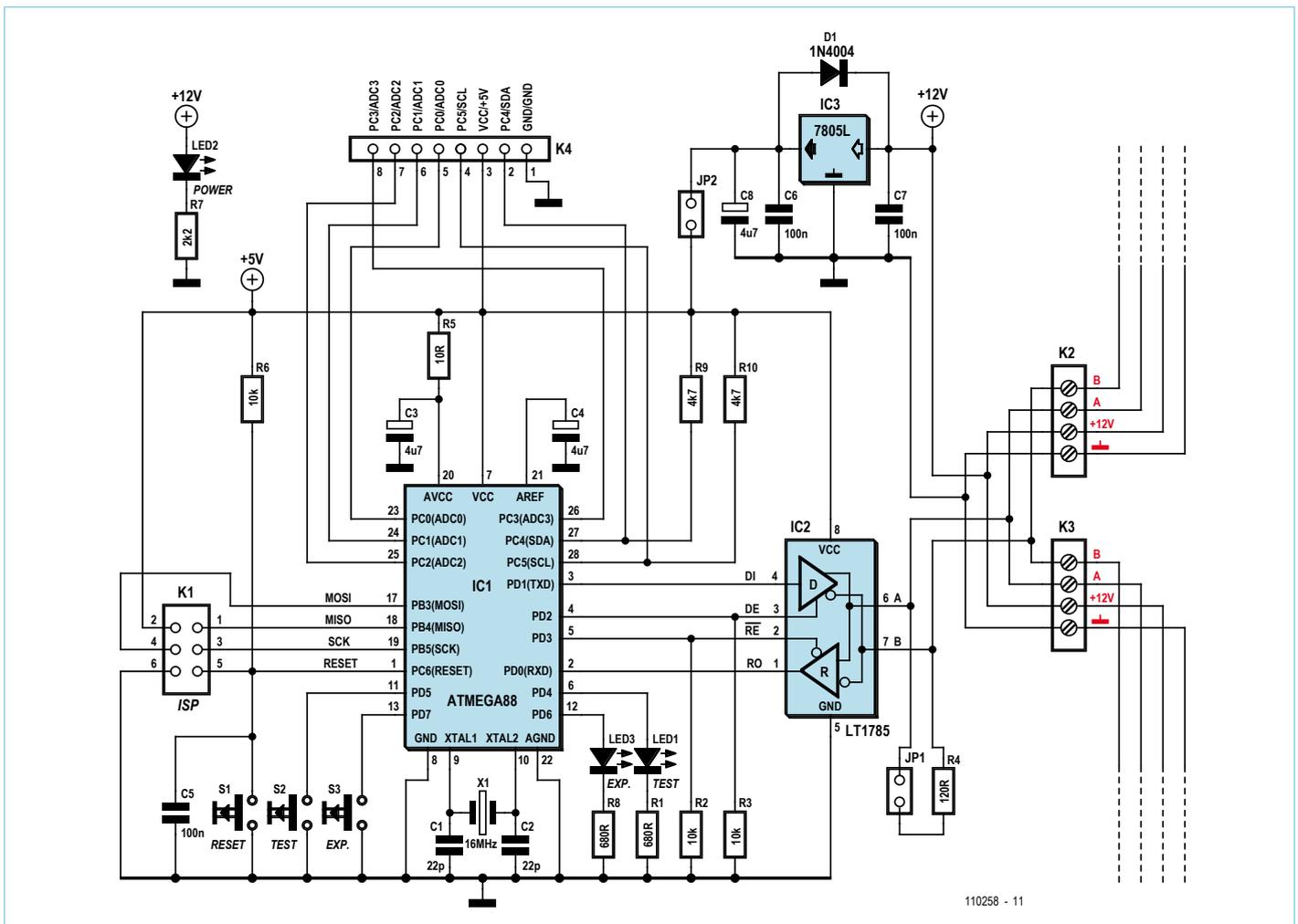
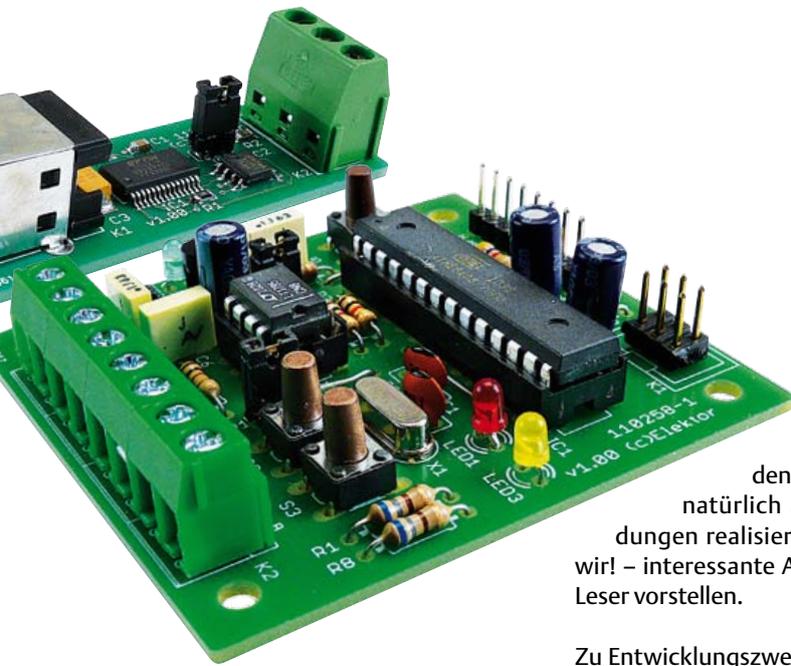


Bild 1. Schaltplan des Experimental-Knotens.



## Experimental-Knoten

Das Schaltbild des neuen ATmega-basierten Busteilnehmers sieht man in **Bild 1**. Die Funktionsweise des RS485-Treibers haben wir schon in den letzten Teilen der Serie erläutert. Über die doppelt ausgeführten Schraubklemmen schließt man den Knoten an und kann gleichzeitig alle Busleitungen durchschleifen. Als kleines Extra kam der jumperbare 120-Ω-Widerstand hinzu, hiermit lassen sich die RS485-Leitungen an beiden Ende des Busses vorschriftsgemäß terminieren.

An die Portpins PD6 und PD7 haben wir nun noch eine weitere LED und einen zusätzlichen Taster angeschlossen, den wir „Experimental-LED“ und „Experimental-Button“ nennen wollen. Die wesentliche Neuerung stellt aber der Steckverbinder K4 dar, an den sechs weitere Controllerpins, die Controller-Spannungsversorgung und die Busmasse herausgeführt sind. Die Portpins PC0 bis PC5 wurden mit Bedacht gewählt, denn diese sind – wie bei vielen Controllern üblich – gleich mit mehreren Aufgaben betraut. So lassen sich die Pins wahlweise als digitale Ein- oder Ausgänge nutzen. Über PC0 bis PC3 kann man mit Hilfe des mehrkanaligen Controller-ADCs aber auch vier analoge Spannungen messen (0..5 V). Und schließlich bieten die Pins PC4 (=SDA) und PC5 (=SCL) noch Zugang zur I2C-Schnittstelle des ATmegas. Gemäß deren Spezifikation haben wir die Leitungen SDA und SCL noch jeweils mit einem Pullup-Widerstand versehen.

An K4 lassen sich nun einfache Sensoren, I2C-Slaves (wie zum Beispiel Temperatursensoren) sowie diverse weitere Elektronik anschließen. Das Datenblatt des ATmega88 [1] verrät, wie man die Pins mit

Hilfe der Controller-Register ansteuern kann. Doch werden wir im Laufe der Serie natürlich auch eigene Anwendungen realisieren sowie – so hoffen wir! – interessante Applikationen unserer Leser vorstellen.

Zu Entwicklungszwecken kann der Experimental-Knoten auch über K4 (statt über den Bus) versorgt werden. Dann ist der Jumper JP2 zu entfernen. Im laufenden Betrieb wird aber unbedingt empfohlen, die Bus-Masse über die entsprechende Leitung weiterzugeben und diese zu erden (siehe hierzu den

## Experimental-Knoten

### Widerstände:

R1,R8 = 680 Ω  
R2,R3,R6 = 10 k  
R4 = 120 Ω  
R7 = 2k2  
R9,R10 = 4k7

### Kondensatoren:

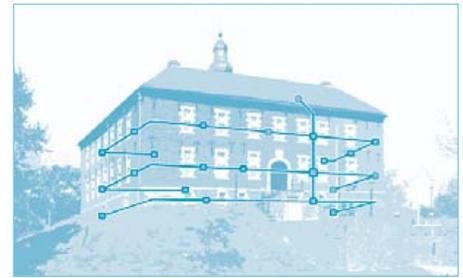
C1,C2 = 22 p  
C3,C4,C8 = 4µ7  
C5,C6,C7 = 100 n

### Halbleiter:

D1 = 1N4004  
IC1 = ATmega88-20PU  
IC2 = LT1785  
IC3 = 78L05

### Außerdem:

X1 = Quarz 16 MHz  
S1,S2,S3 = Taster  
JP1,JP2= Jumper  
LED1 = LED 3 mm, rot  
LED2 = LED 3 mm, grün  
LED3 = LED 3 mm, gelb  
K1 = 2x3-polige Stiftleiste (RM 2,54 mm)  
K4 = 1x8-polige Stiftleiste (RM 2,54 mm)



## USB/RS485-Konverter

Hier können wir uns ganz kurz halten: Die Platine des USB/RS485-Konverters entspricht dem Schaltplan des schon im letzten Teil gezeigten Konverters (siehe **Bild 3**). Weil der FTDI-Chip sowieso nur in SMD erhältlich ist, haben wir uns dazu entschlossen, gleich die ganze Platine für die SMD-Bestückung zu entwickeln und vollständig bestückt anzubieten (**Bild 4**). Meines Wissens ist dies der bisher einfachste und kompakteste USB/RS485-Konverter, der jemals in Elektor vorgestellt wurde. Er eignet sich hervorragend für eigene Bus-Entwicklungen, zum Beispiel auf Basis der in diesem und dem vorangegangenen Teil vorgestellten PC-Software. Als kleine Einschränkung

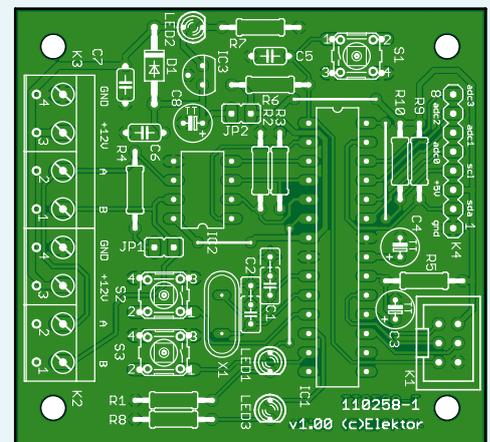


Bild 2. Die Platine ist durchweg mit konventionellen Bauteilen zu bestücken.

K2,K3 = Schraubklemme für Printmontage  
4-polig, RM 5,08 mm  
Platine 110258-1 [3]

letzten Teil der Serie [2]). Der Vollständigkeit halber sei noch erwähnt, dass wir die Controllereingänge AVCC und AREF für die ADC-Messungen nach Datenblatt beschaltet haben. Die einseitige Platine (**Bild 2**) ist einfach zu bestücken, mit den Drahtbrücken sollte man beginnen.

möchten wir anführen, dass die Entwicklung des ElektorBus-Systems noch nicht abgeschlossen ist. Im Moment können wir noch nicht sagen, ob wir später noch eine intelligentere Verbindung zwischen der RS485- und der PC-Seite benötigen werden (hierzu unten mehr). Einstweilen sind wir jedoch gut mit dem Konverter bedient, und

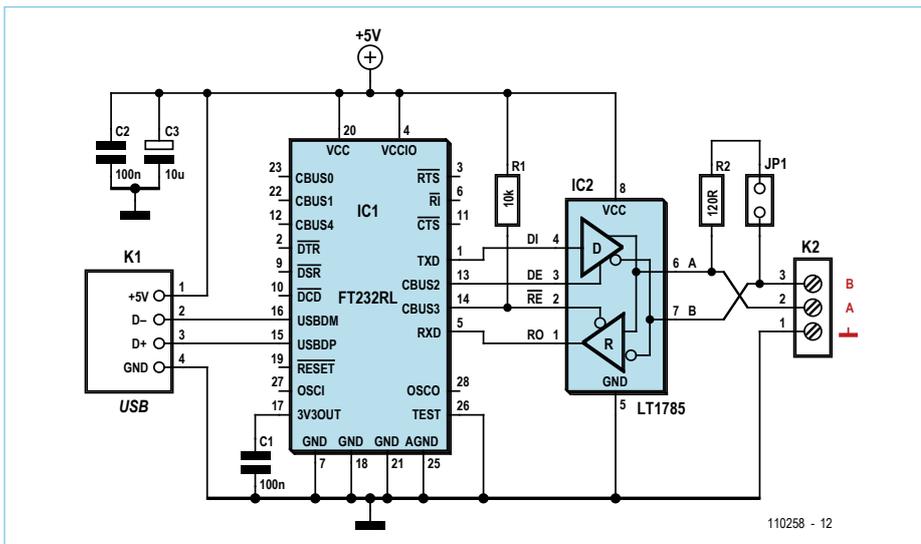


Bild 3. Schaltplan des USB/RS485-Konverters.

natürlich lässt sich dieser auch prima für andere RS485-Anwendungen verwenden.

Da uns nun Platinen für vielfältige Experimente zur Verfügung stehen, wollen wir auch in puncto Software vorankommen. Die in Folge 5 [2] vorgestellte Test-Software bot ja noch keinen Mechanismus, um Kollisionen auszuschließen oder zumindest festzustellen. Im Folgenden soll daher ein einfaches System vorgestellt werden, das eine effiziente und sichere Bus-Kommunikation gewährleistet. Aus unserer Bus-Mailing-Gruppe kamen jedoch auch alternative Vorschläge, eine kleine Zusammenfassung zeigt der Textkasten.

### Round-Robin-Polling

In den vorangegangenen Folgen der Serie hatten wir schon das Für und Wider eines

zyklischen Abfragens (Polling) aller Knoten erläutert. Jeder Knoten wird (reihum) mit genau einer 16-Byte-Elektor-Massage von einem Scheduler angesprochen und antwortet darauf mit wiederum genau einer Nachricht. Doch ein solch simples System hätte zwei gravierende Nachteile.

1. Ein Sender, der etwas mitteilen möchte, muss warten, bis er an der Reihe ist. Als Beispiel betrachten wir einen Schalter in einem Domotik-System, den ein Nutzer betätigt. Im schlimmsten Fall wird die entsprechende Message erst verschickt, wenn vorher alle anderen Busteilnehmer an der Reihe waren. Bei 9600 Baud und 32 Busteilnehmern könnte es dann über eine Sekunde dauern, bis das Licht angeht, was vor allem Gäste ziemlich verwirren dürfte.
2. Es entsteht eine Menge unnützer Kommunikation auf dem Bus. Es macht bei-

spielsweise keinen Sinn, den ganzen Tag ein Mal pro Sekunde einen Lichtschalter abzufragen, dessen Zustand sich höchstens 10 Mal am Tag ändert.

### Das andere Extrem

Bei Knoten, die recht selten senden (etwa Schalter), wäre eine andere Vorgehensweise nützlicher: Der Schalter darf dann senden, wenn er betätigt wird, ohne sich um die anderen Teilnehmer zu kümmern. Natürlich können dann Kollisionen auftreten, was bewirkt, dass eine recht willkürliche Folge von Bits auf dem Bus entsteht. Statt einer schönen 16-Byte-Nachricht kommen dann plötzlich mehr Bytes mit völlig falschen Werten an, beide Nachrichten „kommen nicht durch“. Bei wertvollen Messages muss der Empfänger deshalb mit einer Acknowledge-Message (Bestätigungs-Nachricht) antworten. Wenn diese ausbleibt, wiederholt der Sender den Vorgang. Diese ebenfalls recht simple Methode schließt übrigens auch aus, dass Nachrichten durch andere Störungen verloren gehen.

Dieses Verfahren hat aber ebenfalls Nachteile. Wenn einige Busteilnehmer im Spiel sind, die recht oft eine Nachricht senden, dann treten viele Kollisionen auf. Außerdem müsste etwa ein Auswerte-Knoten, der von einem Sensor-Knoten regelmäßig Temperaturwerte empfängt, jedes Mal mit einer Acknowledge-Message antworten, was die Buslast und die Gefahr von Kollisionen weiter erhöhen würde.

### Gute Mischung

Unabhängig voneinander hatten unser Leser Jürgen Lange und ich eine Idee: Nehmen wir doch das Beste aus beiden Welten! Wir wechseln periodisch einfach zwischen dem Polling (Knoten spricht nur, wenn er gefragt wird) und dem anderen Extrem (Knoten darf senden, ohne sich um die anderen zu kümmern).

Nachdem jene Knoten reihum abgefragt wurden, die regelmäßig etwas mitzuteilen haben (etwa Temperatursensoren), gibt der Scheduler den Bus für das Senden ohne Abfrage frei. Dann dürfen alle Busteilnehmer senden, die nur sehr selten etwas sagen (etwa Schalter). Diese Freie-Bus-Phase (Free-BusPhase) muss selbstverständlich zeitlich begrenzt werden, so dass nicht versehent-

## Stückliste USB/RS485-Konverter

#### Widerstände:

- R1 = 10 k (0805)
- R2 = 120 Ω (0805)

#### Kondensatoren:

- C1, C2 = 100 n (0805)
- C3 = 10 µ/16 V (6032)

#### Halbleiter:

- IC1 = FT232RL
- IC2 = LT1785 (SO-8)

#### Außerdem:

- JP1 = Jumper
- K1 = USB-Buchse Typ A
- K2 = Schraubklemme für Printmontage 3-polig, RM 5,08 mm

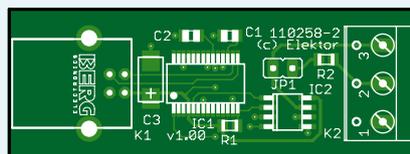


Bild 4. Der USB/RS485-Konverter wird von Elektor bestückt und getestet angeboten.

Platine 110258-2 [3]  
oder  
Platine fertig bestückt und getestet  
110258-91 [3]

## Die Alternative: Bit-Level-Lösungen

Bei der Konzeption unseres Experimental-Knotens konnte ich mein Gehalt erstmals als Schaltungsentwickler verdienen ;-). Doch das war geradezu Kinderkram gegen die Aufgabe, als Diskussionsleiter unserer Bus-Mailing-List den Überblick zu behalten. Wie schon in den letzten Teilen der Serie angeklungen: Es gibt eine Fraktion erfahrener Bus-Entwickler, die sich vehement für eine hardwarenahe Kollisionserkennung einsetzt. CAN-Experte John Dammeyer brachte eine Methode ins Spiel, bei der jeder Sender zuerst eine „Space-Time“ von 12 bits abwarten muss, bevor er eine Message senden darf. Da er gleichzeitig auf den Bus lauscht, kann er so erkennen, ob gerade ein anderer Teilnehmer spricht. Für verschiedene andere Vorschläge der Kollisionserkennung, die wir in der Runde diskutierten, muss das so genannte Bit-Banging genutzt werden (also das Manipulieren von UART-Portpins per Software).

Immer wieder nach vorne gebracht wurden auch verschiedene hardwarenahe Methoden der (Re-)Synchronisation der Busteilnehmer auf einen Message-Beginn. Elektor-Leser Walter Trojan sprach sich dafür aus, den so genannten 9-bit-Modus zu benutzen, den der ATmega88 und diverse weitere Controller beherrschen (bei Atmel Multi-processor Communication Mode genannt, siehe [1]). Hier werden in einem Frame statt 8 bit gleich 9 bit gesendet, der Empfänger kann hiermit zum Beispiel erkennen, wann ein Adressbyte und wann ein Datenbyte ankommen.

Hier können wir leider nicht alle Vor- und Nachteile der beschriebenen Methoden wiedergeben. Nur so viel: Die durchaus cleveren Ver-

fahren haben für mich einen entscheidenden Nachteil: Sie engen die Zahl der Prozessoren und Plattformen ein, die als Teilnehmer unseres Busses fungieren können. Nicht alle Controller verfügen eben über einen 9-bit-Mode, und ein Bit-Banging per PC-Software schien mir – wengleich nicht unmöglich – keine besonders elegante Methode zu sein. Und was, wenn wir unsere Messages einmal über ein anderes Netzwerk (etwa drahtlos) übermitteln wollten? Bytes sind Informationseinheiten, mit denen praktisch jedes System umgehen kann, bei 9-bit-Worten und einzelnen Bits wurde es da schon schwieriger.

Sie können sich denken, dass die Diskussion nun erst so richtig in Fahrt kam... Vor allem, dass ich mir die Möglichkeit vorbehalten wollte, unseren RS485-Bus direkt von der PC-Software aus an zu steuern, trug mir Schelte ein. Ein Windows-PC sei wegen der Timing-Tücken des Microsoft-Betriebssystems sowieso nicht gut geeignet; vielmehr müssten wir statt eines USB/RS485-Konverters gleich ein USB/RS485-Gateway nutzen, das einen zusätzlichen Controller mitbringt, der die Buskommunikation übernehmen kann.

Um es kurz zu machen: Mir gefiel die Möglichkeit, dass unsere 16-Byte-Messages nahtlos auch über die Grenzen von Plattformen und Netzwerken laufen konnten, zu gut, als das ich sie an dieser Stelle aufgeben wollte. Um möglichst flexibel zu bleiben, wollte ich eine sichere und robuste Kommunikation nach Möglichkeit nicht hardwarenah auf Bitebene, sondern auf der Ebene der Messages sicherstellen (siehe Text).

Auch hier ist Feedback natürlich willkommen!

lich noch gesendet wird, wenn das Polling wieder beginnt.

Um das von mir *Hybrid-Mode* genannte Verfahren praktisch umzusetzen, habe ich die Software aus dem letzten Heft [2] erweitert. Zuerst wird eine Implementierung der allgemeinen Funktionen beschrieben, die man später vielleicht in eine Bibliothek packen könnte; unten folgt eine kleine Applikation. Das Ganze ist wie immer von der Elektor-Website als Quellcode downloadbar [3].

### Scheduler

Der PC übernimmt hier die Rolle des Schedulers. Letzterer besitzt grundsätzlich die

Sender-Adresse 0, wodurch dessen Nachrichten von allen Knoten erkannt werden können. In einem Array ( $0.intPolledNodesMax-1$ ) speichert der Scheduler die Adressen der Knoten, die reihum abgefragt werden. Natürlich ist es so auch möglich, einen besonders gesprächigen Knoten öfter als die anderen zu „pollen“.

Zum Polling eines Knotens sendet der Scheduler eine Abfrage-Nachricht (*SchedulerRequest*), bei der als Empfängeradresse die Adresse des abgefragten Knotens eingetragen ist. Dann wartet der Scheduler auf eine Nachricht mit der entsprechenden Senderadresse (*ResponseMessage*), die jedoch an einen beliebigen Empfänger gerichtet sein

kann. Nun kommt der nächste Knoten an die Reihe und so fort. Falls einer der Knoten ausgefallen ist, käme das Ganze natürlich zum Stillstand. Deshalb muss beim Absenden des *SchedulerRequests* ein Timer gestartet werden, der nach Ablauf veranlasst, dass ebenfalls zum nächsten abgefragten Knoten weitergezählt wird.

Der Variablen *intFreeBusCycle* kommt eine besondere Bedeutung zu. Sie bestimmt, nach wie vielen abgefragten Knoten eine *FreeBusPhase* dazwischengeschaltet wird. Eine 2 bedeutet beispielsweise, dass nach dem Polling zweier (hintereinander im Array gelisteten) Knoten ein freies Senden möglich wird (siehe **Bild 5**).

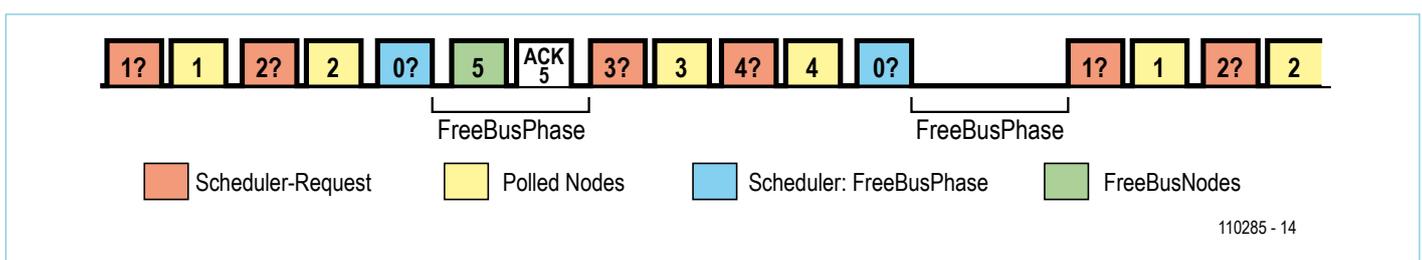


Bild 5. Beim *Hybrid-Mode* wechseln Phasen des Pollings mit freien Bus-Phasen ab. Während der *FreeBusPhase* können Kollisionen auftreten, weswegen der Empfänger mit einer Bestätigungsnachricht (ACK) antworten muss.

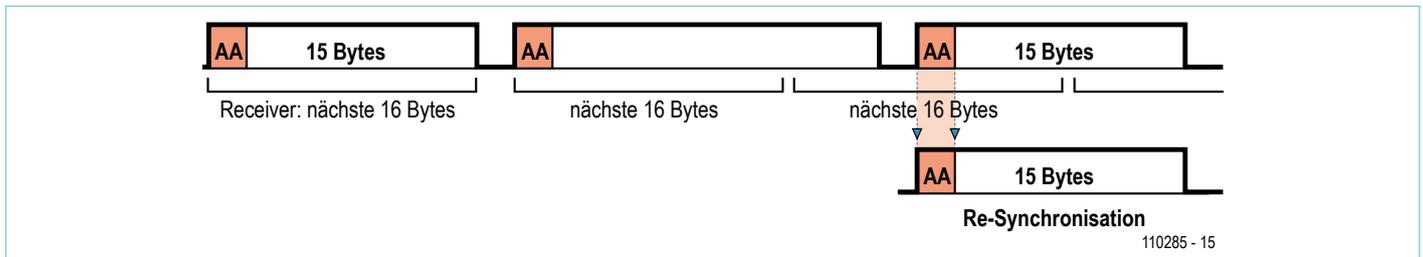


Bild 6. Bei Kollisionen können Byte-Gruppen entstehen, die länger als 16 Bytes sind. Dann müssen sich alle Busteilnehmer neu auf den Beginn der Messages synchronisieren, indem nach einem AA<sub>hex</sub> im Bytestrom gesucht wird.

Zur Einleitung der *FreeBusPhase* sendet der Scheduler eine spezielle *FreeBusMessage*, bei der als Empfängeradresse einfach 0 angegeben wird (die Datenbytes werden nicht benutzt). Danach wartet der Scheduler eine Zeit *intFreeBusTime* (Angabe in Millisekunden) ab, bis das Polling wiederaufgenommen wird.

### Firmware

Auf Seiten der Controller-Knoten wurde die BASCOM-Software aus dem letzten Teil als Grundlage genutzt. Auf Anraten von Günter Gerold aus unserer Bus-Mailing-Liste habe ich die Interruptroutine allerdings so verändert, dass sie bereits nach dem Empfang eines Zeichens zurückkehrt. Wie gehabt werden die empfangenen Zeichen in einem Byte-Array gespeichert. Wenn das 16. Zeichen empfangen wurde, wird in der Routine lediglich noch untersucht, ob die Empfängeradresse gleich der eigenen Adresse oder eine 0 (*FreeBusMessage*) ist. Daraufhin wird das *ReceivedEventFlag* gesetzt. Dies ist das Zeichen, dass eine Nachricht empfangen wurde, die in der Hauptschleife verarbeitet werden muss.

Dabei gilt, dass ein Knoten, der direkt (also mit der eigenen Adresse) vom Scheduler angesprochen wurde, immer und sofort antworten muss. Auf eine *FreeBus-*

*Message* antworten nur alle nicht abgefragten Knoten (*FreeBusNodes*), und zwar nur dann, wenn im Controller inzwischen ein Nachrichten-Ereignis aufgetreten ist (*SendEventFlag=True*). Wenn eine solche Nachricht stets sofort nach dem Eingang einer *FreeBusMessage* gesendet wird, können wir uns die Nutzung eines Timers sparen, der die Einhaltung der *FreeBusTime* im Controller überwacht.

Im EEPROM des Controllers wird abgelegt, um welchen Typ Knoten es sich handelt (*PollingStatus*). Auf diese Weise kann in beiden Fällen die gleiche Firmware genutzt werden. Beim Senden der Controller-Nachricht nutzen wir das bisher noch freie Bit des Mode-Bytes (*ElektorMessageProtocol* siehe **Bild 7**). Bit 0=0 bedeutet, dass es sich um eine *ResponseMessage* handelt, bei der prinzipiell keine Kollisionen auftreten können. Bit 0=1 bedeutet dagegen, dass wir eine Nachricht in der unsicheren *FreeBusPhase* übermitteln; der Empfänger muss daher mit einer Acknowledge-Message antworten. Bleibt diese aus, gehen wir von einer Kollision aus. Der Sender muss die Nachricht nun erneut senden, wobei mindestens die nächste *FreeBusPhase* abgewartet werden muss. Damit es dann nicht schon wieder zu einer Kollision kommt, muss ausgeschlossen werden, dass der zweite Sender ebenfalls wieder in dieser

Phase sendet. Dies erreichen wir mit einem einfachen System: Jeder Sender wartet eine unterschiedliche Anzahl *x* der *FreeBusPhases* ab. Diese Zahl *x* (*FreeBusPriority*) ist wie auch die eigene Adresse des Knotens einstweilen statisch im EEPROM abgelegt. Während der weiteren Entwicklung soll die Software aber so erweitert werden, dass die Adresse, der *PollingStatus* und auch die *FreeBusPriority* dynamisch angepasst werden können.

Im Zuge der möglichen Kollisionen können Byte-Gruppen auf dem Bus auftreten, die länger als 16 Bytes sind. Daher ist ein Mechanismus nötig, der neu auf den Beginn der nächsten Message synchronisiert. Das Problem der Re-Synchronisation haben wir in dieser Software-Version erst teilweise gelöst: Der Datenstrom wird einfach auf Bytes untersucht, die den Wert AA<sub>hex</sub> besitzen, dies wird als Start einer Nachricht angenommen (**Bild 6**). Damit ergibt sich vorerst die Einschränkung, dass ein solches Byte nicht innerhalb der übertragenen Nutzdaten vorkommen darf (auch die CRC-Fehlerkontrolle ist nun noch nicht möglich).

### Kleine Anwendung

Jetzt musste unbedingt eine kleine Anwendung her, bei der ich gleichzeitig unseren ersten Experimental-Knoten testen konnte. Um kontinuierlich abgefragte Messwerte zu

**Elektor Produkte & Service**

- Platine Experimental-Knoten, bei Elektor erhältlich (110258-1)
- USB/RS485-Konverter, bei Elektor fertig bestückt und getestet erhältlich (110258-91)
- Software-Download (gratis): Quellcode PC-Software und BASCOM-Firmware
- Software-Download, Platinen-Download und Bestellung unter [www.elektor.de/110258](http://www.elektor.de/110258)

simulieren, beschaltete ich K4 mit einem 100-kΩ-Poti (ADC0/5V/GND). Den Experimental-Knoten verband ich über den Bus mit den zwei Test-Knoten, die Günter für den letzten Teil der Serie geroutet und bestückt hatte.

Der Experimental-Knoten bekam die Adresse „02“ und einen *PollingStatus* „01“ ins EEPROM gebrannt. Die beiden Test-Knoten stattete ich mit den Adressen „01“ und „03“, einem *PollingStatus* „00“ und einer *FreeBusPriority* von „01“ bzw. „02“ aus.

Der Einfachheit halber blieb ich bei einer Firmware für alle drei Knoten. Ein Druck auf den Test-Taster setzt das *SendEventFlag*, toggelt die Test-LED und übernimmt den Status dieser LED in ein *LEDbyte*, das bei ausgehenden Nachrichten später als erstes Datenbyte in unserer Message mitgeschickt wird (Bild 7). Der *PollingStatus* bestimmt, ob zusätzlich noch der ADC0 abgefragt wird (es darf dabei nicht vergessen werden, dem ATmega88 mitzuteilen, dass er AREF als Spannungs-Referenz nutzen soll, siehe Quellcode). Die 10 bits des ADC-Ergebnisses teilen wir auf zwei Bytes auf, die später als zweites und drittes Datenbyte übermittelt werden. Da kein AA<sub>hex</sub> in den Daten vorkommen darf, nimmt das Byte *ADClow* statt der unteren 8 bit nur 7 bit, das Byte *ADChigh* die höchstwertigen 3 Bits auf. Als Empfänger der Nachrichten dient der PC, der den Status der drei LEDs und den Wert des ADCs (entsprechend zurückverwandelt) anzeigt. Um nicht mit der Scheduler-Adresse (=0) durcheinanderzukommen, wurde dem PC eine zweite Adresse (=10) zugeteilt. Von dieser zweiten Adresse aus werden dann auch die Acknowledge-Messages an die Knoten 1 und 3 verschickt (Empfängeradresse =Knotenadresse, Senderadresse=10, erstes Datenbyte=16+*LEDbyte*).

### Auf das Timing kommt es an

Nach ein paar Versuchen stellte sich eine *FreeBusTime* von 50 bis 70 ms als ausreichend heraus, wobei ich jedoch ein wenig „dirty“ gearbeitet habe. Da bei meiner kleinen Anwendung der PC gleichzeitig der Scheduler und der Empfänger der Controller-Nachrichten ist, lasse ich ihn eine Acknowledge-Message erst nach der *Free-*

*BusPhase* (asynchron) senden, erst danach werden die Knoten weiter abgefragt. In der Regel sollten das Scheduling und das Aussenden der Bestätigungs-Nachrichten jedoch synchron erfolgen. Dann müssen sowohl die Ereignis-Nachricht als auch die Acknowledge-Message in eine *FreeBusPhase* passen, so wie in Bild 5 dargestellt.

Bis die Software fehlerfrei lief, waren zugegebenermaßen mehrere Anläufe nötig. Einer der Bugs bereitete mir am meisten Kopfzerbrechen, doch schließlich konnte ich den Fehler einkreisen: In der Controllerfirmware gingen ab und zu eingehende Messages verloren. Der Grund war der folgende: Ich hatte innerhalb der Interruptroutine nicht auf die richtige Empfänger-Adresse untersucht. Wenn während der Abarbeitung der Hauptschleife (Taster, ADC) nun mehr als zwei Nachrichten über den Bus gesendet wurden, überschrieb die zweite Nachricht die Variablen, die zur Auswertung der ersten Message benötigt wurden. Man sollte daher in der Interruptroutine wenigstens auf die richtige Empfängeradresse untersuchen und die Nachricht nur dann für die weitere Verarbeitung „annehmen“. Dann hat man mehr Zeit für die eigentliche Controller-Applikation (etwa die Messwertverarbeitung). Hier wirkt sich günstig aus, dass im *Hybrid-Mode* niemals zwei Nachrichten hintereinander an denselben Empfänger verschickt werden.

### Ausblick

Dass ein PC in Personalunion Scheduler, Beteiligter und Anzeigeeinheit ist, macht die Software in der vorliegenden Version ein wenig unrund. In Zukunft könnte es einfacher sein, den Scheduler auf einem Mikrocontroller zu realisieren. Eine Möglichkeit wäre, den Controller zu nutzen, der ein noch zu entwickelndes USB/RS485-Gateway steuert (was von verschiedenen Teilnehmern unserer Bus-Mailing-List gefordert wird, siehe Kasten).

Den Mechanismus der Re-Synchronisation müssen wir noch dahingehend erweitern, dass Bytes mit dem Wert AA<sub>hex</sub> auch innerhalb der Nutzdaten und des CRC-Felds vorkommen dürfen, hierfür existieren schon Vorschläge.

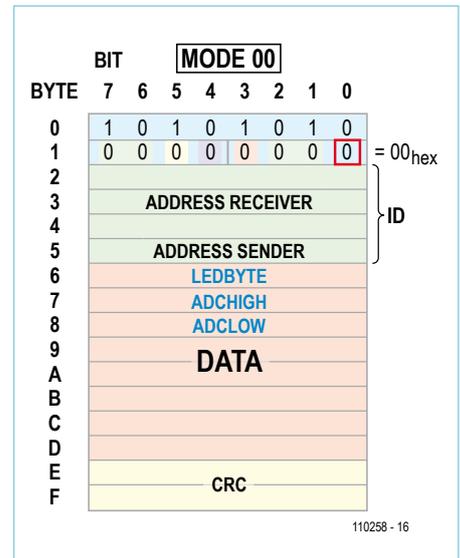


Bild 7. Unsere kleine Anwendung nutzt die ersten 3 Daten-Bytes des *ElektorMessageProtocols* (CRC noch nicht implementiert). Bit 0 des Mode-Bytes zeigt an, ob die Nachricht von einem abgefragten Knoten stammt (=0, sonst =1).

Daneben werden wir uns noch um das Anmelden eines Knotens beim Scheduler (und die dynamische Adressierung) kümmern. Last not least soll innerhalb der nächsten Folgen auch eine erste sinnvolle Anwendung entwickelt werden – wobei Sie, liebe Leser, wieder herzlich zum Mitmachen aufgerufen sind!

(110258)

Entwickeln Sie mit! Hinweise und Ideen sind willkommen unter [redaktion@elektor.de](mailto:redaktion@elektor.de)!

### Weblinks

- [1] [www.atmel.com/dyn/resources/prod\\_documents/doc2545.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf)
- [2] [www.elektor.de/110225](http://www.elektor.de/110225)
- [3] [www.elektor.de/110258](http://www.elektor.de/110258)

# Apps für Android entwickeln

Für PC,  
BeagleBoard,  
Handy oder  
Tablet



Von Clemens Valens

Ende 2008 erschien das erste Smartphone mit dem Betriebssystem Android – inzwischen kommen jedes Jahr geschätzt 350.000 Android-Handys hinzu. Ein Grund für diesen phänomenalen Erfolg könnte sein, dass Android „open source“ ist. Hier wollen wir zeigen, wie man an dieser Erfolgsgeschichte mit einer eigenen App teilhaben kann.

Dabei beschränken wir uns aber nicht auf Smartphones, sondern stellen auch ein Mikrocontrollerboard vor, auf dem das Betriebssystem laufen kann.

## Was ist Android?

Ende 2003 wurde in der kalifornischen Stadt Palo Alto die Firma Android Inc. gegründet. Ziel des Startups war die Entwicklung von Handy-Software. Genau zwei Jahre nach Gründung wurde Android von Google übernommen. Weitere zwei Jahre später wurde die „Open Handset Alliance“ ins Leben gerufen, bei der Google Mitglied ist. Diese Allianz hatte die Entwicklung von Standards für mobile Geräte zum Ziel. Ihr erstes Produkt war das Betriebssystem Android, das auf dem Linux-Kernel 2.6 basiert. In 2008 erschien dann mit dem HTC Dream das erste Handy mit Android unter der Haube.

## Android ist nicht Linux

Auch wenn Android ursprünglich auf Linux basiert, ist es mittlerweile zum eigenständigen und vollwertigen OS herangewachsen. Der Source-Code von Android ist immer noch offen, hat aber keine gemeinsame Code-Basis mit Linux mehr. Google hat bestimmte Teile von Linux modifiziert, so dass Android kein X-Windows enthält und

alle GNU-Standard-Bibliotheken fehlen. Der Transfer bestehender Programme von Linux nach Android ist deshalb nicht so einfach. Außerdem hat Google auch eigene Funktionen hinzugefügt, die speziell mit der Sicherheit mobiler Geräte zu tun haben.

Ein weiterer Unterschied zu Linux ist das andere Lizenz-Modell von Android. Linux steht unter der GPL von GNU. Für Android gilt hingegen die Apache-Lizenz der ASF (Apache Software Foundation). Diese Lizenz erlaubt im Gegensatz zur GPL kommerzielle Varianten offener Software, deren Source-Code selbst nicht veröffentlicht werden muss.

## Android ist auch kein Java

Die Programme (oder neudeutsch Apps) für Android werden zwar in Java geschrieben, aber eben gerade nicht als Java-Applikationen ausgeführt, da Android weder über eine virtuelle Java-Maschine noch über Java-Bibliotheken verfügt. Es wird lediglich die Syntax der Sprache Java benutzt. Ausgeführt werden die Apps mit der virtuellen Maschine Dalvik.

Deshalb kann man prima mit Hilfe anderer Programmiersprachen wie C oder C++ Bibliotheken für Android entwickeln. Diese Bibliotheken (analog zu DLLs unter Win-

dows) können durch Dalvik importiert und genutzt werden.

## Apps entwickeln

Im Prinzip kann jeder Apps für Android entwickeln. Alle Tools hierfür sind kostenlos verfügbar. Man benötigt nur einen Internet-Anschluss für den Download. Es ist auch kein spezieller Zugang zu einer Entwickler-Plattform erforderlich. Auf einer frei zugänglichen Seite für Android-Entwickler [1] findet sich alles Notwendige, sogar Emulatoren für Android-Hardware. Es gibt SDKs (Software Development Kits) für Windows, Linux und Mac OS X.

Obwohl Android lediglich die Java-Syntax nutzt, machen die SDKs regen Gebrauch von Java, denn es muss eine Java-Entwicklungsumgebung (JDK) installiert und eine Runtime-Umgebung für Java (JRE) vorhanden sein. Als IDE (Integrated Development Environment) wird Eclipse zusammen mit dem ADT-Plug-in (Android Development Toolkit) empfohlen.

Die Installation des SDK (samt JDK, Eclipse, ADT...) wird auf der angeführten Webseite detailliert beschrieben. Was vielleicht noch wichtig zu wissen ist, bevor man loslegt: Abhängig von der Schnelligkeit der Internetverbindung kann der Download leicht

einige Stunden in Anspruch nehmen, denn es werden mehrere Gigabyte übertragen. Wenn erst einmal alles installiert ist, kann man sich zum Eingewöhnen das Tutorial „HelloAndroid“ vorknöpfen. Alle Aspekte werden gut erklärt - bis auf die Eigenheiten der Android-Versionen. Es gibt nämlich verschiedene Versionen des Betriebssystems, die Leckereien als Code-Namen haben: 1.5 = *Cupcake*, 1.6 = *Donut*, 2.0/2.1 = *Eclair* (längliches gefülltes Gebäck aus Brandteig), 2.2 *Froyo* (gefrorener Joghurt), 2.3 *Gingerbread* (Lebkuchen) und 3.0 *Honeycomb* (lockeres Gebäck). Die Version von Android bestimmt auch darüber, auf welcher Hardware eine App läuft. Version 3 ist für Tablets gedacht. Zur Zeit ist die Version 2.2 (vor 2.1) am meisten verbreitet. Es soll mit einigen Tricks angeblich sogar möglich sein, Version 2.2 auf einem iPhone oder iPod touch zu installieren. Zum Testen benötigt man ein virtuelles Android-Gerät = AVD (**A**ndroid **V**irtual **D**evice), das mit der angepeilten Android-Version kompatibel ist. Der Start eines AVDs kann – abhängig von der Rechenleistung des PCs – ziemlich dauern. Außerdem wird ordentlich Speicher belegt. Je größer die Versionsnummer, desto träger verhält sich das jeweilige AVD. Bei einem virtuellen Tablet (also Version 3) dauerte so ein Start auf einem durchschnittlichen Laptop (Dual-Core-CPU T4200 @ 2 GHz, 2 GB RAM, Windows XP SP3) schon einige Minuten. Man muss sich also etwas gedulden!

### Programmieren für Android: Ein Kinderspiel?

Will man ohne Programmierkenntnisse Apps für Android bauen, ist sogar das nicht unmöglich. Für diese Zeitgenossen gibt es nämlich mit dem „App Inventor for Android“ [2] das passende Entwicklungssystem. Man benötigt dazu lediglich einen PC mit halbwegs modernem Betriebssystem (Windows, Linux oder OS X), eine Internet-Verbindung, einen Web-Browser und Java. Auf der obigen Webseite kann man sich darüber informieren, ob die eigene Ausstattung den Anforderungen von App Inventor entspricht. Anschließend kann man die Datei „App Inventor Setup“ mit ihren fast 100 MB downloaden und dann die Software mit Standard-Einstellungen installieren. Erst jetzt wird ein auf Android basierendes

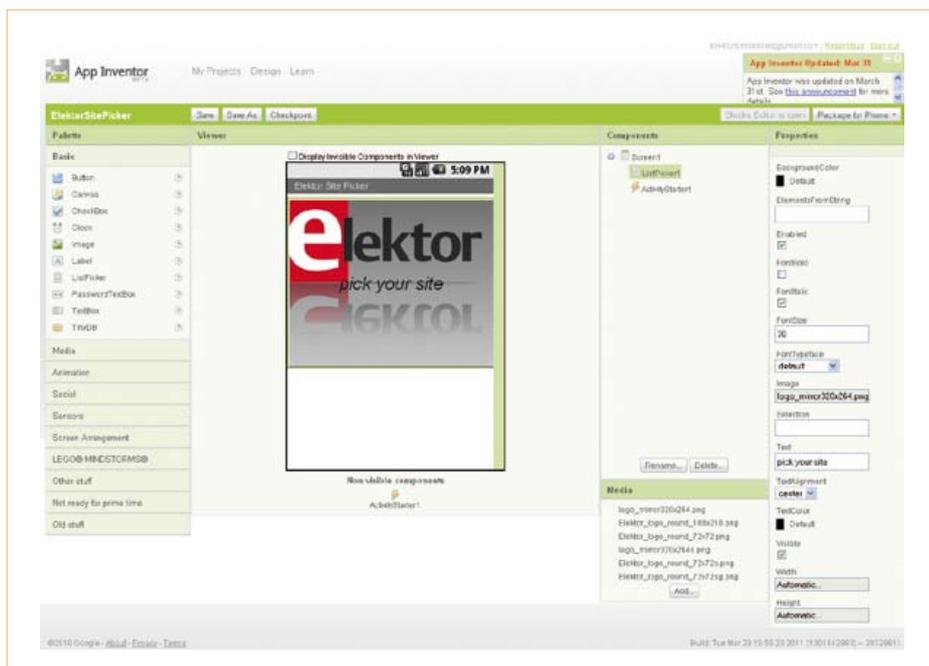


Bild 1. Android App Inventor: Designer.

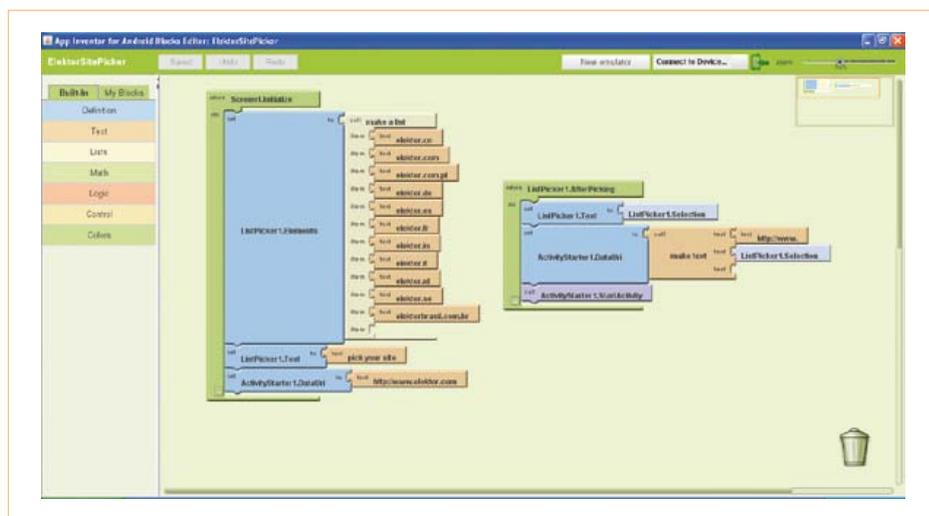


Bild 2. Die Test-App im Blocks Editor.

Gerät angeschlossen. Damit das klappt, benötigt man den passenden Treiber, den man im Internet suchen kann (falls dieser nicht schon vorinstalliert ist; so wird zum Beispiel das Smartphone „Nexus One“ automatisch erkannt).

Wenn Sie wie der Autor gar kein Android-Gadget haben, dann können Sie trotzdem mit der App-Entwicklung loslegen, denn der App Inventor bringt schon einen Emulator mit sich. Laut Google kann sich dieser Emulator einige Minuten zum Starten genehmigen, doch beim Autor war er schon nach etwa 30 s betriebsbereit.

Ob reale Hardware oder Emulator - das Vorgehen bei der Applikationsprogrammierung unterscheidet sich nicht. Zunächst wird der App Inventor geöffnet, indem man auf der Startseite auf „My Projects“ klickt (unter der Voraussetzung, dass Sie sich bereits mit Ihrem Google-Account eingeloggt haben). Es öffnet sich ein „Designer“ genanntes Fenster (Bild 1). Hier wird am sichtbaren Teil der Anwendung gearbeitet, indem man Objekte wie Knöpfe, Bilder oder Sounds auf dem Untergrund platziert, der wiederum der Oberfläche eines Android-Geräts entspricht. Durch Einstellung bestimmter Eigenschaf-

## Achtung: Fallen!

Bei der Installation von Android auf dem BeagleBoard begegnete der Autor verschiedenen Problemen. Mit dem Start des Installations-Scripts „**mkmmc-android.sh**“ ging es schon los. Es kam eine Fehlermeldung und es wurde nur die erste Partition auf der SD-Karte erstellt. In einem lichten Moment fiel dem Autor ein, dass das Script mit „grep“ nach dem Begriff „Disk“ sucht – aber sein Ubuntu war eine französische Version. Es müsste daher nach „Disque“ gesucht werden! Anstelle der Änderung des Scripts hat er die Sprache von Ubuntu geändert, um ähnliche Probleme in anderen Scripts zu vermeiden. Wenn man Scripts ändert und unter einer neuen Bezeichnung sichert, sollte man nicht vergessen, sie auch ausführbar zu machen (Rechtsklick -> properties -> permissions).

Das zweite Problem betraf die Bildausgabe. Die beim BeagleBoard mitgelieferte SD-Karte enthält die Linux-Distribution Ångström. Wenn das BeagleBoard unter diesem OS bootet, ist alles in Ordnung und es erscheint ein Bild auf dem Monitor. Versuche mit Android oder Ubuntu frustrierten mit einem schwarzen Bildschirm, obwohl das Board sonst prima funktionierte. Die Ursache hierfür war eine mit dem Monitor inkompatible Bildschirm-Auflösung. Android und Ubuntu stellten das BeagleBoard auf die etwas ungewöhnliche Ausgabe mit 1280 x 720 Pixel ein, womit der etwas ältere Bildschirm nichts anfangen konnte.

Um Problem Nr. 2 zu lösen, wird die Binär-Datei „**boot.scr**“ geändert. Hierzu wird das Tool „**mkbootscr**“ benötigt, das sich im Verzeichnis „Tools“ des SDK von TI befindet. Dieses Script produziert eine Datei „boot.scr“ für eine andere Karte. Zum Glück wird auch eine Datei „boot.cmd“ erstellt, die man mit einem Texteditor verändern kann, bevor diese in „boot.scr“ umgesetzt wird.

Hierzu ersetzt man die „**bootargs**“ in der Datei „boot.cmd“ durch diejenigen aus der Anleitung zum SDK von TI und fügt an das Ende vor dem Zeichen „.“ das Folgende hinzu: „**omapfb.mode=dvi:1024x768MR-16@60**“. Der Wert „1024x768“ kann natürlich auch durch eine andere Auflösung ersetzt werden, die der eingesetzte Monitor darstellen kann. Die neue Datei „boot.scr“ erstellt man mit dem Befehl „**mkimage -A arm -O linux -T script -C none -a 0 -e 0 -n ‚Execute ulmage.bin‘ -d boot.cmd boot.scr**“ und kopiert diese dann auf die Boot-Partition der SD-Karte.

Laut Application Note von TI wird in der Datei „boot.scr“ eine Taktfrequenz von 1 GHz (**mpurate=1000**) spezifiziert. Allerdings sind im Internet viele Berichte zu lesen, dass das BeagleBoard bei hohen Taktfrequenzen nicht sehr stabil ist. Der Autor hat dies dann auch gar nicht erst ausprobiert, sondern mit „**mpurate=800**“ gleich eine reduzierte Frequenz festgelegt.

Die funktionierende Datei „boot.scr“ des Autors ist unter [4] verfügbar.

Noch ein Hinweis zu den Benutzer-Rechten beim manuellen Präparieren der SD-Karte. Dies muss komplett mit **root**-Rechten geschehen, ansonsten wird Android nicht richtig starten und ein am seriellen Port angeschlossenes Terminal wird „Permission denied“ anzeigen. Um zum root-User unter Ubuntu zu werden, wird in die Kommandozeile der Befehl „**sudo -s**“ eingetippt. Ab da kann man die SD-Karte von dort aus auch manuell bearbeiten. Wichtig ist, dass die Partition „boot“ auf der SD-Karte eine Startpartition sein muss. Man sollte auch nicht den Befehl „**/media/rootfs# chmod -R 755 /mnt**“ vergessen, der Zugang zum Dateisystem „rootfs“ verleiht.



ten der Objekte kann das Aussehen einer Applikation den eigenen Wünschen angepasst werden. Die von der App zu erledigenden Aufgaben werden mit dem „Blocks Editor“ festgelegt. Das Programm wird nach dem Baukastenprinzip realisiert, indem man Eigenschaften und Aktionen von Objekten wie Bauelemente zusammensteckt (**Bild 2**). Diese Methode ähnelt Scratch [3], einer Programmiermethode für Kinder ab acht Jahren für den Bereich Mathematik und Informatik. Es ist also sachlich richtig, beim Programmieren unter Android von einem Kinderspiel zu sprechen, das

### Es funktioniert! Die Test-App läuft auf einem echten Android-Handy.

auch für Elektor-Leser ohne Programmiererfahrung geeignet ist. Obwohl auf diese Weise schnell kleine Apps auf die Beine gestellt werden können, kann man damit nicht alles machen und manche Dinge sind mit dieser Methode nicht unbedingt einfach zu realisieren. Außer-

dem ist App Inventor noch immer im Beta-Stadium, weshalb man mit Bugs rechnen muss. Der Autor hatte zum Beispiel größere Probleme mit dem Voice-Synthesizer, wenn die Anwendung über ein *ActivityStarter*-Objekt verfügte.

Wenn die Anwendung steht, kann sie gepackt (Package for Phone) und entweder auf den PC oder gleich auf das Android-Gerät herunter geladen werden. Wenn das Smartphone nicht mit dem PC verbunden ist, dann kann man auch einen so genannten „flash code“ (QR-Code, **Bild 3**) gene-

rieren, der einen Link auf die App enthält und damit auf das Handy geladen werden kann (sofern das Handy mit dem Flash-Code etwas anfangen kann und eine Verbindung ins Internet besteht).

Die Experimente des Autors kann man übrigens von [4] downloaden und diese in das eigene Projekt importieren.

### Hardware-Anforderungen

Ein Elektroniker überlegt sich natürlich sofort, wie er Android-fähige Hardware bauen könnte, auf der die so erstellte Software laufen würde. Die Frage ist, welchen Spezifikationen diese Hardware genügen müsste. Als realistische untere Grenze gilt ein ARM-Prozessor mit mindestens 200 MHz Takt, 32 MB RAM und 32 MB Flash-Speicher. Tatsächlich werden aber minimal 128 MB RAM und 256 MB Flash empfohlen. Und Android 3 benötigt einen Prozessor mit mindestens 1 GHz Takt plus 512 MB RAM. Weitere Voraussetzungen sind ein USB-Port, ein QVGA-TFT mit 65.536 Farben oder mehr und mindestens zehn Tasten. Nützlich aber nicht notwendig sind ein (Micro-)SD-Kartenleser, eine Kamera mit mindestens 2 Megapixel und Bluetooth.

Man kann so ein System selbstverständlich von Grund auf selbst bauen, doch ein fertiges Board ist einfacher und preiswerter. Im Internet lassen sich passende Boards finden, für die man nicht allzu viel hinblättern muss.

### BeagleBoard

Vor ein paar Jahren entwickelten einige Ingenieure von TI (Texas Instruments) eine Platine mit dem schönen Namen BeagleBoard [5], das für die OMAP-Multimedia-Prozessoren gedacht war. Das Board ist so etwas wie „Open Hardware“ und wird von der Open-Source-Community unterstützt. Obwohl sowohl Schaltung als auch Platinen-Layout kostenlos erhältlich sind, kann man sich kaum ein BeagleBoard selbst herstellen, da der RAM-Speicher mit seinen 512 MB über dem Prozessor montiert ist (PoP = Package on Package). Doch zum Glück ist das fertige Board nicht so teuer (die xM-Ausführung kostet lediglich 149 US-\$) und leicht erhältlich.

Das BeagleBoard hat eine umfangreiche Ausstattung. Auf der xM-Variante als neuester Version (**Bild 4**) befinden sich vier USB-Ports (mit Host- und Peripherie-Funktion), ein Ethernet-Port, ein RS232-Port, ein Port für USB-OTG, ein Micro-SD-Slot, ein Audio-Ein-/Ausgang in stereo, ein S-Video-Ausgang, ein HDMI-Port und Steckverbinder für LCD, die Kamera und eigene Schaltungen. Der Controller oder genauer gesagt das SoC (System-on-Chip) ist ein „Digital Media

Processor“ vom Typ DaVinci DM3730, der einen ARM-Cortex-A8-Kern mit dem DSP TMS320C64 und der Grafik-Beschleunigung SGX vereint. Der Controller selbst ist mit gängigen Betriebssystemen wie Windows CE, Linux, QNX und eben auch Android kompatibel.

Im Gegensatz zur ersten BeagleBoard-Version verfügt die xM-Variante über keinen Flash-Speicher. Das RAM kann hier aus dem Inhalt einer Micro-SD-Karte gefüllt werden, worauf sich neben Daten auch das Betriebssystem und die Applikationen befinden. Der Vorteil dieser Konfiguration ist, dass für ein anderes Betriebssystem nur die SD-Karte gewechselt werden muss. Der Nachteil ist die geringere Leistung, denn SD-Karten sind nun einmal nicht für extreme Geschwindigkeit berüchtigt. In manchen Fällen kann ein USB-Stick höheres Tempo bringen.

Das BeagleBoard ist allerdings ein sehr vielseitig einsetzbarer und leistungsfähiger Einplatinen-Computer, der sich für sehr viele Anwendungsfelder eignet. Auch wenn es nicht unbedingt die ideale Hardware für

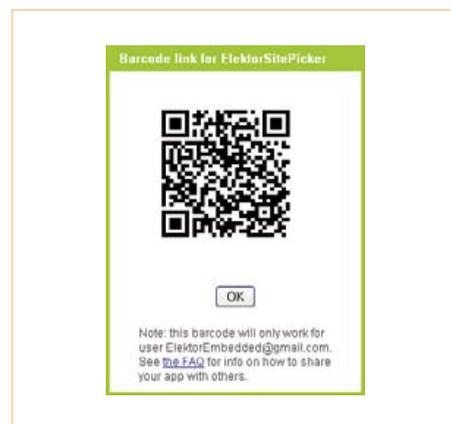


Bild 3. Der Flash-Code für den Zugriff auf die erstellte App.

Android darstellt, kann das Board recht gut ein Android-Handy oder -Tablet als Testplattform ersetzen.

Für den Einsatz des BeagleBoards benötigt man noch:

- Ein externes 5-V-Netzteil das mit 1 A belastet werden kann (unter Android genehmigt sich das Board ca. 3,5 W = 700 mA an 5 V).
- Einen Monitor mit DVI-D-Interface (obwohl das BeagleBoard über einen HDMI-Anschluss verfügt, liefert es lediglich DVI-D-Signale).

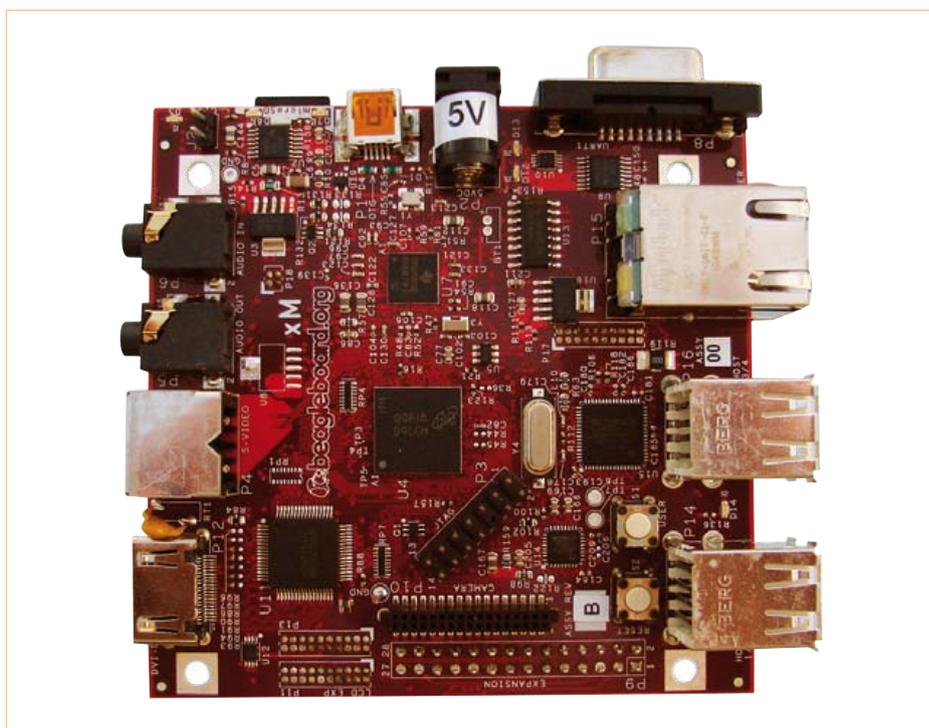


Bild 4. Das BeagleBoard-xM wird mit 5 V versorgt. Die Stromaufnahme unter Android beträgt etwa 700 mA.

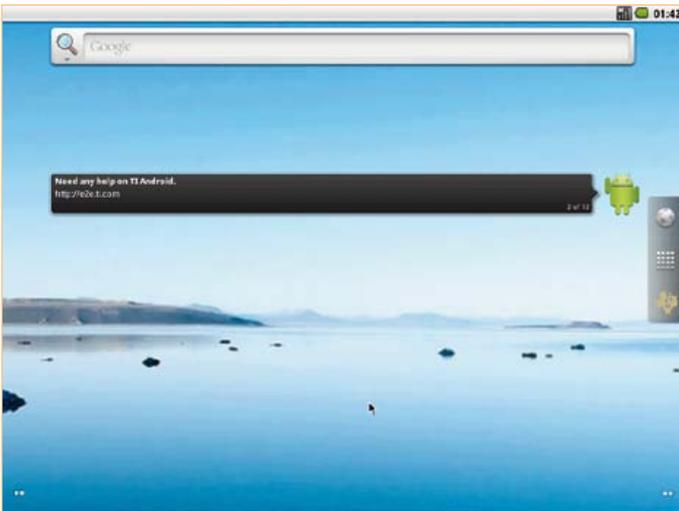


Bild 5. Der Start-Bildschirm von Android auf dem BeagleBoard.

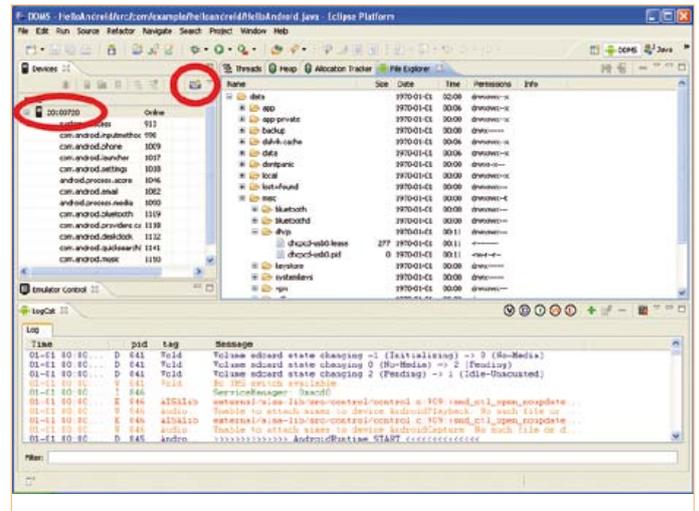


Bild 6. DDMS-Wiedergabe in Eclipse. Unser Android-Gerät nennt sich „20100720“ Mit dem Kamera-Icon können Screenshots erstellt werden.

- Einen aktiven USB-Hub (= mit eigenem Netzteil, da die Ports auf dem Board nicht genug Strom liefern können).
- Eine USB-Tastatur und eine USB-Maus, die am Hub eingesteckt werden.

Für die Installation von Android (oder Ubuntu) ist eine Micro-SD-Karte mit mindestens 4 GB Kapazität sinnvoll. Eine solche Karte ist beim BeagleBoard schon im Lieferumfang. Um die darauf befindliche Ångström-Distribution (eine Linux-Variante für embedded Systeme) nicht zu beschädigen, empfiehlt sich allerdings der Kauf einer zusätzlichen SD-Karte, am besten gleich mit 8 GB.

TI stellt zwar alle Software zur Verfügung, die für den Betrieb des Boards unter Android nötig ist, doch zum Befüllen der zusätzlichen SD-Karte ist (zumindest ein virtueller) Linux-PC mit SD-Kartenleser erforderlich. TI empfiehlt hierzu Ubuntu ab der Version 8.04. Der Autor setzte Ubuntu 10.4 ein. Nach dem Download des 1,1 GB umfassenden „TI\_Android\_FroYo\_DevKit-V2.2“ [6] kann man loslegen. Zwar ist das Procedere recht einfach, aber dennoch können immer Probleme auftreten. Werfen Sie einen Blick auf den Kasten „Achtung: Fallen!“ wenn sich die Sache anders verhält als sie soll...

Zunächst öffnet man auf dem Ubuntu-PC ein Kommando-Fenster (zum Beispiel mit Alt-F2 - dann „Run“ im Terminal aktivieren) und gibt „**sudo -s**“ ein, um **root**-Rechte zu erhalten. Die nachfolgende Prozedur ist zwar nicht absolut notwendig, denn die Karte kann auch allein mit Hilfe von „**sudo**“ präpariert werden, aber sie ist vor allem im Fall von Problemen viel eleganter. Vorab packe man das Archiv mit den Befehlen „**tar**

**-xvzf TI\_Android\_FroYo\_DevKit-V2.2.tar.gz**“ aus und wechsele mit dem Befehl „**cd**“ in das Verzeichnis „**TI\_Android\_Froyo\_DevKit-V2.2\Prebuilt\_Images\beagle-board-xml**“ des PCs. Hier startet man das Script „**mkmmc-android**“ mit dem Befehl „**./mkmmc-android.sh /dev/sd<device>**“ (falls man nicht mit root-Rechten eingeloggt ist, muss ein „**sudo**“ voran gestellt werden). Hierbei muss „**<device>**“ durch die Laufwerksbezeichnung des Kartenlesers ersetzt werden. Beim Autor war das „**c**“, weshalb der Pfad „**/dev/sdc**“ lautete. Es gibt verschiedene Wege, um die Laufwerks-Bezeichnung zu erfahren. Man kann zum Beispiel das Disk-Utility von Ubuntu (System -> Administration) verwenden. Man sollte an dieser Stelle aber Sorgfalt walten lassen, da das Android-Installations-Script das hier angegebene Laufwerk direkt formatiert!

Es kann etwas dauern, bis das Script seine Arbeit erledigt hat. Wenn alles geklappt hat, ist der Ubuntu-PC nicht mehr erforderlich. Auf der SD-Karte befinden sich nun drei Partitionen: „boot“, „rootfs“ und „data“ (mit Kapazitäten von 74 MB, 4 GB und 3,9 GB auf einer 8-GB-SD-Karte). Nun steckt man die SD-Karte in den Slot auf dem BeagleBoard und schaltet dieses ein. Der erste Boot-Vorgang dauert etwas länger (beim Autor über eine Minute), aber irgendwann erscheint dann auf dem Bildschirm die erhsehnte Meldung „Android“, gefolgt vom Begrüßungsbildschirm von Android (**Bild 5**). Es ist sehr aufschlussreich, wenn man mittels eines Terminals die Meldungen am RS232-Port des BeagleBoards (Parameter:

„**115200n81**“) mitliest, die während des Boot-Vorgangs ausgegeben werden.

Android ist ja bekanntlich ein OS, das für einen Touch-Screen optimiert ist und dessen Benutzerschnittstelle auf Fingergesten ausgerichtet ist. Da ein normaler PC-Bildschirm aber nicht berührungsempfindlich ist, muss die Maus den Finger ersetzen. Angetippt wird mit der linken Maustaste und bei gedrückter Maustaste kann man das so Angetippte verschieben; die rechte Maustaste bedeutet „zurück“. Man kann also sehr viel mit der Maus anstelle eines Fingers machen, doch das Öffnen des Menüs wollte dem Autor beispielsweise nicht wirklich gelingen. Das ist aber nicht sonderlich störend, außer wenn man den Bildschirmhintergrund ändern will. Für Letzteres gibt es allerdings auch noch andere Wege, denn das Menü kann man nämlich auch mit der Funktionstaste „**F1**“ der Tastatur öffnen. Das war allerdings die einzige Taste, die bei den Experimenten des Autors je gedrückt werden musste.

Wenn das BeagleBoard unter Android fehlerfrei gebootet hat, ist Ubuntu überflüssig und man kann den PC wieder wie gewohnt unter Windows XP etc. starten.

Nun folgt noch die Integration des BeagleBoards in das Android-SDK (wenn das SDK noch nicht installiert ist, schauen Sie bitte nochmals in den Abschnitt „Apps entwickeln“). Wie das Ganze vor sich geht, das hängt vom Betriebssystem ab. Erläuterungen findet man in der Application Note von TI. Da die meisten Anwender Windows verwenden dürften, wird das Vorgehen unter Windows beschrieben.

## Cool, das Icon taucht auf!

Nach dem Download der Datei „usb\_driver\_r03-windows.zip“ [7] entpackt man diese. In der Kommandozeile (-> Ausführen in Windows XP) gibt man ein:

```
echo 0x18D1 > „%USERPROFILE%\android\adb_usb.ini“
```

Nach dem Betätigen der Eingabe-Taste erzeugt dieser kryptische Befehl die Text-Datei „adb\_usb.ini“ mit dem Inhalt „0x18D1“ im Verzeichnis „.android“ (dies wiederum im Verzeichnis mit den persönlichen Einstellungen). Mit dem Befehl „echo %USERPROFILE%“ erhält man die genaue Pfad-Information.

Nun öffnet man mit einem Text-Editor die Datei „android\_winusb.inf“, die sich im Verzeichnis des zuvor entpackten USB-Treibers befindet, und fügt unterhalb des Headers „[Google.NTx86]“ die folgenden Zeilen hinzu:

```
; Beagle Board
%SingleAdbInterface% = USB_Install, USB\VID_18D1&PID_9018
%CompositeAdbInterface% = USB_Install, USB\VID_18D1&PID_9018&MI_01
```

Dann sichert man die Datei.

Jetzt startet man das BeagleBoard und wartet auf das Erscheinen des Begrüßungsbildschirms. Dann wird das Board mit der Mini-USB-Buchse (neben der Buchse der Stromversorgung) an einen PC angeschlossen. Jetzt wird der passende Treiber unter Windows installiert. Da das nicht automatisch passiert, muss man manuell angeben, wo sich der zu installierende Treiber befindet. Man verhilft der Pfadauswahl zu „android\_winusb.inf“ und überlässt Windows den Rest der Arbeit. Anschließend braucht der PC einen Neustart.

Wenn Windows das BeagleBoard einmal als „Android ADB Interface“ erkannt hat, kann man die Verbindung mit Hilfe der folgenden fett gedruckten Befehle in einem Kommandofenster testen:

```
adb kill-server
adb start-server
* daemon not running. starting
it now on port 5037 *
```

```
* daemon started successfully *
adb devices
List of devices attached
20100720          device
```

Laut den Reaktionen wird das Beagle-Board als ADB-Gerät mit der Bezeichnung „20100720“ erkannt. Das bedeutet, dass man das Board von nun an in App Inventor oder Eclipse als Android-Gerät verwenden kann. Unter dieser Bezeichnung taucht das Board auch dann auf, wenn man die App ausführt, die man gerade entwickelt.

Im Kommandofenster kann man nun auch Apps installieren, die man beispielsweise vom „Android Market“ herunter geladen hat (**adb install Anwendung.apk**). Man kann sie so natürlich auch de-installieren (**adb uninstall Anwendung.apk**). Das Deinstallieren klappt aber auch direkt in Android mit dem Android Application Management. Mit dem Befehl „adb shell“ erhält man einen Kommando-Bildschirm für Android, mit dem man mit üblichen Linux-Befehlen Dateien manipulieren kann.

In Eclipse kann man auch die Bildschirmanzeige (perspective) DDMS (Dalvik Debug Monitor Server) nutzen, um zum Beispiel Screenshots vom Android-Gerät zu erstellen, indem man auf ein Kamera-Icon tippt (Bild 6). Dieses Tool gibt es auch außerhalb von Eclipse. Es befindet sich unter der Bezeichnung „ddms.bat“ im Verzeichnis „tools“ der SDK-Installation.

Um das BeagleBoard unter Android mit dem Internet zu verbinden, benötigt man den Ethernet-Port, der wie die vier USB-Ports an einen auf dem Board befindlichen USB-Hub angeschlossen ist. Folglich findet man den Ethernet-Port nicht unter „eth0“



sondern unter „usb0“. In den das Netzwerk betreffenden Befehlen muss daher der typische Ausdruck „eth0“ immer durch „usb0“ ersetzt werden. Nach Verbindung des BeagleBoards mit einem Netzwerk erhält man mit dem Befehl „netcfg usb0 dhcp“ per ADB-Shell eine IP-Adresse (wenn ein DHCP-Server im Netzwerk aktiv ist). Mit „getprop net.usb0.dns1“ erhält man die IP-Adresse des DNS. Mit „setprop net.dns1 <IP-Adresse>“ teilt man Android mit, dass diese Adresse zur URL-Auflösung verwendet werden soll, und ab sofort ist das BeagleBoard im Internet!

Sie verfügen nun über alle notwendigen Informationen, um mit der Entwicklung von Apps zu beginnen. Wir sind neugierig darauf, wie lange es dauert, bis Sie ihre eigene Schaltung mit Bluetooth steuern!

(110265)

### Weblinks

- [1] <http://developer.android.com/index.html>
- [2] <http://appinventor.googlelabs.com/about/>
- [3] <http://scratch.mit.edu/>
- [4] [www.elektor.de/110265](http://www.elektor.de/110265)
- [5] <http://beagleboard.org>
- [6] [http://Software-dl.ti.com/dsp/dsp\\_public\\_sw/sdo\\_tii/TI\\_Android\\_DevKit/02\\_02\\_00/index\\_FDS.html](http://Software-dl.ti.com/dsp/dsp_public_sw/sdo_tii/TI_Android_DevKit/02_02_00/index_FDS.html)
- [7] [https://dl-ssl.google.com/android/repository/usb\\_driver\\_r03-windows.zip](https://dl-ssl.google.com/android/repository/usb_driver_r03-windows.zip)

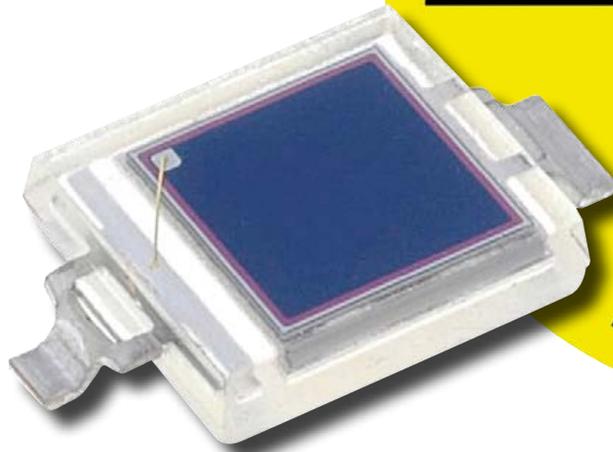
# Fotodiode misst Gammastrahlung

## Strahlungsdetektor mit BPW34

Von Burkhard Kainka (D)

Wenn es um die Messung von Radioaktivität geht, denkt man zuerst an Geiger-Müller-Zähler. Aber Zählrohre sind rar und teuer geworden, und wenn man eines bekommt, muss man auch noch für eine

Betriebsspannung von einigen 100 V sorgen. Allgemein weniger bekannt ist es, dass auch eine ganz normale Fotodiode wie die BPW34 Röntgen- und Gammastrahlen detektiert.



Ionisierende Strahlung ist potentiell gesundheitsschädlich, man sollte jede Belastung daher nach Möglichkeit vermeiden. Ein einfacher Geigerzähler mit einem kleinen Mantelzählrohr reicht allerdings kaum aus, um eine mögliche Gefährdung zu erkennen. Auch der hier beschriebene Halbleitersensor besitzt eine relativ geringe Empfindlichkeit und detektiert nur intensive Strahlungsquellen. Trotzdem eignet er sich für interessante Messungen und Experimente.

Ein Vorteil der Fotodiode ist ihre kleine empfindliche Fläche. Die Nullrate durch Höhenstrahlen ist sehr gering, und Signale kleiner Proben heben sich besser aus dem Hintergrund ab als bei einem Zählrohr.

### Strahlung

Für den Strahlenschutz am relevantesten sind Gammastrahlen. Sie durchdringen selbst dicke Wände, und es ist schwierig, sie abzuschirmen. Harte Gammastrahlen dringen ständig aus dem Weltraum zu uns vor und lassen sich auch durch dicke Wände nicht abhalten. Alpha-Strahlen dagegen haben nur eine kurze Reichweite und schaffen es meist nicht einmal durch

ein Blatt Papier. Das ist der Grund, warum die meisten Zählrohre sie nicht detektieren können. Es geht nur dann, wenn das Zählrohr ein dünnes Glimmer-Fenster besitzt. Betastrahlen haben eine größere Reichweite und durchdringen auch noch dünne Bleche. Die meisten Zählrohre erfassen hauptsächlich Gammastrahlen und sind mit Einschränkungen auch für Betastrahlung empfindlich.

### Diode als Detektor

Das Verhalten einer PIN-Fotodiode BPW34 ist ähnlich wie bei den preiswerteren Zählrohren. Alpha-Teilchen werden spätestens vom Kunststoffgehäuse der Diode abgefangen. Gammaquanten haben damit hingegen kein Problem und erzeugen eine Vielzahl von Elektronen-Loch-Paaren in der Sperrschicht. Betreibt man die Diode in Sperrrichtung, werden praktisch alle Ladungsträger abgesaugt. Es entsteht ein kleiner Stromstoß, den man verstärken und auswerten kann. Beta-Teilchen können ebenfalls ein Signal erzeugen, wenn ihre Energie groß genug ist, um bis zur Sperrschicht vorzudringen.

Die von der Fotodiode gelieferten Signalamplituden sind allerdings deutlich kleiner als bei einem Zählrohr, weshalb ein sehr rauscharmer Messverstärker benötigt wird. Eine weitere Voraussetzung für die Verwendung der Fotodiode als Detektor ist eine vollständige Abschirmung gegen Licht, weil sonst der Fotostrom das Messergebnis überdecken würde. Bei unserem Versuchsaufbau wurde dafür Aluminium-Haushaltsfolie verwendet.

PIN-Dioden besitzen im Gegensatz zu PN-Dioden eine zusätzliche, sehr schwach N-dotierte Zone. Diese hochohmig eigenleitfähige Schicht (Intrinsic-Schicht, auch I-Schicht genannt) liegt zwischen der N- und der P-Region. Im Ergebnis erhält man eine dickere Sperrschicht. Photonen finden so mehr Volumen zur Wechselwirkung mit dem Halbleiter. Bei Fotodioden wendet man diese Struktur an, um möglichst viele Ladungsträger pro Photon zu erhalten, also die Empfindlichkeit zu optimieren.

Eine andere Möglichkeit, mehr Empfindlichkeit zu erhalten, ist eine große empfindliche Fläche. Dies hat aber wiederum den Nachteil, dass die Kapazität steigt und ein ein-

zernes Signal eine kleinere Spannungsamplitude hat. Kommerziell erhältliche Halbleiter-Strahlungsdetektoren haben eine große Fläche und eine dickere I-Schicht. Im Vergleich dazu sind einfache PIN-Fotodioden wie die BPW34 weniger empfindlich, aber auch viel preiswerter.

Die PIN-Fotodioden BPW34 und BPX61 sind praktisch identisch, unterscheiden sich aber in ihrem Gehäuse. Während die (preiswerte) BPW34 in Kunststoff eingegossen ist, ist die BPX61 in einem TO5-Metallgehäuse mit Glasfenster montiert. Dieses Glasfenster kann man vorsichtig entfernen, so dass der Chip freiliegt. So wird die Diode auch empfindlich für Alpha-Strahlung.

Die Strahlen beziehungsweise Teilchen müssen zuerst die 15 µm dicke Alufolie (üblicher Wert von Haushaltsfolie) durchdringen. Das gelingt Gamma- und Betastrahlen mühelos. Alpha-Teilchen kommen ab einer Energie von 4 MeV ebenfalls noch durch. Beim Eintritt in den Kunststoff des Fotodiodegehäuses entsteht Bremsstrahlung in Form kleiner Lichtblitze, die möglicherweise ebenfalls noch detektiert werden können. Deshalb ist nicht auszuschließen, dass auch die BPW34 teilweise Alpha-empfindlich ist.

Prinzipiell ist jeder Halbleiter empfindlich für ionisierende Strahlung. Erstaunlich ist also weniger, dass eine Fotodiode solche Strahlung erfassen kann, sondern vielmehr, dass man sonst nicht viel davon merkt. Bekannt ist der Effekt bei dynamischen RAMs, die durch Strahlung spontan ihren Inhalt ändern können. Zunehmend hat man auch Probleme, Elektronik so zu bauen, dass sie der höheren Strahlenbelastung im Weltall standhält. Ältere Mikrocontroller mit

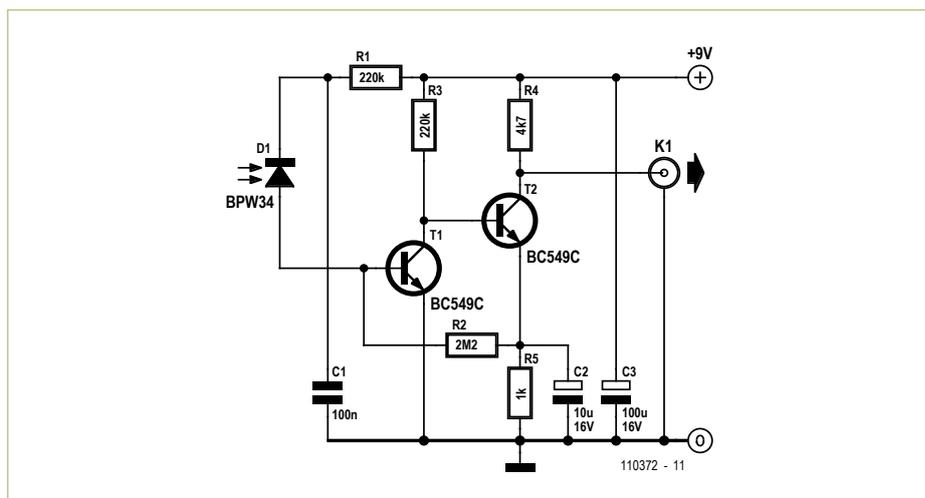


Bild 1. Der Messverstärker.

großen Transistorstrukturen waren weniger empfindlich. Aber seit die Strukturen immer kleiner werden, kann unter Umständen schon ein einzelnes energiereiches Teilchen eine Störung verursachen.

### Verstärker

In der Literatur findet man als Eingangsstufe von Ladungsverstärkern häufig einen rauscharmen Opamp mit FET-Eingängen. Hier wurde jedoch ein anderer Weg beschritten. **Bild 1** zeigt die Schaltung des Sensorverstärkers. Zwei Transistoren reichen aus, um das Signal der Fotodiode zu verstärken. Der direkt gekoppelte Verstärker stellt sich automatisch auf einen mittleren Arbeitspunkt ein, wobei die rauscharmen NPN-Transistoren BC549C zu einem guten Störabstand beitragen.

Der Transistor-Eingang des Verstärkers ist vergleichsweise niederohmig, was eine gute Rauschanpassung ermöglicht. Die erste Stufe arbeitet mit ihrer Basis-Kollektor-Kapazität zugleich als Integrator, macht also aus den sehr kurzen Impulsen der Fotodiode längere Impulse, die sich besser weiter verstärken lassen.

Um die Empfindlichkeit zu steigern, könnte man auch die Sperrspannung an der Diode

erhöhen. Damit verringert man die Diodekapazität und vergrößert gleichzeitig die Dicke der Sperrschicht. Erlaubt sind bis zu 32 V, möglicherweise liegt das Optimum aber tiefer. Bei 9 V arbeitet die Diode schon sehr gut. Man könnte auch zwei oder mehr Fotodioden parallel schalten und damit eine Empfindlichkeit erreichen, die der eines kleinen Zählrohres (zum Beispiel ZP1310) entspricht.

An den Ausgang der Schaltung kann man ein Oszilloskop zur Auswertung anschließen. Wer lieber ein Knacken aus einem Lautsprecher wie bei einem „richtigen“ Geigerzähler hören möchte, der findet im Textkasten „Strahlung hörbar machen“ die passende Lösung.

### Aufbau

Die Schaltung wurde auf einem Lochraster-Platinenstreifen aufgebaut (**Bild 2**), wobei sich die Fotodiode auf der Platinenunterseite (**Bild 3**) befindet. Um Licht vom Sensor fernzuhalten, wurde die ganze Schaltung in Alufolie eingepackt (**Bild 4**). Dafür eignet sich die bereits genannte Haushalts-Alufolie, die dünn genug ist, um auch Beta-Teilchen passieren zu lassen. Die Folie wirkt gleichzeitig als elektrische Abschirmung.

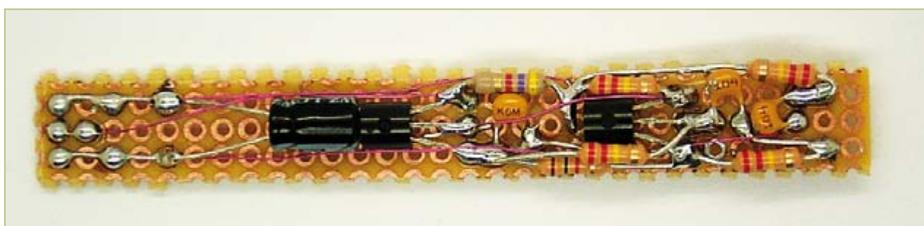


Bild 2. Probeaufbau des Sensorverstärkers.

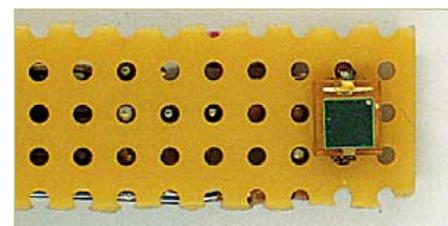


Bild 3. Der Sensor auf der Rückseite.

## Radium-Leuchtfarbe

Eine alte Uhr mit Leuchtziffern eignet sich gut zur Überprüfung von Strahlungsmessgeräten. Es kann auch ein alter Wecker oder ein Kompass vom Flohmarkt sein.

Radioaktive Leuchtfarbe wurde bis etwa 1965 eingesetzt und hat inzwischen fast keine Leuchtkraft mehr. Wenn Sie nicht wissen, ob Ihre Uhr radioaktiv ist, hilft ein einfacher Versuch ganz ohne Elektronik. Alles was Sie brauchen ist eine starke Lupe. Schalten Sie tief in der Nacht das Licht aus und gewöhnen Sie Ihre Augen an die Dunkelheit. Betrachten Sie dann die Leuchtzeiger durch die Lupe. Wenn deren

Farbe radioaktive Beimengungen enthält, sehen Sie ein schwaches Funkeln und Flimmern. Sie werden sozusagen Augenzeuge individueller Zerfallsereignisse. Die dabei entstehenden Alpha-Teilchen regen die Leuchtfarbe an. Wenn Sie kein Leuchten oder ein absolut gleichförmiges Leuchten sehen, ist sicher, dass es sich nicht um eine radioaktive Probe handelt. Vermutlich kann dieser Versuch mit radioaktiv dotierter Leuchtfarbe auch erst durchgeführt werden, weil diese inzwischen so alt ist. Am Anfang gab es vermutlich insgesamt zu viele Ereignisse, um diese einzeln zu erkennen.



Bild 4. Alles gut in Alufolie verpackt.

Damit es durch die Alufolie keine Kurzschlüsse gibt, muss man die Platine zuerst mit Isolierband umwickeln, wobei man das Fenster der Fotodiode aber frei lässt. Erst dann wird die Platine in Alufolie verpackt. Vergessen Sie nicht, die Folie an Masse zu legen.

### Versuche und Ergebnisse

Zur Auswertung verwendet man am besten ein digitales Oszilloskop. Es wird in den AC-Modus geschaltet. Günstige Einstellungen sind 50 mV/Skalenteil und 0,2 ms/Skalenteil. Einige Geräte erlauben einen Persist-Modus, wobei sich die Ergebnisse auf dem Bildschirm akkumulie-

ren. Es kann aber auch ein analoges Oszilloskop eingesetzt werden.

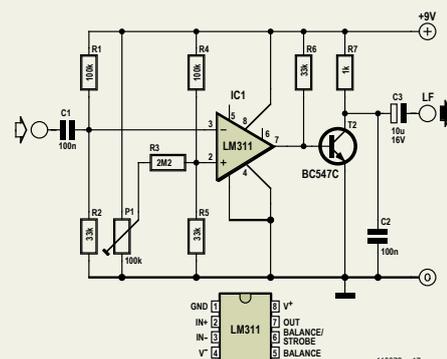
Im Ruhezustand entsteht ein ca. 30 mV<sub>SS</sub> dickes Band des verstärkten Grundrauschens (Bild 5). Wenn ein Gamma-Quant gemessen wird, entsteht ein positiver Impuls, der nur leicht nach unten überschwingt. Falls Sie gleich starke Signale auch im negativen Bereich finden, ist das ein Zeichen dafür, dass die Schaltung nicht optimal abgeschirmt ist und auf HF-Impulse reagiert. Die gesuchten Strahlen zeigen dagegen positive Impulse. In Bild 6 sieht man die über 30 s akkumulierten Signale einer alten Taschenuhr mit Radium-Leuchtzeigern.

Bild 7 zeigt die Messung an einer anderen radioaktiven Probe. Dabei handelte es sich um ein Stückchen Uran-Pechblende, also ein in der Natur vorkommendes Uranerz. Wieder wurde 30 Sekunden lang gemessen. Man erkennt sofort, dass diese Probe eine höhere Aktivität hat. Außerdem sieht man eine andere Energieverteilung. Es gibt mehrere Impulse von über 100 mV, die bei der Leuchtfarbe ganz fehlten. Anders als bei einem Geiger-Zählrohr kann man beim Halbleiter-Detektor die Energie der einzelnen Teilchen erkennen. So lassen sich Rückschlüsse darauf gewinnen, welche Art von Kern jeweils gerade zerfallen ist. Im Fall der Pechblende sollten alle Prozesse der Uran-Zerfallsreihe vorkommen. Bei der Leuchtfarbe dagegen ist der Ausgangskern vermutlich das Radium.

Dank der Möglichkeit, Messungen über einen längeren Zeitraum zu akkumulieren, können nun auch solche Proben untersucht werden, bei denen keine oder nur eine sehr geringe Radioaktivität zu erwarten ist. Das funktioniert besser als mit einem Zählrohr, weil es praktisch keine Nullrate gibt. Bei einem Geigerzähler hat man immer

## Strahlung hörbar machen

Ein Geigerzähler tickt immer so schön. Dagegen ist der Dioden-Sensor bisher noch stumm. Aber das muss nicht sein. Man muss nur einen Komparator nachschalten und die Impulse etwas verlängern, dann knackt es aus dem Lautsprecher. Die hier gezeigte, erprobte Schaltung verwendet einen Komparator LM311, der einen Ausgangsimpuls liefert, wenn Impulse am Eingang den mit dem Trimpoti eingestellten Schwellwert überschreiten. Der Transistor am Ausgang dient zur Impulsverlängerung, damit man die Impulse auch hören kann. Das Ausgangssignal steuert dann zum Beispiel einen Kopfhörer, einen NF-Verstärker mit Lautsprecher oder eine kleine Aktivbox an.



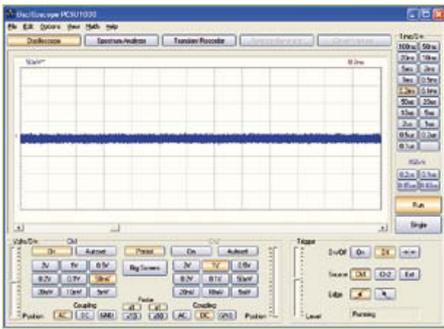


Bild 5. Der Ruhepegel.

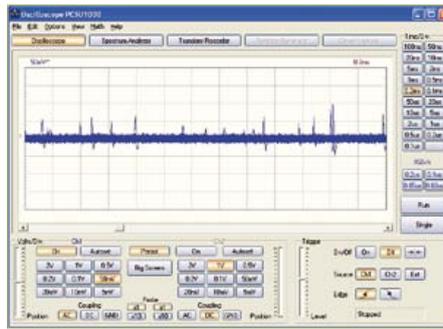


Bild 6. 30-Sekunden-Messung an einer alten Taschenuhr mit Leuchtziffern.

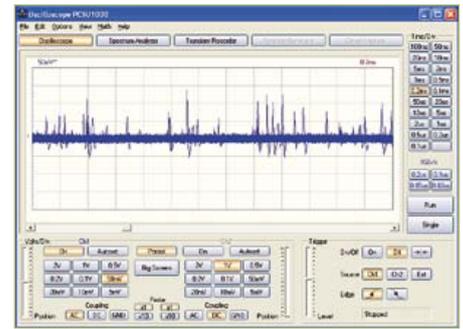


Bild 7. 30-Sekunden-Messung an einem uranhaltigen Mineral.

einige Impulse, die aus der Höhenstrahlung stammen. Diese harten Gammastrahlen treffen zwar ab und zu auch auf die Fotodiode, dies aber wegen der kleinen Fläche extrem selten. Nutzsignale heben sich deshalb deutlich vom Ruhezustand ab. **Bild 8** zeigt eine Messung an einem Stück Bleiglanz, einem Mineral, das (fast) nicht strahlen sollte. Nach einer halben Stunde fanden sich aber doch zwei deutliche Peaks. Ein ähnliches Ergebnis ergab sich mit Granit-

gestein, das bekanntlich ebenfalls schwach radioaktiv ist.

Eine radioaktive Quelle können auch technische Geräte oder Bauteile darstellen, die nicht nach heutigen Normen gebaut wurden. Bekannt ist zum Beispiel, dass bestimmte Glimmlampen und Glimmstabilisatoren für Brennspannungen unter 100 V mit radioaktiven Beimengungen versehen wurden. Eine alte russische Glimm-

röhre für 75 V/3 mA (**Bild 9**) hatte der Autor schon länger unter Verdacht. Auf das äußere Blech ist nämlich ein kleines Metallhütchen aufgeschweißt, unter dem man eine seltsame Pille erkennen kann. Nach innen hin ist ein kleines Loch vorhanden. Eine Langzeitmessung von ebenfalls einer halben Stunde lieferte tatsächlich Impulse mit besonders hoher Energie (**Bild 10**). Und das, obwohl die Strahlung erst noch durch das Glas der Röhre austreten musste.

Anzeige

# Volles Vertrauen in jede Anschaffung



VPO  
Visual Persistence Oscilloscope

## 350/250/150 MHz Digital Speicheroszilloskop

mit 5 GSa/s Abtastrate und VPO-Technologie ( Visual Persistence Oscilloscope ) zeigt die GDS-3000-Serie Kurvenformen originalgetreu und fängt gleichzeitig weniger häufig auftretende Signale wie Glitches oder Runts ein, ohne einen Punkt der Kurvenform-Information zu verpassen.

## GDS-3000-Serie

- 350-/250-/150-MHz-Bandbreite
- 5 GSa/s Real-Time-Abtastrate und 100 GSa/s Equivalent-Time-Abtastrate
- VPO-Technologie zur Anzeige von weniger häufig auftretenden Signalen
- Einzigartiges Split-Screen-Display mit unabhängiger Einstellung für jeden Eingangskanal
- Optionale Software für Leistungsmessung
- Optional Trigger- und Decoder-Software für serielle Bussysteme mit Unterstützung von I<sup>2</sup>C, SPI und UART

**CiEN** The electronics and digital industry Exhibition

May 26 - 28, 2011  
Booth No. Hall 4, H21  
PARIS, FRANCE

GOOD WILL INSTRUMENT CO., LTD.  
[www.gwinstek.com](http://www.gwinstek.com)

**GW INSTEK**  
Simply Reliable

## Radon-Zerfallsprodukte

Wenn Sie keine radioaktiven Substanzen zum Testen haben, könne Sie diese auch aus Ihrer unmittelbaren Umgebung gewinnen. Wir sind ständig von radioaktiven Stoffen umgeben. Aus dem Erdboden tritt nämlich immer etwas Radon aus. Dieses radioaktive Gas zerfällt mit einer gewissen Halbwertszeit und erzeugt dabei weitere Radionuklide. Und diese kann man selbst einsammeln.

Man nehme einen dünnen lackisolierten Kupferdraht (0,2 mm Durchmesser) und spanne ihn isoliert in der Wohnung aus. An den Draht lege man eine negative Hochspannung zwischen -5 kV und -10 kV. Nach zehn Minuten (und nach Abschalten der Hochspannung!) wische man den Draht mit einem Streifen Papier ab. Auf dem Papier findet man dann eine dunkle Spur: Das ist etwas Staub, der vom geladenen Draht angezogen wurde. In diesen Staubpartikeln befinden sich besonders viele der radioaktiven Radon-Zerfallsprodukte. Der Grund ist einleuchtend: Beim Zerfall erhalten die neuen

Kerne eine hohe Geschwindigkeit. Sie streifen dabei einen Teil ihrer Elektronenhülle ab und laden sich positiv auf. Deshalb werden sie vom negativ geladenen Draht angezogen.

Wenn man nun dieses „schmutzige“ Papier an einen Strahlungsmesser hält, ist eine hohe Aktivität zu messen. Sehr gefährlich ist das nicht, denn hätte man die Isotope nicht eingesammelt, hätte man sie vermutlich eingeatmet. Man kann auf diese Weise übrigens untersuchen, welche Räume wie stark mit Radon belastet sind. Im Keller ist es deutlich mehr, denn das Radon kommt ja von unten.

Ein geeigneter Hochspannungsgenerator ist zum Beispiel der als „Miniprojekt“ vorgestellte Ionisator aus Elektor 4/2009, S. 70 ([www.elektor.de/071072](http://www.elektor.de/071072)), bei dem man die Kaskade noch um zwei Stufen (zwei Kondensatoren und zwei Dioden) verlängert, um auf 5 kV zu kommen.

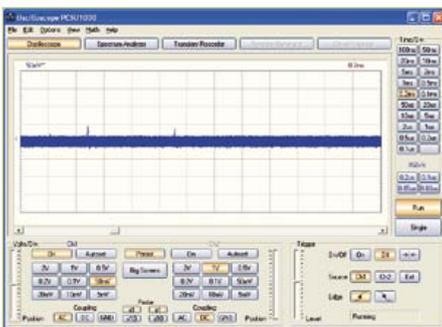


Bild 8. 30-Minuten-Messung an Bleiglanz.



Bild 9. Glimmröhre mit radioaktiver Zündhilfe.

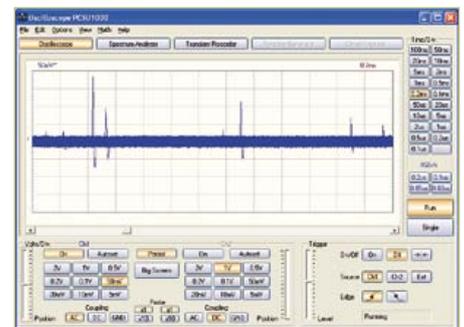


Bild 10. Messung an der Glimmröhre in Bild 9.

### Ausblick

Wir haben hier den Sensor und einen sehr einfacher Messverstärker vorgestellt. Baut man die Schaltung zusammen der im Textkasten beschriebenen Komparator-schaltung und einem Lautsprecher in ein Gehäuse ein, erhält man ein Gerät, das sich auch im Gelände einsetzen lässt, um zum Beispiel Mineralien in einem Steinbruch zu testen. Kombiniert man den Komparator

mit einem digitalen Zähler, so lässt sich auch die Gesamt-Aktivität erfassen. Mit einer zusätzlichen Sample&Hold-Schaltung könnte man auch Energieniveaus registrieren. Das Ergebnis ließe sich dann in Form eines Energie-Spektrogramms darstellen.

Interessant wären auch Langzeitversuche mit anderen Proben. So ist beispielsweise Kalium-Chlorid ein schwacher Betastrahler.

Es wäre sicher interessant zu untersuchen, ob die Fotodiode das nachweisen kann.

(110372)

### Literaturhinweise und Links:

- Videos zum BPW34-Detektor:  
[www.youtube.com/user/bkelektronik](http://www.youtube.com/user/bkelektronik)
- Maxim APPLICATION NOTE 2236 Gamma-Photon Radiation Detector  
<http://pdfserv.maxim-ic.com/en/an/AN2236.pdf>
- Erhan Emirhan und Cenap S. Özben: PIN Fotodiode Based X and  $\gamma$  Ray Detectors  
<http://thm.ankara.edu.tr/tac/YAZOKULU/yazokulu6/dersler/06-09-2010/erhan-emirhan-cenap-ozben-pin-fotodiode.pdf>
- C. W. Thiel: An Introduction to Semiconductor Radiation Detectors  
[www.physics.montana.edu/students/thiel/docs/Detector.pdf](http://www.physics.montana.edu/students/thiel/docs/Detector.pdf)
- Elektor 1/1980 S. 48-53: Geiger-Müller-Zähler  
[www.elektor.de/Geiger-Müller-Zähler](http://www.elektor.de/Geiger-Müller-Zähler)
- Funkschau 1986, Heft 21, S. 63-69: Radioaktivitätsmesser im Miniformat
- Funkschau 1986, Heft 21, S. 99-103: Das Messen radioaktiver Strahlung
- Thomas Rapp: Experimente mit selbst gebauten Geigerzählern, Funken- und Nebelkammern, Franzis-Verlag 2008

### Echte Sinus-Power

Sinus-Wechselrichter wandeln Gleichspannung von 12 V in Wechselspannung von 230 V um. Durch die Ausgabe der 230-Wechselspannung in Sinusform sind auch empfindliche elektronische Geräte wie Fernsehgeräte, PC-Netzteile oder Messgeräte über den Spannungswandler betreibbar.

- Eingangsspannung: 10-15 V DC
- Ausgangsspannung: 220 V - 240 V AC
- Überhitzungsschutz
- Überlastschutz



# 99.95

SN-150	
SN-300	119.95
SN-600	169.95
SN-1000	259.95

Spannung	Dauerleistung	Anschluss
12 V DC	150W	12V-Kfz-Adapter
12 V DC	300W	Klemmen / 12V-Kfz
12 V DC	600W	Klemmen
12 V DC	1000W	Ringklemmen



## Markenqualität - Top-Service - günstige Preise!

Katalog kostenlos!  
 Tagesaktuelle Preise:  
[www.reichelt.de](http://www.reichelt.de)



### KUNG LONG Blei-Vlies-Akkus

• Spannung: 12 Volt



WP	Kapazität	H x L B	Faston	Preis
WP 1,2-12	1,2 Ah	59x97x43	5,8 mm	7.20
WP 1,9-12	2,2 Ah	66x178x34	4,8 mm	8.95
WP 4,5-12	5,4 Ah	107x90x70	4,8 mm	10.50
WP 7,2-12	7,2 Ah	102x151x65	6,3 mm	13.15
WP 10-12	10 Ah	112x151x65	6,3 mm	27.75
WP 12-12	12 Ah	100x151x99	6,3 mm	21.95
WP 18-12	18 Ah	167x180x75	M5	31.75
WP 26-12	26 Ah	125x166x175	M5	47.40

### Stecker-Schaltnetzteil, max. 600 mA

- Ausgangsspannung: 3,0 / 4,5 / 5,0 / 6,0 / 7,5 / 9,0 / 12 VDC
- 9-teiliges Adapterset
- TÜV-/GS-Geprüft



Leistung	Strom	Maße	Preis
MW 3N06GS	7,2 W	69 x 39 x 31 mm	5.95
MW 3K10GS	12 W	75 x 43 x 32 mm	7.50



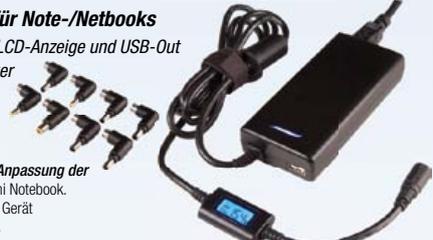
### Universalnetzteil automatisch für Note-/Netbooks

- Notebook- und Netbooknetzteil mit LCD-Anzeige und USB-Out
- inkl. 10 Adapterstecker + Dell Stecker



Einfachste Handhabung durch automatische Anpassung der Ausgangsspannung an ihr Notebook oder Mini Notebook. Eine fehlerhafte Ausgangsspannung die ihr Gerät beschädigen könnte, ist somit ausgeschlossen.

Leistung	Strom	Output	Preis
PWS 90 LCD	90 W	9,5 - 22 auto / 5V USB-Out VDC	37.50
PWS 120 LCD	120 W	15 - 24 auto / 5V USB-Out VDC	43.50



### Wärmebildkamera FLIR i3

- Präzise Thermografie bei der Elektro- und Gebäude-Inspektion
- Sichtfeld / min. Fokussentfernung: 12.5° x 12.5°/0.6 m
- Infrarotauflösung: 60x60 Pixel
- Thermische Empfindlichkeit (N.E.T.D.): 0.15°C
- Display: 2,8" Farb-LCD
- Objekttemperaturbereich: -20 °C bis +250°C
- Bildspeicherung auf MiniSD-Karte, Dateiformat: JPEG, einschließlich Messdaten



FLIR i3 **1184.05**

### Laser-Entfernungsmessgerät

- professionelle Laser-Entfernungsmessgerät mit LCD-Anzeige und Hintergrundbeleuchtung
- Messbereich: 0,05 - 50 m
- Messeinheiten: m (meter), in (inch), ft (feet)
- Messgenauigkeit: ± 1,5 mm
- Schutzklasse: IP 54
- Schnellasten für Addition
- Speicherfunktion



PEAKTECH 2800 **89.95**

### Sprays für den Arbeitsplatz

- Druckluft 67, 400 ml - nicht brennbar



KONTAKT 334 **15.25**

KONTAKT 361 **9.75**

KONTAKT 316 **8.95**



Grenzen überschreiten und neue Maßstäbe setzen – das gilt nicht nur für unser Produktportfolio, sondern auch einmal mehr für unsere Logistik.

## Mindestbestellwert gesenkt!

Für Aufträge aus Deutschland, den EU-Ländern und der Schweiz beträgt der Mindestbestellwert jetzt nur noch 10,00 €.

Weitere Informationen unter:  
<http://versandkosten.reichelt.de>



### 3-in-1-Lötstation

Professionelle Löt-/ Entlöt- und Heißluft-SMD-Rework-Station

- Promotion-Set mit Lötdampfabsauger (Industriequalität) solides Metallgehäuse
- Nullspannungs-Regelschaltung
- galvanisch vom Netz getrennt (ESD-geschützt)
- Löten / Entlöten 100 Watt
- Heißluft-SMD-Rework 600 Watt, ECO-Standby



STATION LF-853D **499.95**

# LED-Fahrradlampe

## Hightech für Feldwege

Von Thomas Finke (D)

Käufliche und in der Straßenverkehrsordnung zugelassene Fahrradlampen verwenden ein Halogenbirnchen mit maximal 2,4 W. Auf dunklen Wegen fernab der Zivilisation kann man damit nicht wirklich etwas sehen. Die hier vorgestellte Lampe ist in Deutschland zwar nicht für den Straßenverkehr zulässig, dafür bringt sie es aber mit vier Hochleistungs-LEDs auf etwa 600 Lumen und lässt sich außerdem auf Knopfdruck dimmen. Die Stromversorgung erfolgt durch vier in die Lampe integrierte Lilon-Akkuzellen.



Die Akkukapazität von 2,2 Ah pro Zelle reicht bei voller Helligkeit im Gelände für etwa drei Stunden und im gedimmten Modus für ein Vielfaches davon. Der Mikrocontroller in der Schaltung übernimmt auch das Lademanagement, so dass zum Aufladen ein einfaches 5-V-Steckernetzteil oder ein Autoadapter ausreicht, der 5 V liefert.

### Auslegung

Bei jedem Fahrzeugscheinwerfer ist dafür zu sorgen, dass er den Gegenverkehr nicht blendet. Bei konventionellen Lampen sorgt eine Optik dafür, dass der Lichtkegel nach oben scharf begrenzt wird. Eine solche Optik ist bei einer Selbstbau-LED-Lampe für Wald- und Feldwege nicht nur schwierig zu realisieren, sondern auch von Nachteil, was die Ausleuchtung des Geländes betrifft. So bald es die Wegebeschaffenheit zulässt, kann man die Leistung der Lampe mit einem Druck auf einen Taster verringern.

Dieses „Abblendlicht“ reduziert nicht nur eine mögliche Blendwirkung, sondern spart auch viel Strom, so dass im Mischbetrieb auch lange Nachtfahrten möglich sind. Wenn man im normalen Straßenverkehr legal unterwegs sein möchte, muss man dafür eine für den Straßenverkehr zugelassene Beleuchtung verwenden. Die LED-Lampe kommt dann zum Einsatz, sobald man den öffentlichen Straßenverkehr verlässt.

### Bedienung

Der Taster ist das einzige Bedienelement der Lampe. Ein kurzer Druck schaltet die Lampe ein, jeder weitere lässt die Helligkeit zwischen „Abblendlicht“ und „Fernlicht“ wechseln. Ausgeschaltet wird durch einen langen Tastendruck. Mit dem Taster lässt sich auch die Helligkeit des Abblendlichts in vier Stufen einstellen. Dazu schaltet man die Lampe mit einem langen Tastendruck ein und wählt anschließend durch weitere

Tasterbetätigungen die gewünschte Helligkeitsstufe aus. Nach kurzer Zeit wechselt die Lampe in den normalen Betrieb.

Die Elektronik der Lampe informiert mit einer in den Taster integrierten LED über den Ladezustand des Akkus, indem sie periodisch und abhängig von der Akkuspannung ein- bis viermal blinkt (viermal bei vollem Akku). Unterhalb einer bestimmten Schwelle schaltet die Lampe automatisch auf die niedrigste Helligkeitsstufe, um die Restbetriebsdauer zu verlängern.

### Schaltung

Weiß LEDs haben eine Fluss-Spannung von etwa 3,5 V, während Lilon-Akkus eine Spannung zwischen 2,7 und 4,2 V liefern. Zum Betrieb der LED müsste die Akkuspannung daher je nach Ladezustand reduziert oder erhöht werden. Dazu bräuchte man einen Gleichspannungswandler, dessen

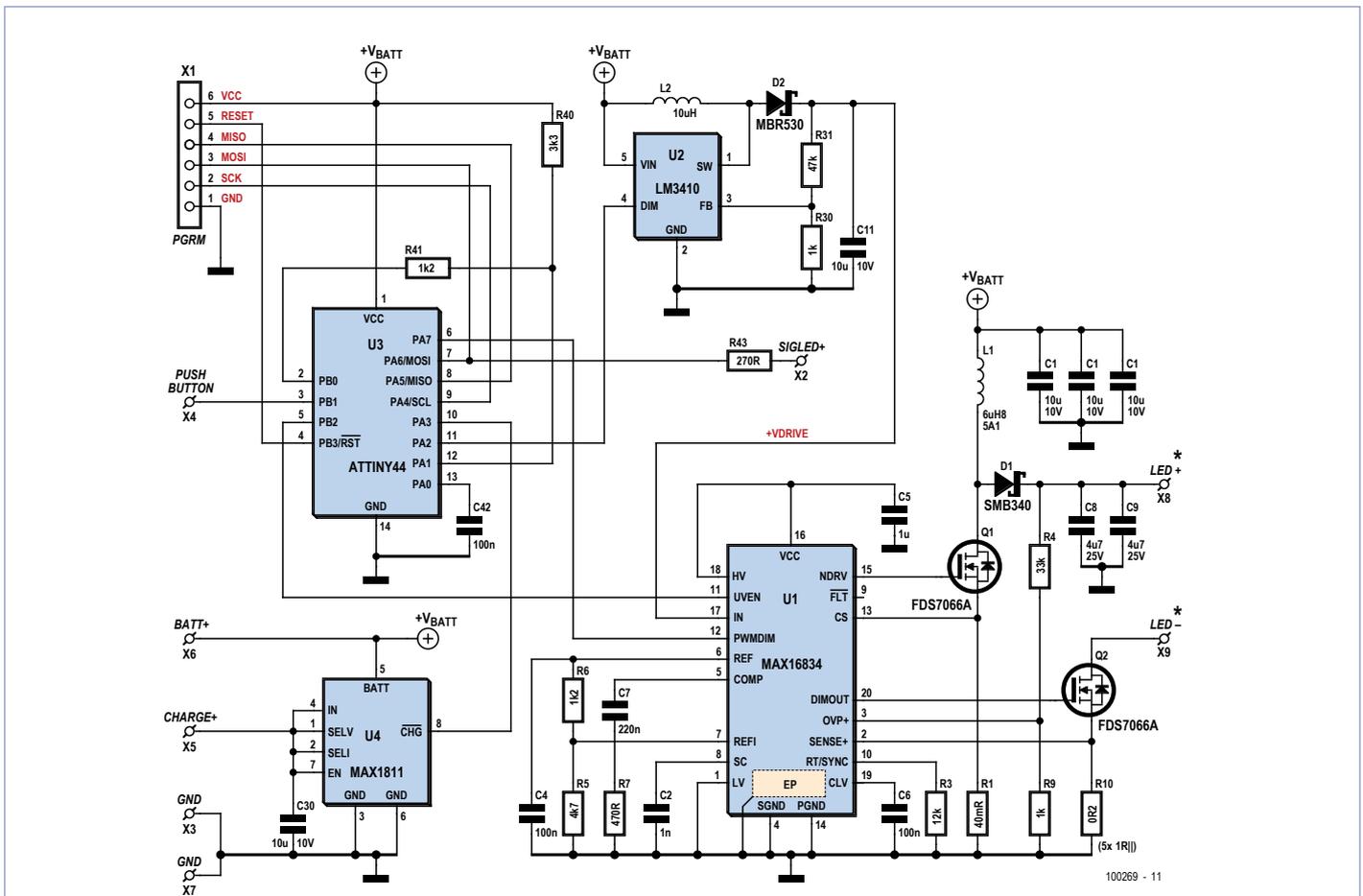


Bild 1. Die Schaltung besteht aus einem Mikrocontroller, zwei Schaltwandlern und einem Akku-Laderegler.

Ausgangsspannung sowohl größer als auch kleiner als die Eingangsspannung sein kann, wie es bei einem so genannten SEPIC-Wandler (single ended primary inductance converter) der Fall ist. Der Nachteil eines solchen Wandlers ist neben dem Schaltungsaufwand ein relativ schlechter Wirkungsgrad. Daher wurde für dieses Projekt ein Modul in Form einer Rundplatine mit vier in Reihe geschalteten LEDs (CREE XR-E) eingesetzt, für das auch eine passende Optik verfügbar ist. Zum Betrieb der LEDs kann dann ein Aufwärtswandler (Step-up- beziehungsweise Boost-Converter) eingesetzt werden, der die erforderliche Spannung von etwa 14 V erzeugt. In der Schaltung (Bild 1) wurde dafür ein MAX16834 (U1) verwendet. Die Beschaltung des ICs entspricht im Wesentlichen dem Applikationsbeispiel aus dem Datenblatt, die Bauteilwerte wurden lediglich an den Verwendungszweck angepasst. Zu beachten ist, dass die Drossel L1 bei einem LED-Strom von 1 A und minimaler Akkuspannung im Schnitt immerhin 5,6 A verkraften muss. R10

bestimmt den LED-Strom. Auf der Platine besteht R10 aus ein bis fünf parallel geschalteten 1-Ω-Widerständen. Der LED-Strom hängt davon ab, wie viele 1-Ω-Widerstände man bestückt. Bestückt man alle fünf Widerstände, ergibt sich für R10 ein Wert von 0,2 Ω. Der LED-Strom kann so bis auf 1 A gebracht werden. Der Autor hat in seinem Prototyp nur drei 1-Ω-Widerstände bestückt und den LED-Strom so auf 600 mA begrenzt. Die Helligkeit ist damit immer noch mehr als ausreichend und die Wärmeentwicklung der LEDs hält sich in Grenzen. Ein an den Pin PWMDIM angelegtes PWM-Signal kann die LEDs dimmen. Hiervon wird bei der Reduzierung der Lampenleistung Gebrauch gemacht. An Pin 11 (UVEN) wird normalerweise ein Spannungsteiler angeschlossen, der den Baustein bei Unterspannung abschaltet und in einen Stromsparmodus versetzt. In der Schaltung der LED-Lampe wird das durch den ATmel-Mikrocontroller (U3) über einen Portpin (PB2) erledigt. Für den Betrieb und das sichere

Durchschalten der MOSFETs benötigt U1 eine Betriebsspannung von mindestens 5 V, wofür die Akkuspannung nicht ausreicht. Daher wurde ein Hilfswandler mit einem LM3410 (U2) eingesetzt, der eine stabilisierte Spannung (+VDRIVE) bereitstellt. Der LM3410 ist eigentlich ein Stromregler für LEDs bis 500 mA. Am Beginn der Entwicklung hat der Autor sogar überlegt, die gesamte Lampenschaltung mit einer Anzahl dieser Bausteine zu realisieren. Übrig geblieben ist nur einer, der aber in der angegebenen Beschaltung einen wunderbar kleinen Aufwärtswandler mit konstanter Ausgangsspannung abgibt. Der ATtiny44-Mikrocontroller von Atmel (U3) übernimmt die Steuerung der anderen Schaltungskomponenten. Er erzeugt das zum Dimmen der Power-LEDs benötigte PWM-Signal und versetzt U1 und U2 bei Bedarf in den Ruhezustand. Der eingebaute Analog-Digitalwandler überwacht über einen Spannungsteiler mit R40 und R41 die Akkuspannung. Damit der Querstrom dieses Spannungsteilers den Akku bei ausgeschalteter Lampe nicht unnötig

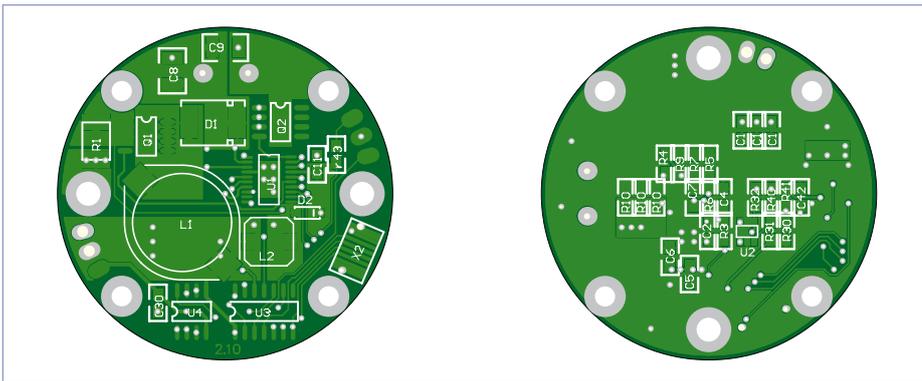


Bild 2. Für den Aufbau wurde eine kreisförmige Platine entwickelt.



Bild 3. Die weitgehend SMD-bestückte kreisförmige Platine.



Bild 4. Blick ins Innenleben der Lampe.

entlädt, ist R41 nicht direkt nach Masse geschaltet, sondern an PB0 angeschlossen. Dieser Port-Pin kann hochohmig geschaltet und so der Stromfluss durch den Spannungsteiler unterbrochen werden. Tatsächlich ist die Lampe nie wirklich ausgeschaltet. Stattdessen wechseln Atmel, Hilfswandler und LED-Treiber in einen Ruhemodus. Zusammen verbrauchen sie dann nur noch gut 100  $\mu$ A, die man aber in Relation zur Akkukapazität von 8,8 Ah getrost vernachlässigen kann. Zum Aufwecken löst der an PB1 angeschlossene Taster einen Interrupt des Controllers aus.

### Stromversorgung

Um die Ladung des Akkus kümmert sich der MAX1811 (U4). Dieser Baustein wurde von Maxim eigentlich für das Laden von Lilon-Akkus am USB-Port entwickelt. Das IC liefert daher nur etwa 500 mA Ladestrom, eine Vollladung kann daher an die 20 Stunden dauern. Es wäre aber ohnehin problematisch, einen höheren Strom über die verwendete Ladebuchse zuzuführen, und bei einem höheren Ladestrom müsste man aus Sicherheitsgründen auch eine Überwachung der Akkutemperatur vorsehen. Da die Lampe aber nur wenige Stunden am Tag eingesetzt und dann fast immer nur teilweise entladen wird, stellt dies im praktischen Einsatz keine Einschränkung dar.

Der Akku selbst besteht aus vier parallel geschalteten Rundzellen, die dem Akkupack eines defekten Notebooks entnommen wurden. Bei einer Reihenschaltung der Zellen hätte man zwar den Hilfswandler einsparen können, hätte dafür aber einen höheren Aufwand beim Laden und Ausbalancieren der Zellen in Kauf nehmen müssen.

Bei dem auf 600 mA begrenzten LED-Strom des Prototyps genügt eine Akkuladung für knapp 3,5 Stunden Dauerbetrieb bei maximaler Helligkeit.

### Aufbau

Für den Aufbau der Schaltung wurde eine runde, doppelseitige Platine entwickelt (Bild 2). Die bestückte Platine ist in Bild 3 zu sehen.

Der LED-Wirkungsgrad wurde in den letzten Jahren stetig verbessert. Trotzdem entsteht im Betrieb immer noch eine Menge Wärme, die es von den empfindlichen LEDs abzuleiten gilt. Das LED-Modul (4 x CREE XR-E auf einer Rundplatine mit 34 mm Durchmesser) wurde daher auf eine 5 mm dicke Alu-Scheibe montiert, die ihrerseits guten Kontakt zum Rest des Alugehäuses hat, das die Wärme an die Umgebung ableitet. Die Rückseite der Scheibe dient zugleich als Kühlfläche für die MOSFETs der Ansteuerschaltung.

Das Gehäuse selbst wurde aus Rundmaterial und einem Stück Rohr mit 50 mm Durchmesser gedreht und selbst schwarz eloxiert, was sich als überraschend unproblematisch erwies. Auf der Webseite [1] zu diesem Beitrag finden sich zwei Skizzen zum Aufbau beziehungsweise Gehäusebau. In Bild 4 erkennt man den Aufbau der Lampe: Gehäusevorderteil mit Linse, LED-Trägerplatine, 5-mm-Alu-Scheibe zur Kühlung, Elektronikplatine, Abstandshalter, Kunststoffplatte und Batteriezellen.

Das Gehäuse muss absolut wasserdicht sein. Die Linse wurde daher mit Silikon in das Gehäusevorderteil eingeklebt, Vorder- und Hinterteil sind mit O-Ringen im Rohr abgedichtet. Der Taster ist eine wasserdichte Ausführung mit integrierter LED und wird mit dem hinteren Gehäuseteil verklebt. Als Ladebuchse dient eine versenkt und damit geschützt angebrachte SMB-Buchse. Diese Miniatur-Koaxbuchse ist vergoldet und daher unempfindlich gegen Feuchtigkeit. Diese Buchse wird ebenfalls wasserdicht im Gehäuse verklebt.

Die Lampe hat sich dank dieser konstruktiven Maßnahmen schon seit fast zwei Jahren im täglichen Einsatz bewährt und ist auf schlechten Wegen einfach unentbehrlich.

(100269)

[1] Webseite zu diesem Leserprojekt mit Downloads: [www.elektor.de/100269](http://www.elektor.de/100269)

# Event-Kalender

Workshops • Seminare • Masterclasses



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



## FPGA – Der Logikbaukasten für Jedermann

München 19.05.2011  
Dortmund 20.06.2011  
Hannover 29.09.2011  
Hanau 05.10.2011

[www.elektor.de/fpga-seminar](http://www.elektor.de/fpga-seminar)

## Grafische AVR-Programmierung mit Flowcode

München 20.05.2011  
Hamburg 22.09.2011  
Berlin 27.09.2011  
Dortmund 10.11.2011

[www.elektor.de/avr-prog](http://www.elektor.de/avr-prog)

## AVR-Mikrocontroller (für Einsteiger)

München 21.05.2011  
Hanau 24.09.2011  
Hamburg 05.11.2011  
Hannover 10.12.2011

[www.elektor.de/avr-workshop](http://www.elektor.de/avr-workshop)

## Echtzeitbetriebssysteme in Theorie und Praxis

Seligenstadt 23.05. bis 25.05.2011  
Hamburg 02.11. bis 04.11.2011  
München 06.12. bis 08.12.2011

[www.elektor.de/ezb-systeme](http://www.elektor.de/ezb-systeme)

## Eagle PCB und Design

Zürich (CH) 29.06.2011  
Hannover 29.09.2011  
Hanau 06.10.2011

[www.elektor.de/eagle-seminar](http://www.elektor.de/eagle-seminar)

## C-Programmierung für Mikrocontroller

Hanau 13.09. bis 15.09.2011

[www.elektor.de/c-prog](http://www.elektor.de/c-prog)

## Workshop zum $\mu$ C-Fernlehrgang 2

Hanau 17.09.2011

[www.elektor.de/tfh-workshop](http://www.elektor.de/tfh-workshop)

## LabVIEW meets $\mu$ C

München 07.10. + 08.10.2011

[www.elektor.de/labview-seminar](http://www.elektor.de/labview-seminar)

NEU!



Änderungen vorbehalten.

Weitere Infos unter

[www.elektor.de/events](http://www.elektor.de/events)



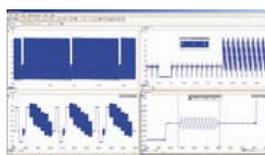
## PicoScope® 3000-Serie

DIE LEISTUNGSFÄHIGSTEN USB-GESPEISTEN OSZILLOSKOPE AUF DEM MARKT

DIE NEUE PICO SCOPE 3000-SERIE

Leistung und Mobilität. Warum Kompromisse eingehen?

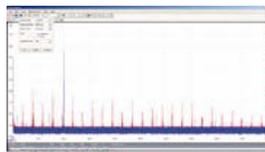
200 MHz Bandbreite  
Riesiger 128-MS-Puffer  
500 MS/s Real-time-Sampling  
10 GS/s Repetitive-Sampling  
Weiterentwickelte digitale Trigger-Modi  
200-MHz-Spectrum-Analyser  
Integrierter Funktionsgenerator/AWG  
Anschluss und Stromversorgung über USB



128 MS Pufferspeicher



Seriell decodieren



200-MHz-Spectrum-Analyser



Arbiträr-Funktionsgenerator

High-End-Funktionen als Standard. Warum Kompromisse eingehen?

Seriell-Decodierung Segmentierter Speicher  
Masken-Grenzwertprüfung

[www.picotech.com/scope3102](http://www.picotech.com/scope3102)



# Positionsbestimmung mit ATM18

## Wo fährt Ihr Auto gerade herum?

Von Grégory Ester (Frankreich)

Das Modem-Modul GM862-GPS von Telit enthält einen SiRFstarIII-GPS-Receiver mit 20 Kanälen, mit dem man die aktuelle Position sehr genau bestimmen kann. Kombiniert mit einem Elektor-ATM18-Modul hat man ein GPS-Tracking-Gerät, das man zum Beispiel ins Auto einbauen kann. Der Eigentümer wird dann jederzeit via E-Mail oder SMS über den aktuellen Aufenthaltsort des Fahrzeugs informiert.

Im Artikel „Fernbedienung per Handy“ von Elektor Mai 2011 [1] wurde ein Projekt mit dem gleichen Modem-Modul beschrieben, doch dabei wurde die GPS-Funktion dieses vielseitigen Moduls überhaupt nicht genutzt. Dieser Beitrag will daran etwas ändern. Auch wenn es um die gleiche Hardware geht, muss man nicht zwingend den Mai-Artikel lesen, denn alles Wissenswerte über die GPS-Funktion findet sich im folgenden Text. Wie zuvor wird ein ATM18-Modul [2] (zusammen mit einem I2C-LCD [3]) verwendet, das mit einer OEM-Variante des GM862-GPS-Moduls [4] gekoppelt ist. Letzteres steckt zusammen mit den Antennen für GPS und Mobilfunk auf einem Prototypen-Board [5]. **Bild 1** zeigt die Verbindungen zwischen den Modulen. Mit dieser Hardware-Kombination lassen sich drei verwandte, aber doch unterschiedliche Anwendungen realisieren, die in **Tabelle 1** aufgelistet sind. Für jedes dieser Projekte gibt es eine eigene Firmware, die kostenlos von der Elektor-Webseite zu diesem Artikel [9] heruntergeladen werden kann.

### Update für das GM862-GPS-Modul

Falls man das GM862-GPS-Modul schon vor einiger Zeit gekauft haben sollte, dann ist die Firmware vermutlich älter als die hier verwendete Version 07.03.402 und es ist ein Update erforderlich. Die aktuelle Firm-

ware-Version des Moduls findet man heraus, indem man es an einen PC anschließt. Hierzu verbindet man die Leitungen RXI und TXO mit TxD (orange) und RxD (gelb) eines USB/Seriell-Konverters (wie zum Beispiel der FTDI-Konverter [6]) anstelle der Pins PC2 und PC3 des ATM18-Moduls. In einem Terminal-Programm (beispielsweise Hercules [7]) gibt man dann den Befehl „AT+CGMR\$0D“ ein. Wenn das Modem dann mit „Version 07.03.402“ (oder neuer) antwortet, kann man die nächsten drei Schritte überspringen.

Das Update wird folgendermaßen durchgeführt:

1. Der Befehl „AT+IPR=115200\$0D“ konfiguriert das Modem mit einer Baudrate von 115.200 Bd. Anschließend wird das Terminal-Programm mit der gleichen Baudrate neu gestartet.

2. Als nächsten Schritt registriert man sich auf der Webseite von Telit [8], damit man Updates herunterladen kann. Für das Update benötigt man das Tool „Xfp“ ab der Version 2.13 und natürlich die Datei mit der neuen Firmware. Beides kann man auch per E-Mail von einem Telit-Support-Mitarbeiter unter „Khaled.Chtourou@telit.com“ anfordern.

3. Nun wird die Stromversorgung des Prototypen-Boards eingeschaltet. Das Modul muss ausgeschaltet sein, falls nötig, hält man den On/Off-Taster gedrückt, bis die „STAT“-LED erlischt. Nun wird Xfp gestartet und dessen Datenrate ebenfalls auf 115,200 Bd eingestellt. Mit einem Klick auf „Browse“ wählt man die neue Firmware-Datei aus. Ein Klick auf „Program“ führt zur Meldung „Linking...“ unten am Fenster. Gleichzeitig beginnt der Fortschrittsbalken zu blinken (siehe **Bild 2**). Nun wird das Modem eingeschaltet. Dies funktioniert ähnlich wie bei einem Handy: Ein kurzer Tastendruck genügt. Nun sollte die grüne „STAT“-LED leuchten und das Update starten. Letzteres sieht man am Fortschrittsbalken (**Bild 3**). Nach beendetem Update erlischt die „STAT“-LED und die Meldung von **Bild 4** erscheint. Nun kommt ein Klick auf „OK“ und ein Druck auf den Taster „On/Off“ sowie nochmals der Befehl „AT+CGMR\$0D“ im Terminal. Jetzt wird zum Zeichen des Erfolgs die Version der neuen Firmware (aktuell „07.03.402“) ausgegeben. Zum Schluss setzt man mit dem Terminal-Befehl „AT+IPR=9600\$0D“ die Baudrate wieder auf 9.600 Bd zurück.

### GPS-Funktionen des GM862-GPS-Moduls

Das Modul kann unterschiedliche Arten geographischer Positionsdaten sammeln.

#### Elektor Produkte & Service

- ATM18-Mikrocontroller-Board: #071035-91
- ATM18-Test-Board: #071035-92

- 2-Draht-LCD-Modul: #071035-93

- Projekt-Software: #110267-11.zip

Erhältlich via [www.elektor.de/110267](http://www.elektor.de/110267)

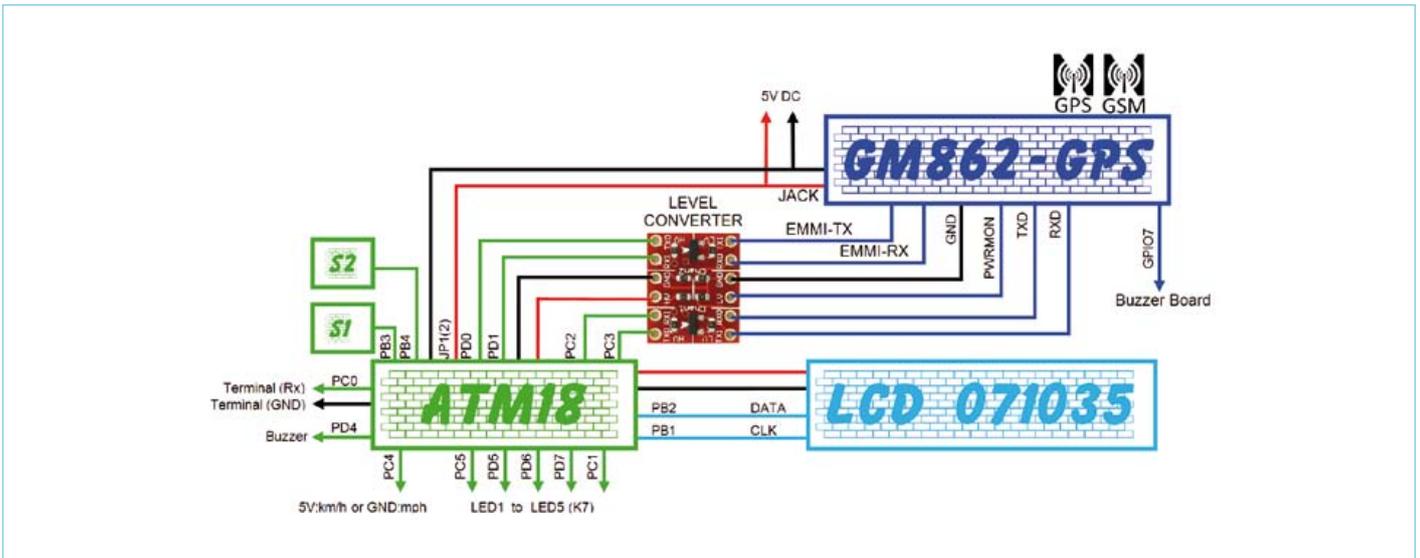


Bild 1. Blockschaltung und Verdrahtung der Module.

Tabelle 1. Firmware-Dateien für drei Anwendungen	
Datei	Funktion
110267-I_GM862-GPS_ATM18_P3.bas	Zeigt die aus NMEA-Frames extrahierten Daten auf dem LCD an.
110267-I_GM862-GPS_ATM18_P4.bas	Sendet Geo-Daten via SMS oder E-Mail.
110267-I_GM862-GPS_ATM18_P5.bas	Sammelt LAC- und CI-Codes neben Signalqualität und Geo-Daten.

Man kann diese Daten zwar auch per üblichem AT-Kommando abfragen, doch gibt es noch eine andere Möglichkeit: Das Prototypen-Board hat einen Ausgang mit der Bezeichnung „EMMI\_TX“ über den die gleichen Geo-Daten ohne AT-Befehl übermittelt werden. Über diesen Port werden die Daten in Frames nach dem NMEA0183-Format ausgegeben, welches der Standard für die maritime serielle Kommunikation darstellt. Damit dieser Port Daten liefert, startet man das Modul zunächst per On/Off-Taster. Jetzt blinkt die „STAT“-LED im Sekundenrhythmus und zeigt so an, dass das Modem noch nicht in ein Mobilfunk-Netz eingeloggt ist. Trotzdem erscheinen schon die interessierenden NMEA-

Frames an diesem Port. Standardmäßig werden die Daten mit 4.800 Bd ausgegeben. Die Frames erscheinen in der Reihenfolge GGA, GSA, GSV und RMC. Die Bedeutung dieser Frames wird nachfolgend erläutert.

Die erste der drei Anwendungen benötigt die Software der Datei „110267-I\_GM862-GPS\_ATM18\_P3.bas“, welche die Frames GGA und RMC decodiert. Unter anderen Daten enthalten GGA-Frames die Zeit in UTC, die geografische Breite und Länge, die Höhe über Meeresspiegel, die Anzahl der empfangenen Satelliten und eine Check-Summe. Wenn man die Geschwindigkeit und das Datum benötigt, muss man die RMC-Frames

decodieren. Wenn weniger als vier Satelliten empfangen wurden, gelten die Daten als ungültig. Bei genug Satelliten werden die gewünschten Daten vom Programm aus den Frames extrahiert und diese entsprechend formatiert zum LCD geschickt. Einen Eindruck von der Datenausgabe vermittelt **Bild 5**. Die Anzeige wird sekundlich aufgefrischt. Bei Betätigung von Taster 2 ändert sich die Anzeige wie in **Bild 6**. Hier wird die Geschwindigkeit in der dritten Zeile angezeigt. Mit Pin PC4 schaltet man die Anzeige zwischen „mph“ (Massepegel) und „km/h“ (+5 V) um (siehe **Bild 7**). Die Zeitzone wird in der Software auf Mitteleuropäische Zeit (MEZ = GMT + 1 h) eingestellt.



Bild 2. Xfp: „Ready“-Dialog.

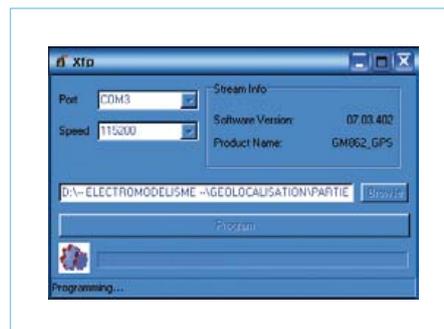


Bild 3. Xfp: „Go“-Dialog.



Bild 4. Xfp: „Done“-Dialog.



Bild 5. Die aktuelle Zeit (GMT + 1) ist 20:43.



Bild 6. Aktuelle Positions-Infos.

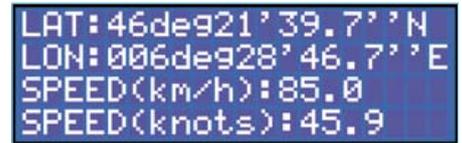


Bild 7. Geschwindigkeit in km/h.

### ATM18 als Petze

Die zweite Anwendung des GM862-GPS-Moduls kann E-Mails via Port 25 eines SMTP-Servers versenden. Ein Handy mit GPRS-Fähigkeiten verfügt prinzipiell über E-Mail-Kompetenzen. Allerdings enthalten einige Handy-Verträge - gerade die preiswerteren - keine GPRS-Dienste. Wenn das der Fall sein sollte, muss man den Mobilfunk-Zugang zum Internet zuvor vom Provider freischalten lassen. Bei aktivierten Datendiensten (siehe **Bild 8**) kann das Modul E-Mails versenden. Zur Nutzung braucht man nur noch die passende Software in Form der Datei „110267-I\_GM862-GPS\_ATM18\_P4.bas“.

**Tabelle 2** zeigt beispielhaft, welche Angaben das GM862-GPS-Modul für den E-Mail-Versand benötigt. Außerdem braucht es natürlich noch die E-Mail-Adressen von Sender und Empfänger. Nach der Eingabe des PIN-Codes kann man die folgenden Befehle verwenden, um einen GPRS-Kontext zu generieren. Die Zeilen mit „OK“ sind die Antworten, die vom Modul kommen.

```
AT+CPIN=7453<CR>
OK
AT+CGDCONT=1,"IP","internet68"
<CR>
```

Services de la ligne	Statut	
Accès WAP CSD	Inactif	<input type="checkbox"/>
Accès WAPWEB 2G	Actif	<input checked="" type="checkbox"/>
Accès WAPWEB 3G+	Actif	<input checked="" type="checkbox"/>
Appels depuis l'étranger	Inactif	<input type="checkbox"/>
Appels vers l'étranger	Inactif	<input type="checkbox"/>
Control Parental	Inactif	<input type="checkbox"/>
Renvoi d'appels	Inactif	<input type="checkbox"/>
Secret Permanent	Inactif	<input type="checkbox"/>

Bild 8. Datendienst aktiviert.

```
OK
AT#esntp="smtp.mail.yahoo.fr"<CR>
OK
AT#euser="gpstracker74"<CR>
OK
AT#epassw="258369"<CR>
OK
AT#eaddr="gpstracker74@yahoo.fr"<CR>
OK
AT#esav<CR>
OK
```

Mit den obigen Befehlen wird der GPRS-Kontext auch gleich gesichert. Anschließend aktiviert man den Kontext mit dem folgenden Befehl:

```
AT#SGACT=1,1<CR>
#SGACT: 10.189.67.153
OK
```

Das GM862-GPS-Modul gibt also die erhaltene IP-Adresse aus. Diese benötigt man zum Versand von E-Mails. Hierzu dient der folgende Befehl:

```
AT#EMAILD="meine_adresse@meine_domain.de", "Test", 0<CR>
```

Selbstverständlich sollte dabei „meine\_adresse@meine\_domain.de“ durch die eigene Mail-Adresse ersetzt werden. Das Modem antwortet mit einem Prompt „>“, nach dem man die Nachricht (abgeschlossen mit „Ctrl-Z“) senden kann.

```
> Dies ist eine Test-Nachricht.
[Ctrl-Z]
```

Bevor die Datei „110267-I\_GM862-GPS\_ATM18\_P4.bas“ kompiliert wird, müssen noch einige Konstanten geändert werden (siehe **Listing 1**). Die meisten sind selbsterklärend. Die vier Passwörter bestehen aus sieben alphanumerischen Zeichen (ohne

Tabelle 2. Notwendige Angaben für das Senden von E-Mails per Mobilfunk.

Parameter	Beispiel-Daten	Eigene Daten
SMTP-Server	smtp.mail.yahoo.fr	
E-Mail-Adresse des Senders	gpstracker74@yahoo.fr	
E-Mail-Adresse des Empfängers	contact@adelek.fr	
E-Mail-Betreff	TEST	
E-Mail-Inhalt	Test GPRS feature, hello World !	
APN (Access Point Name) – abhängig vom Provider	internet68	
Benutzername für E-Mail-Account (User ID)	gpstracker74	
Passwort für E-Mail-Account	258369	

Sonderzeichen). Ihre Bedeutung wird nachfolgend erklärt.

Nachdem das Programm kompiliert und das Resultat in den Controller übertragen wurde, kann man die Elektronik in ein Auto einbauen und testen. Hierzu bittet man einen Bekannten darum, eine kleine Spitztour mit dem so überwachten Auto zu unternehmen. Währenddessen kann man zuhause schauen, ob die Sache funktioniert.

Es sind zwei Abfrageverfahren implementiert. Die erste Methode besteht darin, dass man der Elektronik eine SMS schickt, die das zuvor definierte Passwort (im Beispiel „T090471“) enthält. Kurz darauf wird man eine SMS-Antwort (siehe **Bild 9**) mit den folgenden Informationen erhalten: Breite, Länge, einen Link zu Google Maps, Datum, Zeit, Geschwindigkeit und die Anzahl der empfangenen Satelliten. Empfängt man die SMS mit einem Smartphone, kann man direkt auf den Link klicken und bekommt in Google Maps die Position des Autos mit dem bekannten Pin-Icon angezeigt (siehe **Bild 10**). In dieser URL bedeutet der Ausdruck „t=m“ dass nur eine Karte (m = Map) und weder Satellitenbild noch eine Kombination angezeigt wird. Der Ausdruck „z=10“ setzt den Zoom-Faktor auf den Wert 10.

Die zweite Methode besteht im Verschicken einer SMS mit dem anderen Passwort (im Beispiel „E090471“). Als Antwort erhält man die gleichen Informationen – dieses Mal aber per E-Mail statt SMS, die an die Adresse „Email1“ verschickt wird (siehe **Bild 11**). Im Beispiel ist dies die Adresse „contact@adelek.fr“. In der Betreffzeile der E-Mail ist die Zeit angegeben, bei der die Daten von den Satelliten empfangen wurden.

Man kann auch einer anderen (Vertrauens-)Person Zugang zur GPS-Überwachung geben. Für diesen Zweck sind zwei weitere Passwörter (im Beispiel „T180676“ und „E189676“) vorgesehen. Diese Person erhält die Antworten dann auf ihrem Handy (Phone2) oder unter ihrer E-Mail-Adresse (Email2).

Diese Software nutzt das LCD im Gegensatz zur vorigen überhaupt nicht. Stattdessen wird der Status durch fünf LEDs signalisiert. Bild 1 zeigt, wie diese angeschlossen sind.



Bild 9. Geo-Daten via SMS.

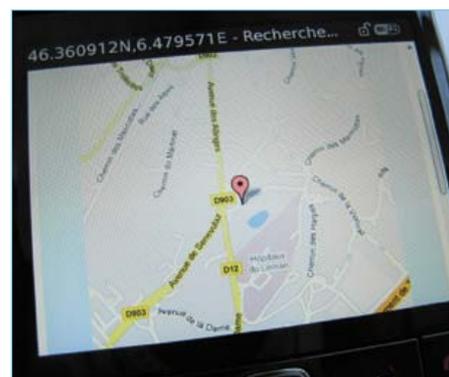


Bild 10. Aktuelle Position des Autos.

#### Listing 1.

```
Const Decal = 1          `UTC + 1
`
Const Apn = «internet68» `APN
Const Esntp = «smtp.mail.yahoo.fr» `EMAIL SENDER SMTP
Const Euser = «gpstracker74» `EMAIL SENDER LOGIN
Const Epassw = «258369» `EMAIL SENDER PASSWORD
Const Eaddr = «gpstracker74@yahoo.fr» `EMAIL SENDER NAME
`
Const Passw1t = «T090471» `PASSWORD USER1
Const Passw1e = «E090471»
Const Phone1 = «0682834725» `PHONE USER1
Const Email1 = «contact@adelek.fr» `EMAIL USER1
`
Const Passw2t = «T180676» `PASSWORD USER2
Const Passw2e = «E180676»
Const Phone2 = «06XXXXXXXXX» `PHONE USER2
Const Email2 = «stephanie.b@free.fr» `EMAIL USER2
`
Const Code_pin = «7453» `SIM PIN
```

#### Basisstationen aufzeichnen

Jedes Handy ist während des Betriebs in ein Mobilfunk-Netzwerk eingeloggt. Dieses Netzwerk besteht aus Basisstationen, die jeweils die in ihrer Zelle vorhandenen Handys registrieren. Wenn man mit dem Handy unterwegs ist, wird die Verbindung von einer Zelle zur nächsten weiter gereicht, sodass man dennoch unterbrechungsfrei weiter telefonieren kann. Zumindest meistens...

Jede Basisstation übermittelt ihren eigenen individuellen ID-Code im Datenstrom zum jeweiligen Handy. Anhand der empfangenen Basisstationen kann man die ungefähre Handy-Position berechnen. Der ID-Code besteht aus zwei Zahlen: dem LAC (Local Area Code) und der CI (Cell Identity), die die Nummer der Zelle angibt. Wenn man sich einer

anderen Basisstation annähert, übernimmt diese die Kommunikation und die Kombination von LAC und CI ändert sich. Es ist kein Problem, diese Daten vom ATM18-Modul aufzeichnen zu lassen. Neben der Erzeugung eines Bewegungsprofils kann man damit auch eine Datenbank der in der eigenen Region vorhandenen Basisstationen anlegen.

Standardmäßig ist die Aufzeichnung dieser Codes deaktiviert. Der Befehl „AT+CREG=2“ aktiviert diese Option. Für diese Anwendung braucht es ebenfalls spezielle Software, die in der Datei „110267-L\_GM862-GPS\_ATM18\_P5.bas“ enthalten ist. Mit dieser Software gibt das ATM-18-Modul an seinem Pin PC0 bei jedem Zellenwechsel entsprechende Daten mit einer Baudrate von 9.600 Bd aus (siehe Bild 1), die man mit einem Terminal-Programm auf

Tabelle 3. Bedeutung der LEDs beim Programm 110267-1_GM862-GPS_ATM18_P4.bas.	
LED	Bedeutung
1	Weniger als vier Satelliten empfangen – die Daten der CGA-Frames sind ungültig. Oder: Die Verarbeitung ist noch nicht abgeschlossen.
2	Entweder werden Konfigurationsbefehle zum GM862-GPS-Modul übertragen, oder aber es wird kein Mobilfunk-Netzwerk gefunden.
3	Warten auf eine SMS.
4	Es wird eine SMS als Antwort auf eine Anfrage für Positionsdaten verschickt.
5	Es wird eine E-Mail als Antwort auf eine Anfrage für Positionsdaten verschickt.

Tabelle 4. Bedeutung der LEDs beim Programm 110267-1_GM862-GPS_ATM18_P5.bas.	
LED	Bedeutung
1	Weniger als vier Satelliten empfangen – die Daten der CGA-Frames sind ungültig. Oder: Die Verarbeitung ist noch nicht abgeschlossen.
2	Entweder werden Konfigurationsbefehle zum GM862-GPS-Modul übertragen, oder aber es wird kein Mobilfunk-Netzwerk gefunden.
3	Warten auf die Verbindung mit einer neuen Basisstation.
4	Empfang von LAC & CI samt Signalqualität einer neuen Basisstation und deren serielle Ausgabe zusammen mit den Positions-Daten.

einem angeschlossenen PC mitschreiben kann, sodass die erzeugten Daten direkt in einer Datei landen. Die Daten sind wie folgt strukturiert:



Bild 11. Geo-Daten via E-Mail.



Bild 12. 27 Basisstationen entlang einer Tour über 40 km.

```

-----
Start with BTS : +CREG: 2,1,"296A", "4437"
Signal Quality : +CSQ: 30,0
13/03/2011
16h16m15.000s
SAT:05 ALT:462.1M
LAT:46deg21'40.9``N
LON:006deg28'44.5``E
http://maps.google.com/maps?q=46.361347N,6.479033E&t=m&z=10
-----
-----
BTS : +CREG: 1,"296A", "28D1"
Signal Quality : +CSQ: 21,0
13/03/2011
16h17m02.000s
SAT:08 ALT:464.4M
LAT:46deg21'40.8``N
LON:006deg28'44.5``E
http://maps.google.com/maps?q=46.361320N,6.479023E&t=m&z=10
-----
-----
BTS : +CREG: 1,"296A", "A2B3"
Signal Quality : +CSQ: 21,0
13/03/2011
16h21m52.000s
SAT:08 ALT:448.7M
LAT:46deg21'33.9``N
LON:006deg27'48.8``E
http://maps.google.com/maps?q=46.359409N,6.463560E&t=m&z=10
-----

```

Auch diese Software verwendet kein LCD, stattdessen wird der aktuelle Status mit vier LEDs signalisiert (siehe Tabelle 4). Das Programm wurde bei einer Fahrt über 40 km getestet (siehe Bild 12). Es konnten dabei insgesamt 27 Basisstationen entlang der

**Weblinks**

- [1] [www.elektor.de/110139](http://www.elektor.de/110139)
- [2] [www.elektor.de/071035](http://www.elektor.de/071035)
- [3] [www.elektor.de/071148](http://www.elektor.de/071148)
- [4] [www.telit.com/en/products/gsm-gprs.php?p\\_ac=show&p=7](http://www.telit.com/en/products/gsm-gprs.php?p_ac=show&p=7)
- [5] [www.sparkfun.com/products/281](http://www.sparkfun.com/products/281)
- [6] [www.elektor.de/products/kits-Module/kits/080213-71-cc2-avr-board.35.441731.lynkx](http://www.elektor.de/products/kits-Module/kits/080213-71-cc2-avr-board.35.441731.lynkx)
- [7] [www.hw-group.com/products/hercules/index\\_en.html](http://www.hw-group.com/products/hercules/index_en.html)
- [8] [www.telit.com/en/products/download-zone.php](http://www.telit.com/en/products/download-zone.php)
- [9] [www.elektor.de/110267](http://www.elektor.de/110267)

**Tabelle 5. Identifikations-Codes von Basisstationen** (10 der insgesamt 27 Datensätze, die bei einer 40 km langen Tour (siehe Bild 12) gesammelt wurden).

LAC	CI
296A	1D43
296A	B93B
296A	F11D
296A	94CC
296A	B93B
296A	94CC
296A	B93B
296A	922B
296A	94CC
296A	C2B8

Route identifiziert werden (siehe **Tabelle 5**). Der Autor hofft, mit diesen Anwendungen Ihr Interesse geweckt zu haben und freut

sich über Erfahrungsberichte, die man ihm an seine E-Mail-Adresse „tellme@adelek.fr“ schicken kann.

(110267-1)

— Anzeige



**Elektor Electronic Toolbox**

**Erhältlich im App Store**

**Umfangreiche Elektroniker-App von Elektronikern für Elektroniker**

Die neue „Elektor Electronic Toolbox“-App ist ganz auf die Belange von Elektronikern zugeschnitten. 28 Einzelprogramme/Anwendungen können über eine übersichtliche Oberfläche ausgewählt werden. Sehr hilfreich im Entwickleralltag sind die Datenbanken für die Bauteilgruppen Bipolar-Transistoren, FETs, Triacs, Thyristoren, Dioden und ICs. Ein Bauteil kann anhand der Typenbezeichnung kinderleicht ausgewählt werden – eine Internetverbindung ist nicht notwendig. Insgesamt sind über 45.000 Bauteile in den Datenbanken verzeichnet. Hinzu kommt eine Spezialdatenbank, in der die Belegung einer Vielzahl von Steckverbindern aus den Bereichen Audio & Video, Computertechnik und Telefon nachgeschlagen werden kann. Nützlich sind auch die interaktiven Bauteilwert-Kalkulatoren.

Tools wie eine virtuelle Widerstandsuhr, ein Umrechner zwischen Maßeinheiten, eine Schaltsymboldatenbank und vieles mehr runden die Elektor-App ab.

Die neue „Elektor Electronic Toolbox“ (geeignet für iPhone, iPod und iPad) kann zum Preis von nur 4,99 Euro heruntergeladen werden.



**PCB-Pool**  
Beta LAYOUT

DAS ORIGINAL SEIT 1994

**NEU!** gut bestückt!  
Schon ab einem Bauteil

**cool!** ALU-Kern Leiterplatten

**dichter!** 5mil track / 8mil drill

**knackiger!** Ritztechnik

**kostenlos!** Free Stencil

Alle emporgelagerten Warenmarken sind emporgelagerte Warenzeichen der jeweiligen Hersteller.

www.beta-layout.com

**Beta**  
LAYOUT

# MIAC – ein PIC für alle Fälle

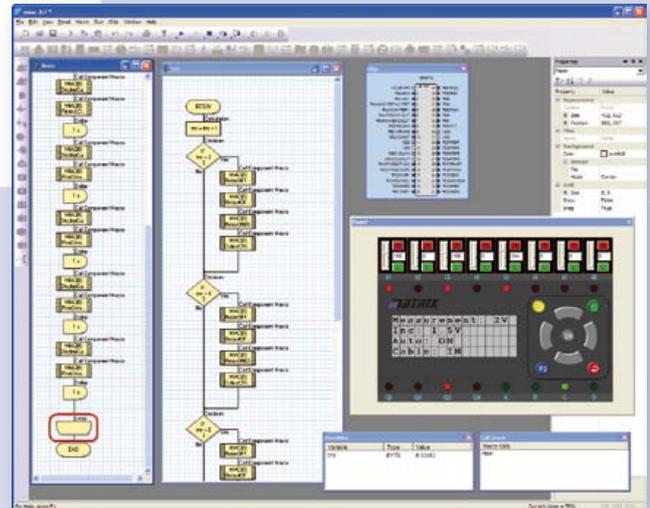


## Ausstattung/Anschlüsse

- |                                    |  |
|------------------------------------|--|
| 1. Hutschienenmontage              | 10. USB-Transfer-LED                   |
| 2. LCD (4x16 Zeichen)              | 11. Tastenbedienfeld                   |
| 3. Power-LED                       | 12. M3-Montagelöcher                   |
| 4. Status-LEDs für Eingänge        | 13. Status-LEDs für Motoren            |
| 5. Stromversorgungsbuchse (2,1 mm) | 14. Schraubklemmen für Motoren         |
| 6. Schraubklemmen für Eingänge     | 15. Hutschienen-Arretierung unten      |
| 7. Hutschienen-Arretierung oben    | 16. Schraubklemmen für Relais-Ausgänge |
| 8. Reset-/Start-Taste              | 17. Status-LEDs für Relais-Ausgänge    |
| 9. USB-Buchse                      |  |

## Was ist ein MIAC?

MIAC ist eine Steuerung in Industriequalität, die sich für eine breite Palette elektronischer Systeme eignet. Neben dem großen Einsatzgebiet bei industriellen Steuerungen und in der Automation eignet sich MIAC auch perfekt für die private Nutzung als Steuerung in PIC-Mikrocontroller-Projekten, die etwas Pepp vertragen könnten.



**Flowcode** – die grafische Programmiersprache (gehört zum Lieferumfang von MIAC)



MIAC ermöglicht mit dem optional erhältlichen FlowKit-Modul (Art.-Nr. HP299) volles In-Circuit-Debugging in Flowcode

## Vorteile

- Flexibel und erweiterbar
- Mit Flowcode, C oder Assembler leicht programmierbar
- Mechanisch und elektrisch sehr robust

## Features

- Programmierbar via USB
- Basierend auf PIC18F4455
- Programmiersoftware Flowcode (im Wert von 180 €) im Lieferumfang inbegriffen
- Kompatibel mit C-Compilern von Drittherstellern
- 8 digitale oder analoge Eingänge
- 4 Relais-Ausgänge mit je 10 A Belastbarkeit
- 4 gesteuerte Motor-Ausgänge (500 mA)
- LCD mit 4 Zeilen à 16 Zeichen
- 9 Steuertasten im Bedienfeld

Realisieren Sie jetzt Ihr PIC-Projekt mit dem vorteilhaften MIAC-Bundle – bestehend aus einem MIAC-Modul sowie der prämierten Programmiersoftware „Flowcode 3 für PIC“ (Professional-Version)!



### Bundle-Preis:

Nur 185,00 Euro (251,60 CHF)

Jetzt bestellen unter

[www.elektor.de/miac-bundle](http://www.elektor.de/miac-bundle)

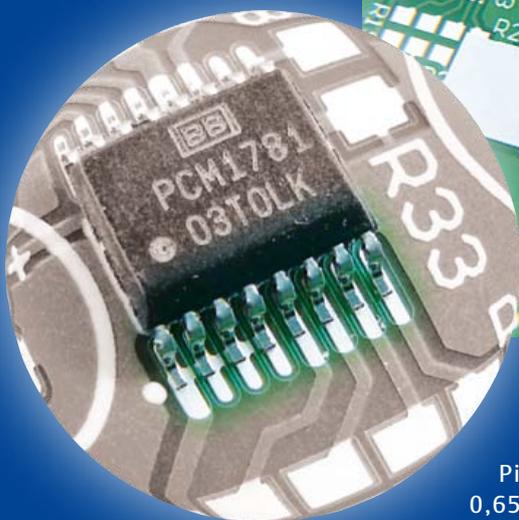
# Platinenlayout: Kleinheiten und Feinheiten!

Von Thijs Beckers (Redaktion NL)

Vielleicht haben Sie die Artikel zu unserem DSP-Kurs im letzten Monat und in dieser Elektor-Ausgabe gelesen (sonst sollten Sie dies tun!). In den ersten beiden Folgen werden die in dem DSP steckenden Fähigkeiten sowie die Software beschrieben, die zum Programmieren nötig ist. Der dritte, im September erscheinende Teil stellt die Praxis in den Vordergrund, dann werden wir die Schaltung mit der zugehörigen Platine veröffentlichen.

Die Platine ist mit SMD-ICs in unterschiedlichen Bauformen bestückt. Die Opamps sind in Gehäusen vom Typ SOIC (*Small-Outline Integrated Circuit*) mit dem Pin-Abstand 1,27 mm untergebracht, und der DSP-Chip hat die

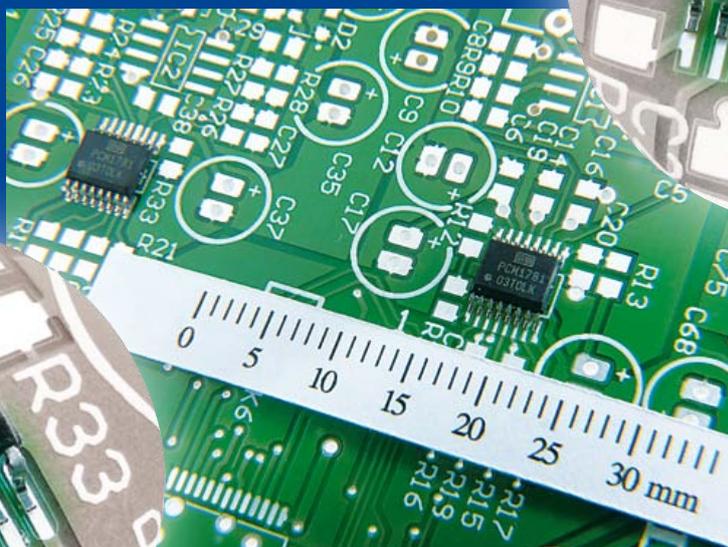
(*Shrink Small-Outline Package/Quarter-Size Small-Outline Package*), die dem TSSOP-Gehäuse des A/D-Wandlers sehr ähnlich ist.



Bauform TQFP (*Thin Quad Flat Pack*) mit 52 Pins im Abstand 0,65 mm. Den gleichen Abstand haben die Pins des Gehäuses TSSOP (*Thin Shrink Small-Outline Package*), das den A/D-Wandler beherbergt.

Noch etwas graziler kommt der Sample-rate-Wandler daher, bei seinem TQFP-Gehäuse beträgt der „Pitch“, das ist hier der englische Begriff für „Abstand“, nur 0,5 mm. Das allein war schon Grund genug, auch für dieses Projekt ein gebrauchsfertig bestücktes Platinenmodul in das Elektor-Shop-Programm aufzunehmen. Die Sorge, die winzigen, von den Herstellern leider nicht größer gefertigten „Käferchen“ erfolgreich auf die Platine zu montieren, ist damit ausgeräumt.

Als Wanderer zwischen zwei Welten kann der D/A-Wandler von Texas Instruments betrachtet werden, der das letzte Glied der Signalkette bildet. Dieser Chip hat die Bauform SSOP/QSOP



Der Pin-Abstand beträgt hier jedoch 0,635 mm, wobei die Differenz von 0,015 mm, das sind fünfzehn Tausendstel eines Millimeters, mit dem bloßen Auge nicht mehr wahrgenommen werden kann. Der minimale Unterschied macht sich

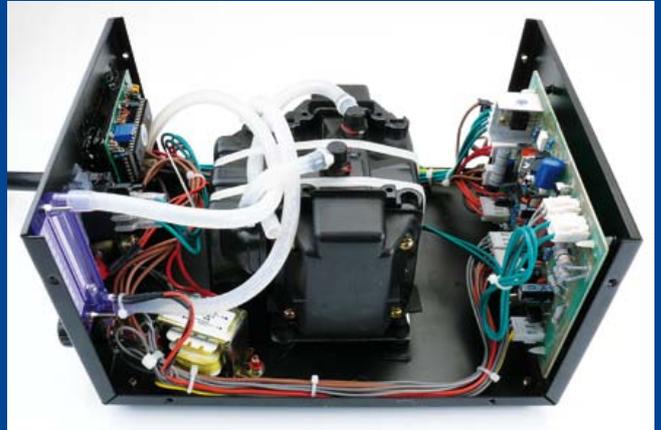
erst bemerkbar, wenn der D/A-Wandler zum Vergleich auf seinen eigenen Platz (links im Bild, Pitch 0,635 mm) und auf den Platz des A/D-Wandlers (rechts, Pitch 0,65 mm) gelegt wird.

Starkes Hineinzoomen beweist, dass die Pins perfekt mit den Löt pads übereinstimmen, wenn der D/A-Wandler auf dem für ihn bestimmten Platz liegt. Dagegen beträgt die Differenz auf dem Platz des A/D-Wandlers bis zu einer halben Pin-Breite. Es liegt auf der Hand, dass hier eine erfolgreiche Lötmontage wenig wahrscheinlich wäre. Zwar ist sie nicht unmöglich, doch die Gefahr von Kurzschlüssen zwischen den Pins ist verglichen mit dem exakten Platinen-Pinlayout um ein Vielfaches höher.

Die „Alten Hasen“ unter den SMD-Platinen-Designern werden jetzt vielleicht mit den Schultern zucken und wohlwollend lächelnd äußern: „Das weiß doch jeder!“. Doch wer weiß schon alles? Ich muss gestehen, dass ich eine SMD-Gehäuse-Variante mit dem Pin-Abstand 0,635 mm bisher nicht kannte. Ein weiterer Beweis dafür, dass der Mensch nie auslernt...

(110394)gd

# Scirocco im Kleinformat



Von Thijs Beckers (Redaktion NL)

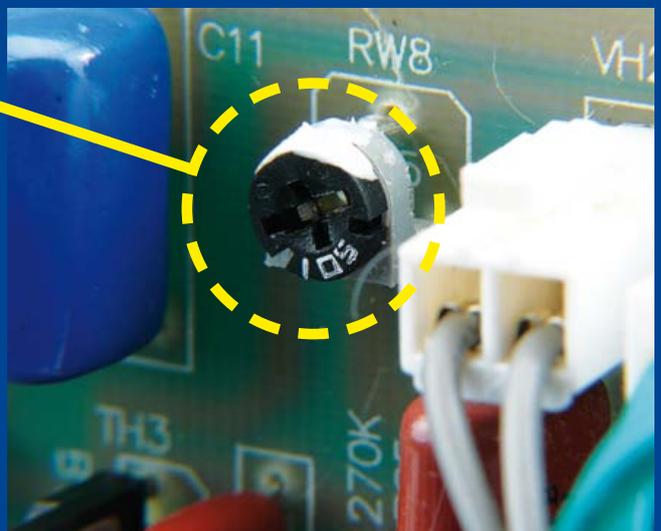
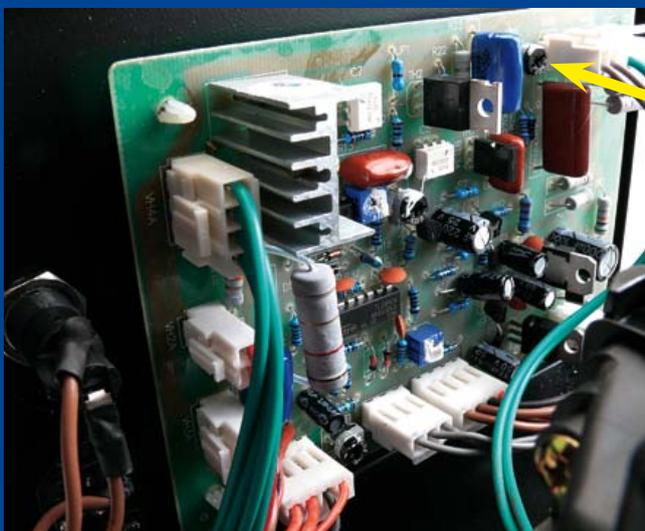
Laborkollege Luc Lemmens hat kürzlich eine Heißluft-Lötstation für seinen privaten Gebrauch erstanden. Im Elektor-Labor arbeiten wir seit langem erfolgreich mit der Aoyue 852A+. Auch nach meiner eigenen Erfahrung möchte ich auf das Löten und Entlöten von SMDs mit Heißluft nicht mehr verzichten. Verglichen mit meiner „alten“ Lötstation ist die Heißluft ein echter Fortschritt. Vor allem das Entlöten ist mit konventionellem Lötgerät ohne passende Entlötdapter so gut wie unmöglich. Die Heißluft-Lötstation macht diese Arbeit fast zum Kinderspiel, die Düse muss nur so lange über die SMD-Pins bewegt werden, bis das Zinn geschmolzen ist. Meistens genügen fünf Sekunden.

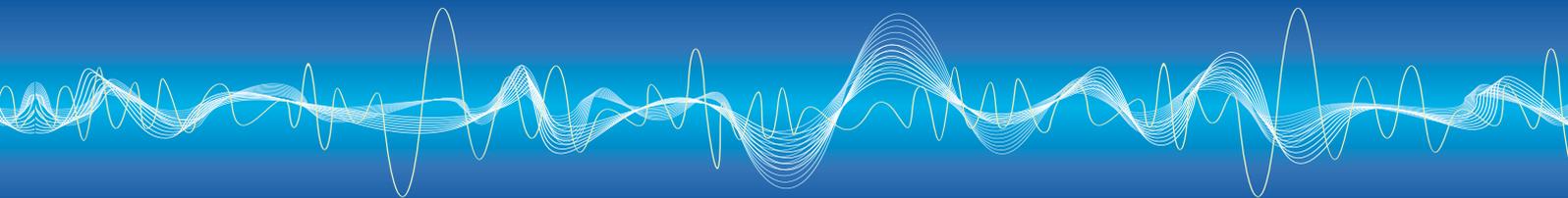
Kollege Luc hatte sich für eine Heißluft-Lötstation Aoyue 852 (ohne „A+“ dahinter!) entschieden. Ein gutes Stück zu einem akzeptablen Preis. Dieser Eindruck blieb ungetrübt, bis die heiße Luft aus der Düse den ersten zu montierenden SMD-Widerstand traf. Dagegen war ein Scirocco der Stärke Fünf fast ein laues Lüftchen, der Widerstand wurde auf Nimmerwiedersehen vom Tisch geblasen. „Die Luftmenge muss niedriger eingestellt wer-

den“, folgerte der Kollege messerscharf, doch das war leichter gesagt als getan. Der Stellknopf stand bereits am Linksanschlag.

Kollege Luc ist nicht der Mensch, der einfach zurückgibt, was er so nicht gebrauchen kann. Zuerst wird überlegt, wie dem Problem auf anderem Weg abgeholfen werden kann. Also wurde das Gerät geöffnet. Zum Vorschein kam ein geordnetes Innenleben mit einer soliden Platine, einem leistungsstark wirkenden Kompressor und einem Standard-Display. Natürlich gehörte auch ein Trafo dazu, uns fiel aber auf, dass er kein CE-Zeichen trug. Die nähere Betrachtung der Platine brachte die Erkenntnis, dass dem Luftstrom-Potentiometer ein Trimpoti vorgeschaltet ist, das Foto zeigt es in voller Größe. Konnte der Luftstrom hier noch weiter reduziert werden?

Tatsächlich, wir hatten ins Schwarze getroffen! Die optimale Einstellung war schnell gefunden, das Gerät wurde wieder zugeschraubt. Zuvor hatten wir noch die Potentiometer gegeneinander getauscht, mit denen der Luftstrom und die Lufttemperatur eingestellt werden. Die Bedienknöpfe sind jetzt logischer angeordnet, an der Anzeige beider Werte auf dem Display hat





sich natürlich nichts geändert. Die Drehknopf-Beschriftungen stimmen nun zwar nicht mehr, doch Luc fand die verbesserte Bedienung wichtiger.

Die Heißluft-Lötstation ist nun so, wie sie sein soll. Die Rücksendung an den Lieferanten hat sich erübrigt, ein Austausch hätte vermutlich ohnehin länger gedauert, und außerdem wäre auch nicht sicher gewesen, dass sich das zweite Exemplar vom ersten unterscheidet. Solche „Eigenmächtigkeit“ führt allerdings meistens zum Erlöschen der Garantie. Für Luc ist auch das kein Problem, falls das Gerät irgendwann einmal streiken sollte, geht er wieder selbst ans Werk...

Nachzutragen ist noch, dass die übrigen Trimpotis der Heißluft-Lötstation nicht verstellt werden dürfen, sie sind für die

Temperaturregelung und weitere Funktionen zuständig. Die Regelung der Temperatur ist das Stichwort für eine Frage, die wir noch nicht beantworten konnten: Wie lässt sich die Lufttemperatur einer Heißluft-Lötstation kalibrieren?

Falls Sie die Antwort auf diese Frage kennen, würden wir uns über eine Nachricht an [redaktion@elektor.de](mailto:redaktion@elektor.de) freuen!

(110261)gd

Und natürlich gilt: Beim Modding von netzbetriebenen Geräten sollte man Vorsicht walten lassen!

## Zurück zur Erde

Von Thijs Beckers (Redaktion NL)

Auf der Suche nach Beiträgen für unsere sommerliche Doppelausgabe (das „Halbleiterheft“) war unser jüngster Kollege Raymond Vermeulen mit einem Schaltungsvorschlag eines belgischen Lesers beschäftigt. Die eingesandte (Mini-)Schaltung hat den Zweck, den korrekten Anschluss des Schutzleiters an einer Netzsteckdose zu überprüfen. Eine Glimmlampe leuchtet auf, wenn die Steckdose ordnungsgemäß installiert ist. Doch mit der Schaltung schien irgendetwas nicht zu stimmen.

Um die Schaltung zu testen, hatte Raymond sie an ein Gerätekabel nach IEC-Standard angeschlossen (siehe Foto). Der Test in unserem Labor bestätigte die korrekte Funktion. Wenn die Schaltung an einer beliebigen anderen Steckdose außerhalb des Labors angeschlossen wurde (bei uns sind alle Steckdosen korrekt installiert!), leuchtete die Glimmlampe manchmal auf - und manchmal nicht. Es war schnell klar, dass dieses Verhalten davon abhing, wie herum Phase und Nullleiter angeschlossen waren. Nur an den Steckdosen im Labor trat dieses Verhalten nicht auf. Eine Email-Rückfrage beim Einsender konnte das Phänomen nicht klären. Er war sich absolut sicher, dass die „Polung“ von Phase und Nullleiter keine Rolle spielen sollte ...

Inzwischen haben wir des Rätsels Lösung gefunden: Im Elektor-Labor werden alle Steckdosen über einen Trenntrafo mit Strom versorgt, während die übrigen Steckdosen des Gebäudes galvanisch mit dem Stromnetz verbunden sind. Die Schutzleiter sind aber nicht unterschiedlich. Bei den Steckdosen im Labor sind

die 230-V-Leitungen netzgetrennt und somit „schwebend“, der Nullleiter ist erdfrei. So können Phase oder Null nicht zugeordnet werden. Die kapazitive Kopplung mit dem Schutzleiter genügt schon, um die Glimmlampe aufleuchten zu lassen.

Kollege Raymond kam zu dem Schluss, dass die Schaltung an einer „normalen“ Steckdose in dem einen Fall einen Tiefpass bildet, der die Wechselfrequenz bei Netzfrequenz passieren lässt. Im anderen Fall (anders herum eingesteckt) bildet die Schaltung einen Hochpass mit einer Eckfrequenz weit über der Netzfrequenz, so dass die Neonlampe einmal leuchtet und einmal nicht.

Die super einfache Schaltung prüft also den Schutzleiteranschluss und funktioniert gleichzeitig als Phasentester. Bleibt noch die Frage, wie die Steckdosen des Einsenders mit Strom versorgt werden, denn ihm zufolge leuchtet die



Glimmlampe, korrekte Steckdosen-Installation vorausgesetzt, unabhängig von der Polung auf. Könnte es sein, dass im Stromnetz am Ort des Einsenders Phase und Nullleiter nicht unterscheidbar sind? Wir wissen es nicht. Über die Antwort eines fachkundigen Lesers auf diese Frage, gerichtet an [redaktion@elektor.de](mailto:redaktion@elektor.de), würden wir uns freuen.

Die Schaltung werden wir unseren Lesern natürlich nicht vorenthalten, achten Sie bitte auf das kommende Halbleiterheft! Vielleicht wissen Sie schon jetzt, wie diese Schaltung aussieht?

(110396)gd

# Radmitnehmerschrauben beim ElektorWheelie

Von Jan Visser (Elektor Labor)

Als Mitarbeiter des Elektor-Labors werde ich auch mit technischen Problemen konfrontiert, die von Seiten der Leser oder auch von Kollegen an mich herangetragen werden. An dieser Stelle möchte ich vom „ElektorWheelie“ berichten, den Elektor im Juli/August und September 2009 vorgestellt hat (siehe auch [1]).

Bei intensivem Einsatz des Rollers können die Radmitnehmerschrauben (**Foto rechts unten**) infolge der starken Beanspruchung oder durch übermäßiges Schwingen verformt werden



gereicht montieren zu können, und setzen Sie die Konstruktion mit den Hülzen in die Radaussparungen ein.

Achten Sie darauf, dass die Schrauben und das Mitnehmerblech nicht am Rad festkleben, sie müssen sich später problemlos abnehmen lassen.

Geben Sie dem Kleber eine Nacht Zeit, um abzubinden, danach demontieren Sie das Mitnehmerblech einschließlich der Schrauben vom Rad. Die Hülzen sind nun mit korrektem Sitz fest mit dem Rad verbunden. Die Stabilität wird zusätzlich erhöht, wenn

Sie den Hohlraum in den Aussparungen mit Kleber ausfüllen. Dazu ist sowohl Zweikomponentenkleber als auch Baukleber geeignet.

Wenn der Kleber vollständig gehärtet ist, können Sie die Konstruktion am ElektorWheelie montieren. Vergessen Sie die zentrale Mutter nicht, die auch das Radmitnehmerblech auf der Achse festhält. Sichern Sie die Mutter mit Schraubensicherungslack, damit sie sich nicht losdrehen kann.

Durch diese unkomplizierte Maßnahme werden die Kräfte und Schwingungen verringert, die auf die Radmitnehmerschrauben einwirken. Das Rad wird ohne Spiel mitgenommen, die Schrauben können sich nicht mehr verformen oder brechen.

(110395)gd

## Weblinks

[1] [www.elektor.de/wheelie](http://www.elektor.de/wheelie)

[2] [www.conrad.de](http://www.conrad.de)

oder sogar brechen. Um dem wirksam vorzubeugen, rate ich Wheelie-Besitzern, die folgenden Maßnahmen durchzuführen.

Nehmen Sie vier Messing- oder Alu-Hülzen, Innendurchmesser 5 mm, Außendurchmesser 6 mm, Länge 25 mm. Die Hülzen können Sie auch von einem längeren Rohr absägen. Mit diesen Abmessungen passen die Hülzen exakt in die Aussparungen der Räder am ElektorWheelie. Geeignete Messing- oder Alu-Rohre sind in Modellbauläden und eventuell auch in Baumärkten erhältlich. Auch der Elektronik-Sortimenter Conrad [2] hat das Passende in seinem Angebot, suchen Sie dort nach der Artikelnummer 297321.

Die Messing- oder Alu-Hülzen müssen in die Aussparungen geklebt werden (**Foto links oben**). Dazu tragen Sie auf den Außenseiten der Hülzen Zweikomponentenkleber auf. Orientieren Sie sich am Mitnehmerblech, um die Hülzen pass-



# Stellaris ARM-Controller & Biologie

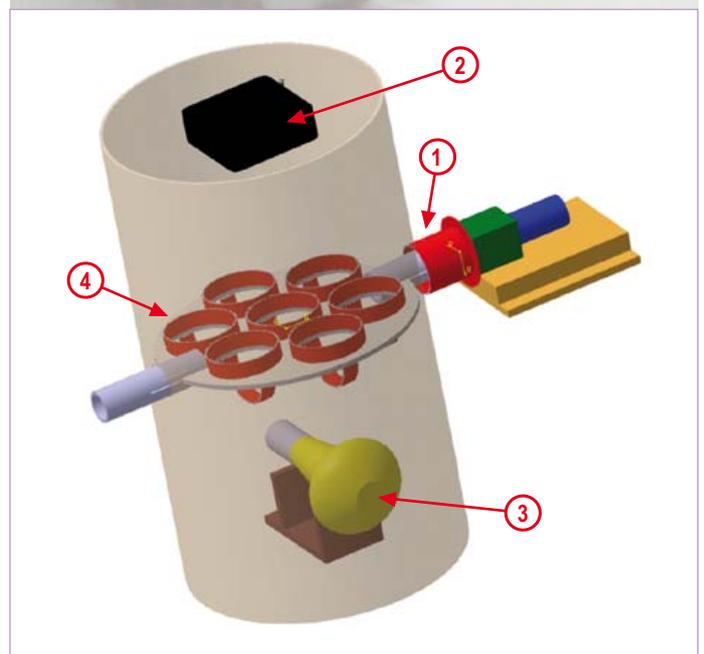
## Design-Überlegungen für einen Eier-Inkubator



Von Tianyu Chen (China)

Diese Mikrocontrollerschaltung ist die Antwort auf die Anforderungen biologischer Prozesse. Obwohl es dabei um ein zuverlässiges System zum Ausbrüten von Hühnereiern geht, hat der verwendete Ansatz auch genug Potential für alle Fälle, wo es um das Überwachen und Steuern von Parametern wie Temperatur, Bewegung und Luftfeuchtigkeit geht.

Bild 1. So sieht das vom Autor entwickelte Konzept eines Brutkastens für Hühnereier aus: 1. Motor und koaxialer Antrieb, 2. Lüfter, 3. 25-W-Glühlampe und 4. Eierhalter.



Die Nachahmung eines brütenden Huhns mit technischen Mitteln mündet in die Entwicklung eines Inkubators (Brutkastens) und ist ganz sicher nicht trivial. Die Aufgabe verlangt vom Entwickler neben elektronischer auch mechanische Kompetenz und außerdem auch noch Wissen in ovarialer Biologie.

Statt ein ausentwickeltes Fertigprodukt zu präsentieren, zeigt dieser Beitrag, wie bei der Entwicklung die grundlegenden Faktoren von Elektronik, Software und Mechanik analysiert und kombiniert werden können.

### Ei-Ausbrüt-Kriterien

Auf den ersten Blick scheint die Temperatur der wichtigste Faktor beim Ausbrüten eines Eis zu sein. Doch Aspekte wie die Luftfeuchtigkeit und weitere Faktoren sind ebenfalls entscheidend. Nach der Recherche des Autors sind die folgenden fünf Parameter in der Reihenfolge ihrer Relevanz zu berücksichtigen:

1. Temperatur
2. Relative Luftfeuchtigkeit

3. Rotation
4. Belüftung
5. Sauberkeit

**Temperatur:** Da ein Embryo die Temperatur nicht selbst regulieren kann, muss man eine externe Temperatursteuerung vorsehen. Auch wenn die meisten Eier erfolgreich bei Temperaturen zwischen 35 °C und 40,5 °C ausgebrütet werden können, liegt das Optimum doch bei konstanten 37,8 °C. Höhere Temperaturen beschleunigen zwar die embryonale Entwicklung, ziehen allerdings auch eine erhöhte Sterblichkeit nach sich und haben negativen Einfluss auf die Gesundheit der Küken. Von daher sind möglichst geringe Abweichungen von den idealen 37,8 °C anzustreben.

**Relative Luftfeuchtigkeit:** Auch hier gibt es ein mögliches Spektrum der Luftfeuchtigkeit von 40...70 %, doch optimalerweise schränkt man dieses Spektrum auf 50...60 % ein. Bei einer optimalen Luftfeuchtigkeit kann das Ei am Anfang des Brütens besser Wärme

*Leserprojekte sind Beiträge von Elektor-Lesern für experimentelle Zwecke oder zur Anregung für andere Leser. Die in dieser Rubrik vorgestellten Schaltungen wurden vom Elektor-Labor nicht auf Reproduzierbarkeit und Funktion getestet.*

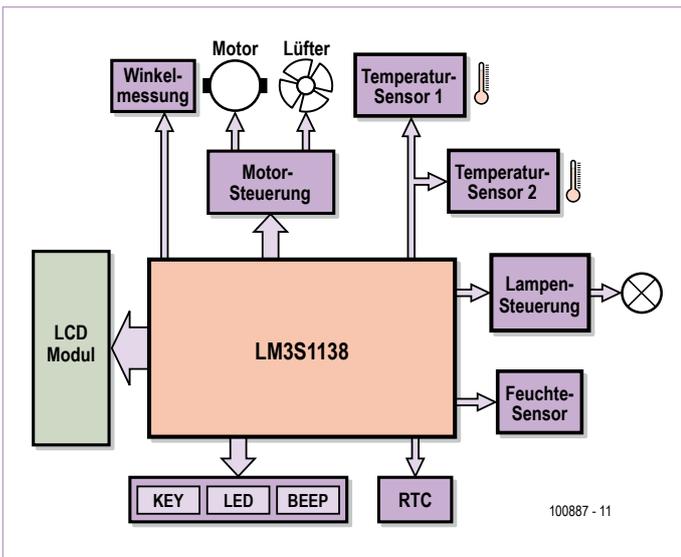


Bild 2. Diese Peripherie macht einen Mikrocontroller zur Brutkastensteuerung.

aus der Umgebung aufnehmen und am Ende des Brutzeitraums an die Umgebung abgeben.

**Rotation:** Die Eier sollten zur Vermeidung des Anhaftens des Embryos und zum Stimulieren der Bewegungen der inneren Eihaut gedreht werden. Ideal wären sechs bis acht Drehungen pro Tag, besonders während der ersten zwei Wochen.

**Belüftung:** Außer am Anfang ist während der Embryonalentwicklung der Luftaustausch der Luftblase wichtig. Mit dem Wachstum des Embryos steigert sich sein Sauerstoffbedarf. Der perfekte Sauerstoffgehalt beträgt 21 % bei einem CO<sub>2</sub>-Gehalt von <0,5 %. Schon bei einem CO<sub>2</sub>-Gehalt >1 % sterben viele Embryonen ab.

**Sauberkeit:** Auch wenn der Embryo von Eiklar umgeben ist und von Häuten und einer Kalkschale geschützt wird, besteht dennoch das Risiko der Infektion durch Mikroorganismen, was eine erfolgreiche Brut beeinträchtigen kann. Es muss daher großer Wert auf Sauberkeit und Desinfektion des Brutkastens gelegt werden.

**Brutkasten im Selbstbau**

Es lohnt sich, einigen Aufwand bei der Suche nach einem geeigneten Behälter zu betreiben, der sich in einen Brutkasten umbauen lässt. Das Behältnis muss nämlich thermisch gut isoliert sein und eine gute Belüftung zulassen. Selbstverständlich sollte es ausreichend Platz für die gewünschte Anzahl von Eiern bieten. Der Autor plante für sieben Eier und entschied sich für eine Art Kreuzung aus Elektrokocher und Thermoskanne.

**Bild 1** vermittelt eine Idee davon, was für eine Anordnung dem Autor vorschwebte. Oben befindet sich ein Lüfter (2) aus einem alten PC. In der Mitte befindet sich eine Ablage für die Eier (4). Dieser Eierhalter wird durch einen koaxial angebrachten Motor bewegt (1), damit man die Eier bei Bedarf in Abständen drehen kann. Die Belüftung ist kritisch: Es muss genug Sauerstoff verfügbar sein und die Lufttemperatur muss stabil gehalten werden. Bei zu viel Frischluft könnte die Temperatur zu schnell fallen. Experimente führten zu einer guten Balance in Form einer gewöhnlichen 25-W-Glühlampe (3), die als Heizung dient. Sobald die Temperatur den Sollwert übersteigt, bläst der Lüfter etwas frische Luft durch ein Loch in der Abdeckung. Da sich frische und warme Luft mischen, erreicht man eine relativ konstante Temperatur mit gutem Sauerstoffgehalt.

**Elektronik...**

Nach den biologischen und mechanischen Aspekten kommt die Elektronik, die das Ganze zum Brüten bringt. Zentraler Baustein ist der Chip LM3S1138 – ein ARM-Cortex-M3-Mikrocontroller [1] der Stellaris-Serie von Luminary Micro (übernommen von Texas Instruments).

Die Blockschaltung (**Bild 2**) zeigt, welche Peripherie (Taster, LEDs und Tongeber) zur Bedienung und Signalisierung an den Controller angeschlossen ist. Nach dem Start werden Eier-Typ und geschätzte Brutzeit abgefragt. Während des Brütens informiert ein LCD über

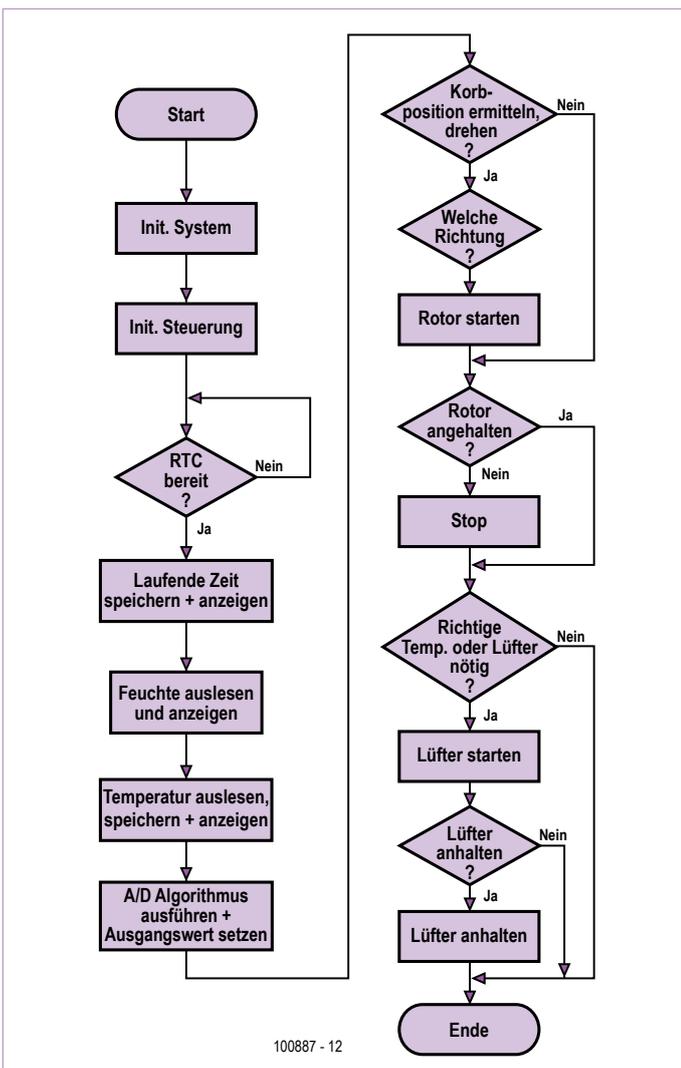


Bild 3. Flussdiagramm des Programms für den LM3S1138-Mikrocontroller (alias digitale Henne).

Temperatur sowie Luftfeuchtigkeit im Brutkasten und zeigt die noch verbleibende Brutdauer in Tagen an. Wenn Temperatur oder Feuchtigkeit vom Sollwertbereich abweichen oder wenn ein unzulässiger Zustand erkannt wird, wird der Hühnerzüchter durch Warn-töne alarmiert, damit er etwas tun kann, um den Tod der Embryos zu verhindern.

Das System verfügt über zwei Temperatur-Sensoren und einen Feuchtigkeits-Sensor. Die beiden Temperatur-Sensoren sollten so angebracht werden, dass sie zuverlässig Unter- oder Übertemperaturen erfassen können und sich nicht unbedingt im direkten Luftstrom des Lüfters befinden. Zwei Sensoren sind zur Sicherheit (Redundanz) vorgesehen, falls einer der beiden ausfallen sollte.

Die Elektronik verteilt sich auf zwei Platinen. Die zentrale Platine beherbergt den Mikrocontroller, das Debugging-Interface, die Reset-Schaltung, Taster und LED. Die Zusatzplatine enthält Peripherie wie RTC und Motor-Treiber. Aus Platzgründen ist die komplette Schaltung auf der Elektor-Webseite zu diesem Beitrag [2] als kostenloser Download verfügbar. Die wichtigsten Bauteile sind:

Controller = LM3S1138, Motor-Treiber = L298N, RTC = DS1305, Feuchtigkeits-Sensor = HS1101/NE555 und die Temperatur-Sensoren = LM75. Mit Ausnahme des Controllers sind es im Wesentlichen Standard-Komponenten.

**...und Software**

Die Software ist in **Bild 3** als Flussdiagramm wiedergegeben. Alle Aspekte der Mechanik, der Elektronik und der wichtigen Faktoren für das Eier-Ausbrüten wurden im Code berücksichtigt. Die Datei mit dem Source-Code für den ARM-Controller LM3S1138 kann ebenfalls kostenlos von [2] herunter geladen werden.

(100887)

**Weblinks**

- [1] LM3S1138-Datenblatt: [www.luminarymicro.com/products/LM3S1138.html](http://www.luminarymicro.com/products/LM3S1138.html)
- [2] Schaltung und Projekt-Software: [www.elektor.de/100887](http://www.elektor.de/100887)

— Anzeige



**e-weekly**  
elektor-newsletter

**Elektor-Newsletter E-weekly jetzt gratis abonnieren!**

Jeden Freitagmorgen erscheint E-weekly, der kostenlose Newsletter von Elektor. Unsere E-weekly-Redakteure halten Sie mit neuesten und interessanten Meldungen, Tipps & Trends aus der Welt der Elektronik auf dem Laufenden. Außerdem werden Sie schnell und umfassend über aktuelle Elektor-Projekte (Nachlesen & Updates) sowie über das umfangreiche Elektor-Sortiment und spezielle Angebote als Erster informiert.

Daneben erhalten E-weekly-Abonnenten exklusiv vollen Zugang zu allen Newsberichten und zu unserem Forum, das von kompetenten Elektronik-Experten moderiert wird.

**Klicken Sie jetzt auf [www.elektor.de/newsletter](http://www.elektor.de/newsletter)!**



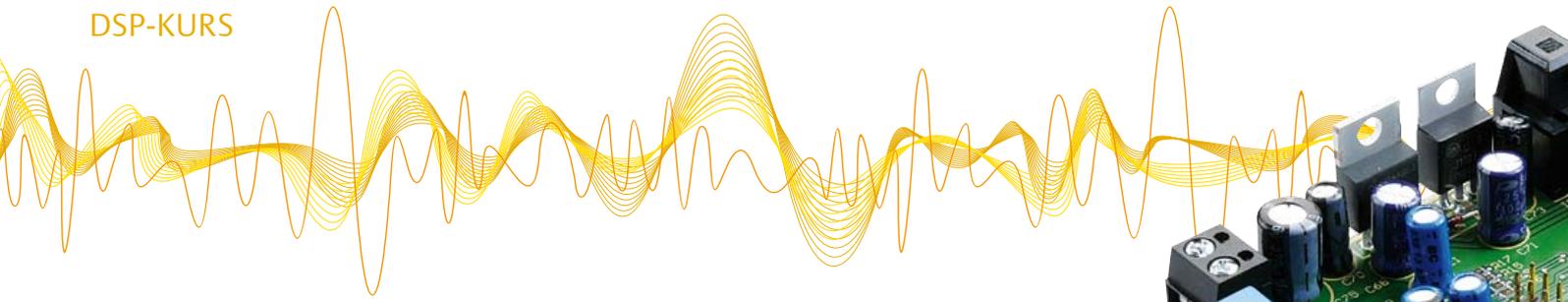
**BESSER GLEICH ONLINE KALKULIEREN.**  
STARRE- UND FLEXIBLE LEITERPLATTEN.

ab 12h  
Leiterplatten-Express! Für Ihre Bestellungen gratis!

**LEITON**  
RECHNEN SIE MIT BESTEM SERVICE

Schluss mit umständlichen Rechenoperationen! Bei uns kalkulieren Sie auch Ihre exotischsten Leiterplatten jederzeit schnell und einfach online. Doch nicht genug: Bei LeitOn gilt die Online-Kalkulation auch für Serien und flexible Leiterplatten! Ebenso einmalig ist der LeitOn Leiterplatten-Expressdienst mit Top-Garantie: Platinen sind gratis bei überschrittenem Liefertermin! Neugierig? Unsere persönliche Telefonberatung und unser kompetenter Außendienst helfen Ihnen gerne weiter. Denn Sie wissen: Bei LeitOn rechnen Sie immer mit bestem Service.

[www.leiton.de](http://www.leiton.de) Info-Hotline +49 (0)30 701 73 49 0



# Audio-DSP-Kurs

## Teil 2: Programmierung des DSPs

Der DSP56374 von Freescale wird vorzugsweise in seiner Assembler-Sprache programmiert. Zunächst wirkt das wie ein Anachronismus, ein unerwünschtes Überbleibsel aus den Frühtagen der Digitalrechner. Beim DSP56374 trifft dies nicht zu. Zum einen lassen sich die zu der Signalarithmetik parallelen Datenbewegungen in einer Hochsprache nicht leichter verwalten als in der Assembler-Sprache. Dann ist ein mit einer Hochsprache erzeugter Code weit weniger ausführungsoptimiert als ein solcher, der in der Assembler-Sprache erzeugt wurde, zumal DSP-Programme in den meisten Fällen von sehr einfacher Struktur sind. Die Schwierigkeiten des Programmierens liegen eher im Bereich von hohen Genauigkeitsanforderungen. Und auch in diesem Fall hilft die Hochsprache nicht weiter.

Von Alexander Potchinkov (D)

Die Assembler-Sprache der DSP563xx-Familie ist vergleichsweise leicht zu erlernen. Sie ist sehr strukturkonsistent und setzt auf einem durchdachten Programmiermodell auf, das auch davon profitiert, dass der Hersteller schon vor der eigenen Markteinführung von vorhandenen DSPs lernen und vor allem die Erfahrungen mit den eigenen Prozessoren nutzen konnte.

Zur Softwareentwicklung benötigt man drei PC-Programme, Assembler, Simulator und Debugger, die entweder als unabhängige Programme oder aber, zusammengefasst in einer IDE (*Integrated Development Environment*) verfügbar sind. Im Prinzip reicht alleine der Assembler. Aber auf Simulator und Debugger zu verzichten, hieße, auf die wichtigsten Werkzeuge zur Fehlersuche in DSP-Programmen zu verzichten. Mit dem Assembler wird ein Sourcecode in den Objectcode übersetzt. Debugger und Simulator dienen der Fehlersuche in den DSP-Programmen und sind sehr ähnliche Werkzeuge mit nahezu gleichem Leistungsumfang, die sich im Wesentlichen darin unterscheiden, ob ein DSP verfügbar sein muss oder nicht.

### Assembler

Ausgangspunkt ist eine Textdatei, die das in der Assembler-Sprache erstellte DSP-Programm enthält. Die Textdatei kann mit einem beliebigen Editor erstellt werden. Ein Befehl kann bis zu sechs durch Leerzeichen abgetrennte Felder enthalten.

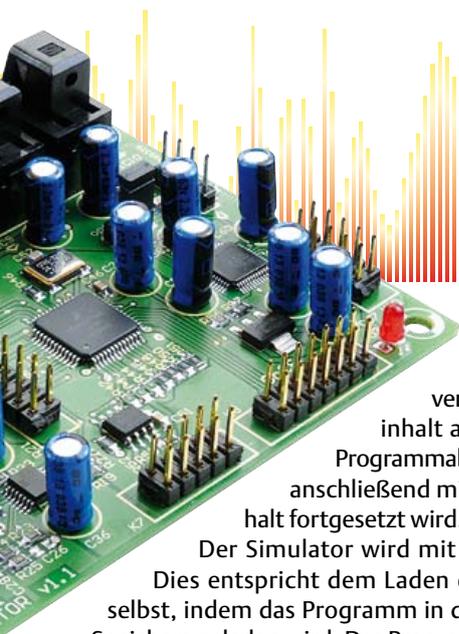
Ein Kommentar wird mit einem Semikolon begonnen. Der Assembler erzeugt eine Datei mit der Kennung .cld, die den Object-Code enthält. Diese Datei kann vom Simulator verwendet oder mit dem Debugger auf den DSP geladen und ausgeführt werden.

### Simulator

Der kostenfreie DSP-Simulator ist ein sehr nützliches Werkzeug zur Softwareentwicklung. Der Simulator ist sozusagen ein nachgebildeter DSP in einer PC-Umgebung bei reduzierter Ausführungsgeschwindigkeit und ohne die DSP-Peripherie zur Verfügung zu stellen. Die Nachbildung schließt nicht nur den Datenpfad des DSPs, sondern auch alle Schnittstellen und den gesamten Speicher bis hin zum Interruptsystem ein. Da der Simulator auch das Pipelining nachbildet, erlaubt er es, die Anzahl der für jedes beliebige Programmstück benötigten DSP-Prozessorakte zu messen, was natürlich für die Echtzeitverarbeitung eine wichtige Information ist, da diese Information nur mit großem Aufwand aus dem DSP-Programm selbst gewonnen werden kann.

Der Simulator kann zum Debuggen genutzt werden, da für jeden Schritt in einem Programm die (virtuellen) Register und Speicherzellen des DSPs gelesen und geschrieben werden können. Man kann an einem fehlerhaften Registerinhalt den Grund für ein fehlerhaftes Programmverhalten sehen oder erkennen, wie sich ein Programm

Marke	Operator	Operanden	X-Bus Transfer	Y-Bus Transfer	Kommentar
Label	mac	x0, y0, a	x: (R0)+, x0	y: (r4)+, y1	; Mac-Operation



verhält, wenn man einen Registerinhalt an einer bestimmten Stelle im Programmablauf ändert und das Programm anschließend mit dem geänderten Registerinhalt fortgesetzt wird.

Der Simulator wird mit dem Object-Code gestartet. Dies entspricht dem Laden eines Programms in den DSP selbst, indem das Programm in das Simulator-Abbild des DSP-Speichers geladen wird. Das Programmabarbeiten kann im Einzelschritt-Modus oder in Blöcken erfolgen. Hierzu bietet der Simulator bedingte und bedingungsfreie Breakpoints, Befehls- und Prozessor-Taktzyklen-Zähler, einen Einzeilen-Assembler und die Möglichkeit, alle Speicher- und Registerinhalte einzeln oder blockweise anzuzeigen und als ASCII-Files zu speichern.

Eine wichtige Anwendung des Simulators ist der File-IO-Betrieb. In diesem Betrieb lässt sich ein DSP-Programm auf seine Signalverarbeitungseigenschaften hin testen, indem es z.B. mit einem in einer Datei abgelegten Testsignal gespeist wird und das verarbeitete Signal, das Programmausgangssignal, in eine Datei geschrieben wird, die mit einem Standardprogramm ausgewertet wird. Einzelne Werte, Abtastwerte von Signalen, werden durch Zeilen in den Dateien voneinander getrennt.

Um die Bedienung des Simulators an einem einfachen Beispiel zu zeigen, wird das DSP-Programm `fileio.asm` benutzt, aus dem mit dem Assembler das Objekt-File `fileio.cld` erzeugt wurde:

```

                org x:$0 ;
simdatain      ds 1 ; memory location for input data
simdataout     ds 1 ; memory location for output data
                org p:$100
start          move x:simdatain,x0 ; x0 <- input data
                move x0,x:simdataout ; x0 -> output data
                jmp start ; repeat process

```

Im DSP-Programm wird der Inhalt der DSP-Speicherzelle `simdatain` in die DSP-Speicherzelle `simdataout` kopiert, wobei das Register `x0` als Zwischenspeicher dient. Jeder Aufruf `move x:simdatain,x0` liest beim Simulatorlauf eine Zeile nach der anderen aus der Eingabedatei und jeder Aufruf `move x0,x:simdataout` schreibt eine Zeile nach der anderen in die Ausgabedatei. Dieser Vorgang wird unendlich oft wiederholt, was mit dem Rücksprungbefehl `jmp start` erreicht wird. Für den Simulatorbetrieb wurde eine Eingabedatei mit dem Namen `infile.dat` erzeugt. Die nachfolgend angeführten Simulatorkommandos dienen dem Einlesen der Eingabedatei und dem Erzeugen einer Ausgabedatei `outfile.dat`, in der am Ende derselbe Inhalt wie in der Eingabedatei erscheinen sollte.

```

(1)  reset s ; reset simulator
(2)  input off ; reset all input files
(3)  output off ; reset all output files
(4)  load fileio.cld ; load the dsp program
(5)  input #1 simdatain infile.dat -rh
      ; input from infile.dat

```

```

(6)  output #1 simdataout outfile.dat -rh -o
      ; output to outfile.dat
(7)  break EOF ; stop when an input file reaches
      end-of-file
(8)  go ; execute program

```

Die Simulatorbefehle bedeuten im Einzelnen:

- In den Zeilen 1 bis 3 wird der Simulator zurückgesetzt. Jetzt noch geöffnete Ein- und Ausgabedateien werden geschlossen.
- In der Zeile 4 wird das DSP-Programm geladen.
- In der Zeile 5 wird die Eingabedatei eingebunden. Die Option `-rh` zeigt an, dass die Eingabedaten als Hexadezimalzahlen gelesen werden sollen. Mit der alternativen Option `-rf` können die für Signale wichtigen Fractionals verwendet werden.
- In der Zeile 6 wird die Ausgabedatei geöffnet. Mit der Option `-o` wird das Überschreiben einer bereits vorhandenen Datei gleichen Namens veranlasst.
- Die Zeile 7 gibt dem Simulator an, dass der Vorgang mit dem Erreichen des Endes der Eingabedatei beendet werden soll.
- Schließlich wird in der Zeile 8 die Ausführung des DSP-Programms im Simulatorlauf gestartet.

Die Simulatorbefehle lassen sich in einer sog. Command-Datei zusammenfassen.

## Debugger

Im Prinzip leistet der Debugger Ähnliches wie der Simulator, nur, es wird nicht simuliert, sondern der DSP selbst führt ein Programm aus. Mit dem Debugger werden Programme in den DSP geladen und lassen sich im Einzelschrittmodus oder in Abhängigkeit von Breakpoints ausführen.

Der Debugger kann den DSP in einen speziellen Modus versetzen, in dem alle Register und Speicherinhalte gelesen oder geschrieben werden können. Da nicht simuliert wird, ist die Ausführungsgeschwindigkeit erheblich höher. Statt der File-Technik mit Test- und Antwortsignal kann man DSP-Programme auch über die Audio-schnittstellen, z.B. mit einem Wave-Editor, testen.

## Benutzung der Software

Das Bild 1 zeigt das Zusammenspiel zwischen den PC-Programmen (grau hinterlegte rechteckförmige Kästchen) und den in diesem Zusammenspiel benötigten und erzeugten Dateien (Kästchen mit den gerundeten Kanten). Ausgangspunkt ist der Sourcecode, der mit dem Assembler in Objectcode übersetzt wird, wobei man den Assembler mit der Option `-l` anweisen kann, das oft nützliche List-File anzulegen.

Wir benötigen einen absoluten Objectcode, der mit der Assembler-Option `-a` erzeugt wird. Der Objectcode kann in drei Programme

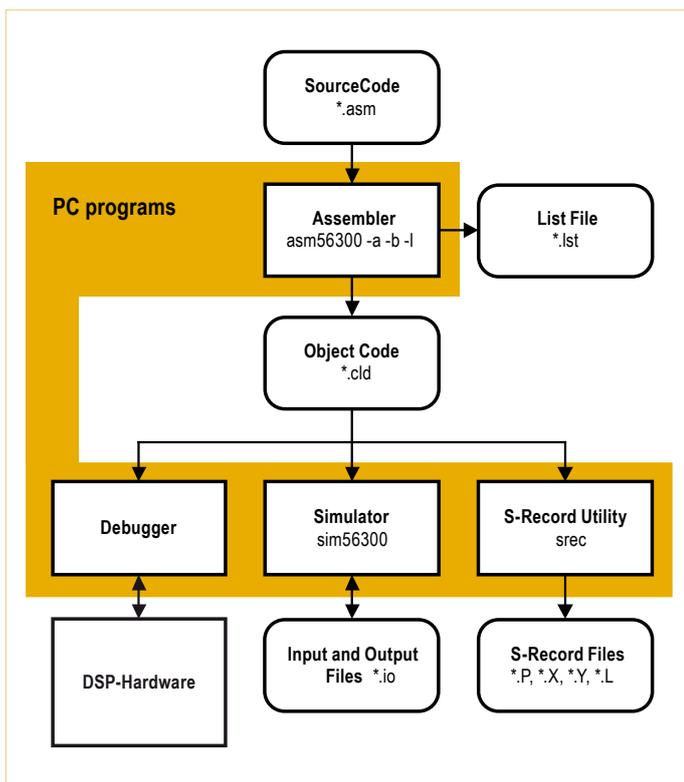


Bild 1. Freescale Softwareentwicklungs-Programme im Zusammenspiel.

eingespeist werden. Es sind dies der Debugger mit angeschlossenem DSP, der Simulator und zuletzt ein noch nicht erwähntes Programm, das Programm srec. Mit diesem Programm kann man sogenannte *S-Records* erzeugen, die zum Brennen eines Boot-PROMS benötigt werden. Diese *S-Records* werden von den üblichen Programmiergeräten verarbeitet.

### Softwareentwicklungsumgebungen

Man benötigt die **Software** und einen **Adapter**, um den auf dem PC laufenden Debugger mit dem DSP zu verbinden.

#### Freescale Suite 56

##### Software

Die Software [1] umfasst Assembler, Linker, Simulator und Debugger und kann kostenfrei über die Freescale-Web-Präsenz heruntergeladen werden. Simulator und Debugger liegen sowohl als Kommandozeilen-Programme als auch mit graphischer Benutzeroberfläche vor (GUI56300 für den Simulator und GDS56300 für den Debugger). Die Software ist nach vergleichsweise geringer Einarbeitungszeit nutzbar. Das **Bild 2** zeigt die Suite-56-Softwarekomponenten. Suite 56 ist wegen des Linkers und der Libraries im besonderen Maße für die Verwendung in Arbeitsgruppen geeignet.

##### Adapter

Die DSP-Anbindung erfolgt über den PC-Parallelport mit einem Adapter, der entweder von Freescale [2] oder, baugleich, von Macraigor, Wiggler [3], erworben werden kann. Da es sich hierbei aber nur um Bustreiber und einige wenige weitere Bauelemente handelt, ist auch ein Selbstbau möglich, wofür man im Internet z.B. bei Gerrit Buhe, UniDSP56 [4], ausgearbeitete und dokumentierte Vorschläge findet. Falls am PC kein Parallelport mehr vorhan-

den ist, lässt sich ein solcher mit einer Standard-PCI-Einsteckkarte einrichten.

#### Freescale Symphony Studio

##### Software

Freescale stellt ebenfalls kostenfrei die Software-Plugins Assembler, Linker, C-Compiler, Simulator und Debugger für eine Eclipse-Umgebung zur Verfügung [5]. Allerdings ist die Einrichtung der Plugins und somit die Erstellung einer Softwareentwicklungsumgebung nicht gut dokumentiert und wird vor allem den mit Eclipse nicht vertrauten Anwender vor Probleme stellen. Auf der anderen Seite ist es aber auch sinnvoll, sich mit moderner Softwareentwicklungsumgebung vertraut zu machen, die für viele Prozessoren verwendet werden kann.

##### Adapter

Die DSP-Anbindung erfolgt wie bei Suite56 über den PC-Parallelport mit einem Adapter von Freescale [2] oder, baugleich, von Macraigor, Wiggler [3]. Zusätzlich kann man auch USB-Adapter erwerben, z.B. von Macraigor (usbWiggler [6]) oder Domain Technologies [7].

#### Domain Technologies BoxView

##### Software

Der Hersteller Domain Technologies bietet eine Debug-Software unter der Bezeichnung *BoxView* [8] an. Zum Assemblieren und Simulieren lassen sich dann die entsprechenden kostenfreien Freescale-Produkte asm56300 und sim56300 nutzen.

##### Adapter

Domain Technologies bietet einen USB-Adapter an [7].

#### Domain Technologies EVM-Bundle-Debugger

Zur Nutzung mit den EVM-Boards, die von Freescale für DSP563xx Signalprozessoren angeboten werden, ist ein *BoxView-Debugger* vorgesehen, der kostenfrei unter [9], Bv30xEvm.exe, heruntergeladen werden kann, falls er nicht zusammen mit einem EVM-Board erworben wurde. Dieser Debugger wird vom Autor bevorzugt benutzt und ist besonders anwenderfreundlich. Das **Bild 3** zeigt beispielhaft die Bildschirm-Kopie einer Debugger-Anwendung. Es handelt sich um den *BoxView-Debugger*. Neben dem Fenster mit dem Source-Code erkennt man weitere Fenster mit Register- und Speicherinhalten. Man kann Speicherinhalte auch graphisch anzeigen, was in vielen Fällen sehr aufschlussreich sein kann.

##### Adapter

Man kann sich ein EVM-Board besorgen, wie man es manchmal für wenig Geld gebraucht erhält, und es als Adapter für ein eigenes DSP-Board nutzen.

In Ergänzung zu den bereits auf dem Markt verfügbaren Adaptern und zum Parallel-Port-Selbstbauadapter arbeitet der Autor in Koordination mit Elektor an zwei USB-Adaptoren, die für die Elektorleser preiswert zu beschaffen sein werden, damit dem schnellen Einstieg in die digitale Audiosignalverarbeitung unter Nutzung unserer DSP-Kurs-Hardware nichts im Wege steht.

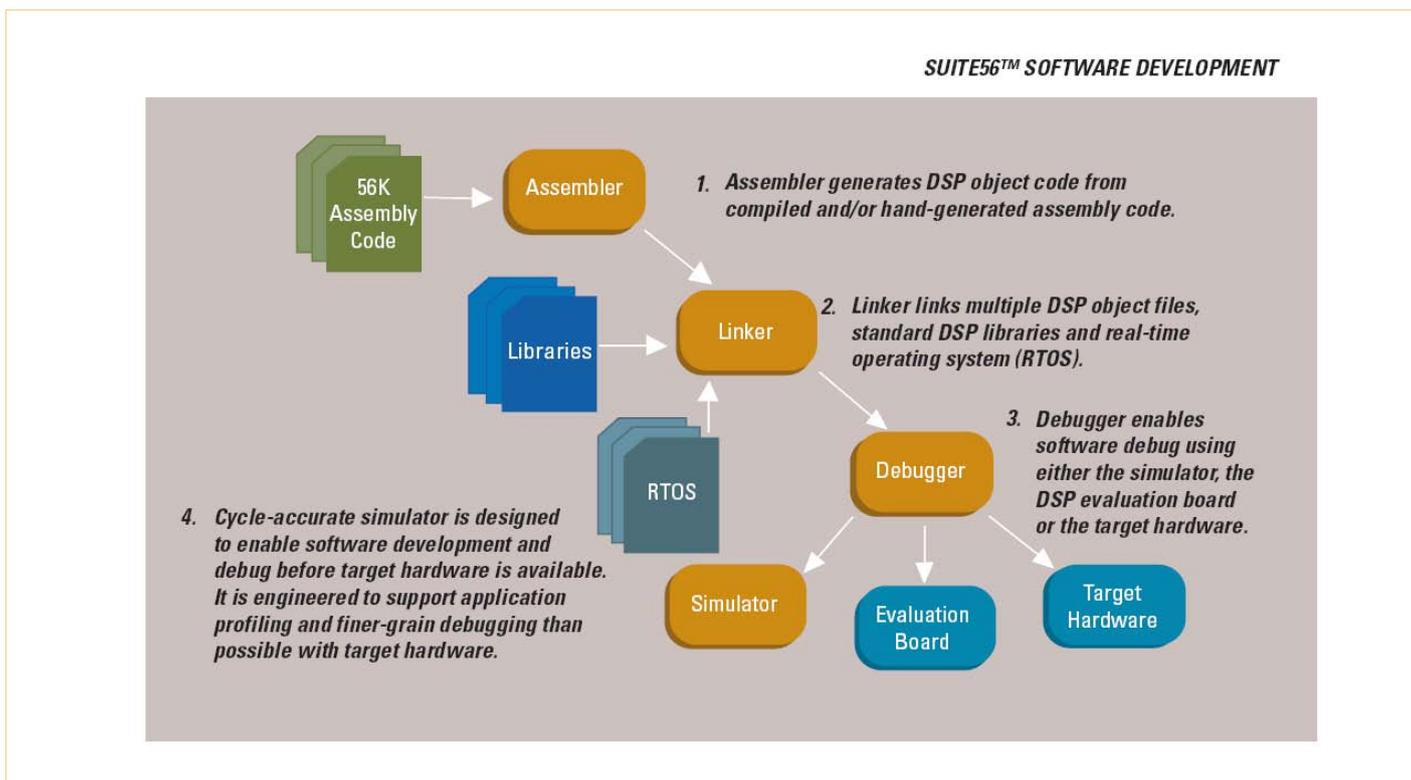


Bild 2. Die Software in Suite 56 (Bild: Freescale).

### DSP-Programme und die Audioloop

Die DSP-Programme dieser Artikelserie folgen derselben Grobstruktur mit einem

- Rahmenprogramm und einer
- Audioloop.

Das Rahmenprogramm ist für alle Anwendungen weitestgehend gleich und enthält die Einrichtung sowie die Bedienung der DSP-Peripherieschnittstellen. Hinzu kommen die Initialisierung des SRC und der Anwendung wie z.B. das Löschen von Zustandsspeichern, damit stets derselbe Programmstart gewährleistet ist. Die Audioloop enthält die Audio-Synchronisation und die digitale Signalverarbeitung.

Im Rahmenprogramm werden je zwei Eingangs- und Ausgangskanäle eingerichtet, die mit den Audiodatenanschlüssen RX0 als Zweikanalempfänger und TX0 als Zweikanalsender zur Verfügung gestellt werden. Im DSP-Programm werden zwei Pufferspeicher als Audiopuffer benötigt, die wir im X-RAM einrichten und mit zwei Zeigern auf die Pufferbasisadressen versehen.

```
TxBuffBase EQU $000000 ; BaseAddress of TX-Buffer (X)
RxBuffBase EQU $000010 ; BaseAddress of RX-Buffer (X)
TXPTR EQU $000020 ; Address of Tx-BufferPointer (X)
RXPTR EQU $000021 ; Address of Rx-BufferPointer (X)
```

Im Weiteren benötigen wir ein Flag, das Links/Rechts-Flag, mit dem die Audiodatensynchronisation im Hauptprogramm durch Pollen des Flags durchgeführt wird.

```
LRFlag EQU $000022 ; Address of Left/Right-Flag
RightRx EQU 0 ; Bit Position in Left/Right-Flag
```

Bei der zweikanaligen Audiodatenübertragung wird zuerst der linke und anschließend der rechte Kanal übertragen, die der DSP mit 0 und 1 zählt. Hierzu werden die DSP-Audioschnittstellen im Netzwerk-Modus betrieben, der für zwei Kanäle zu konfigurieren ist. Das Interrupt-System der Audioschnittstellen stellt einen Interrupt für das erfolgte Einlesen des *Last Slots* im Netzwerkmodus zur Verfügung. Die Schnittstelle ist in der Lage, bis zu 32 Audiokanäle für jeden Audiodatenanschluss zu verwalten. Wir benötigen davon aber nur zwei Kanäle. Dann bezieht sich Last Slot auf den rechten oder zweiten Kanal. Das Flag, das mit dem Wert 1 den Empfang des rechten Kanals anzeigt, wird in einer ISR gesetzt, die vom Last-Slot-Interrupt ausgelöst wird.

In der Initialisierungsphase tragen wir Nullen in die beiden Audiopufferspeicher ein und setzen das Synchronisationsflag zurück.

```
org x:TxBuffBase
dc 0
dc 0
org x:RxBuffBase
dc 0
dc 0
org x:LRFlag
dc 0
```

Das Programm verwendet sechs Long-Interrupt-ISR zur Bedienung der Audioschnittstellen. Wir tragen `jsr`-Befehle, gefolgt von den ISR-Basisadressen, in die Interrupt-Vektortabelle ein.

Das DSP-Programm kann nun beginnen. Zuerst muss die Audioschnittstelle konfiguriert werden. Die dazu nötigen Einstellungen sind individuell zu wählen, da sie von der Hardware abhängen. Die wichtigste Festlegung ist, ob der DSP als Audio-Master oder als Audio-Slave arbeiten soll. Wenn der DSP Audio-Master ist, müssen

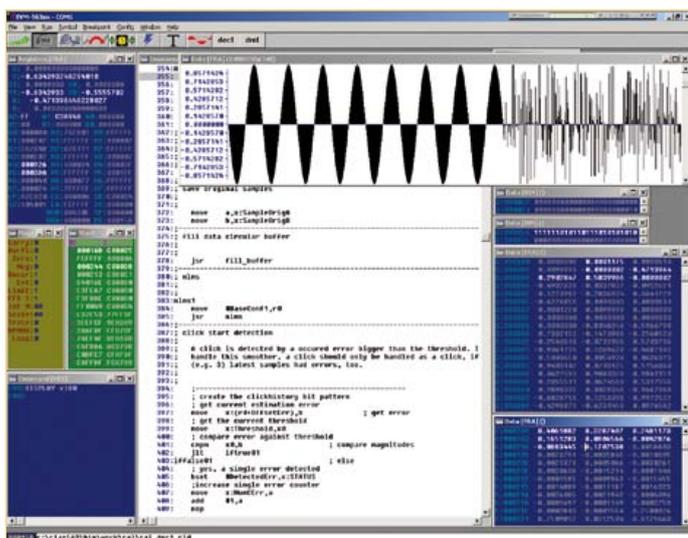


Bild 3. Domain Technologies BoxView-Debugger.

```

move  x:RxBuffBase,a      ; Left Channel ->a
move  x:RxBuffBase+1,b   ; Right Channel ->b

; insert signal processing here

move  a,x:TxBuffBase     ; a-> Left Channel
move  b,x:TxBuffBase+1  ; b-> Right Channel
jmp   AudioLoop
    
```

schreiben zunächst den 2-Worte-Empfangsbufferinhalt in die beiden Akkumulatorregister a und b. Nun können die Audiodaten mit einem DSP-Programm verarbeitet werden. Wenn die Verarbeitung abgeschlossen ist, werden die zwei gewonnenen Signalwerte in den Sendebuffer geschrieben. Im Beispielsprogramm werden hierzu die beiden Akkumulatorregister a und b gelesen. Wenn keine Signalverarbeitung durchgeführt wird, reicht der DSP die empfangenen Daten um einen Abtasttakt verzögert durch. Der letzte Befehl ist der Rücksprung zum Pollen des Synchronisationsflags. So viel zur Einführung in die Programmierung des DSPs. Im dritten Teil werden wir die Hardware beschreiben, die speziell für diesen Kurs entwickelt wurde.

(110002)

die Audiotakte vom DSP-Prozessor abgeleitet werden. Sinnvollerweise wird dann der Taktoszillator eine ganzzahlige Teilerfrequenz des Audiomastertakts verwenden. Wenn z.B. die Audiomastertaktfrequenz 24,576 MHz beträgt, kann ein Standard-Quarzoszillator mit 6,144 MHz verwendet werden. Die DSP-Takt-PLL ist dann auf einen Multiplikationsfaktor von 24 einzustellen und der DSP wird mit einer Prozessortaktfrequenz von 147,456 MHz betrieben. In der Audioschnittstelle sind dann mit 1/6 der Prozessortaktfrequenz der Mastertakt, mit 1/48 der Prozessortaktfrequenz der Bit-Takt zur Bedienung der Schnittstellenschieberegister (mit einer Frequenz von 3,072 MHz) und mit 1/3072 der Prozessortaktfrequenz die Abtastfrequenz von 48 kHz, mit der der Links-Rechts-Takt läuft, zu gewinnen. Wenn der DSP als Audio-Slave arbeiten soll, kann z.B. ein ADC oder ein Digital-Audio-Empfänger als Master die benötigten Takte erzeugen. Zur weiteren Konfiguration der Audioschnittstellen werden die Angaben zum Netzwerkmodus benötigt (welche Empfänger und Sender, welches Datenformat, welche Interrupts werden verwendet u.s.w.). Die Audioloop beginnt mit dem Pollen des Synchronisationsflags.

```

AudioLoop
    jclr #RightRx,X:LRFlag,* ; right word received?
    bclr #RightRx,X:LRFlag ; reset synchronisation flag
    
```

Der `jclr`-Befehl leistet das Pollen. Es wird mit immer wiederkehrender Abfrage solange gewartet, bis das Flag einen Wert ungleich Null hat. Mit dem Stern als Sprungziel wird die wiederholende Abfrage des Flags veranlasst. Der folgende `bclr`-Befehl setzt das Flag zurück auf Null, damit es durch die entsprechende ISR wieder gesetzt werden kann. Die Befehle

**Weblinks:**

- [1] Freescale Suite 56:Software: [www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=CW-SUITE56&fsrch=1&sr=4](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CW-SUITE56&fsrch=1&sr=4)
- [2] Freescale Adapter: [www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=DSPCOMMPARALLEL](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=DSPCOMMPARALLEL)
- [3] Macraigor Wiggler: [www.macraigor.com/wiggler.htm](http://www.macraigor.com/wiggler.htm)
- [4] Selbstbau-Adapter von Gerrit Buhe, UniDSP56: [www.unidsp56.de/downloads.html](http://www.unidsp56.de/downloads.html)
- [5] Freescale Symphony Studio Software: [www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=SYMPH\\_STUDIO&fsrch=1&sr=16](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=SYMPH_STUDIO&fsrch=1&sr=16)
- [6] Macraigor usbWiggler: [www.macraigor.com/usbWiggler.htm](http://www.macraigor.com/usbWiggler.htm)
- [7] Domain Technologies Adapter: [www.domaintec.com/usbemu.html](http://www.domaintec.com/usbemu.html)
- [8] Domain Technologies Debug-Software BoxView: [www.domaintec.com/BoxView.html](http://www.domaintec.com/BoxView.html)
- [9] Domain Technologies EVM-Bundle-Debugger: [www.domaintec.com/FSsoftware.html](http://www.domaintec.com/FSsoftware.html)

**Der Autor**

Alexander Potchinkov ist Inhaber des Lehrstuhls für digitale Signalverarbeitung an der TU Kaiserslautern und führt ein Ingenieurbüro für Audiosignalverarbeitung. Sein Interesse gilt nicht nur den DSPs

und ihren Algorithmen, sondern auch den Röhrenverstärkern und der SPICE-Simulation.

# Abonnieren Sie jetzt die führende US-amerikanische Fachzeitschrift für Mikrocontroller-Anwendungen und Embedded Systems!



Wählen Sie unter  
[www.elektor.de/cc-abo](http://www.elektor.de/cc-abo)  
Ihr gewünschtes  
Abonnement aus!

**12 Original-Ausgaben jährlich für nur**

Digital: 38 US-Dollar

Print: 63 US-Dollar

Digital + Print: 90 US-Dollar

**CIRCUIT  
CELLAR®**  
THE MAGAZINE FOR COMPUTER APPLICATIONS



# Inside USB

Für die Kommunikation zwischen PC und angeschlossenen Geräten ist der Universal Serial Bus der unumstrittene Standard. Dessen größte Errungenschaft ist die Nutzerfreundlichkeit, im einfachsten Fall muss nur das Kabel eingesteckt werden. Doch mit diesem Wissen geben sich Elektor-Leser natürlich nicht zufrieden! Wir stellen die Grundzüge des Protokolls vor.

Von Guy Weiler (Lux)

Das USB-Protokoll für die Kommunikation zwischen PC (Host) und Gerät (Device) besteht aus mehreren Schichten. Die umfassendste Übertragungseinheit (Anfrage, Vorbereitung und Ausführung der Kommunikation) wird als USB-Transfer bezeichnet. Bis auf eine Ausnahme (den sogenannten Control-Transfer) laufen während eines Transfers Daten nur in eine Richtung.

Ein Transfer besteht aus einer oder mehreren Transaktionen, deren Merkmal es ist, dass sie in einem Zug durchgeführt werden müssen. Jede Transaktion ist wiederum aus einem Token-Paket (Header), einem optionalen Datenpaket und einem Status-Paket (Erfolgskontrolle) zusammengesetzt. Die Pakete selbst sind weiter untergliedert, je nach Art des Pakets gehören eine Paket-ID, die Empfänger-Adresse und eine Fehlerkontrolle dazu (siehe Bild 1). Doch nun der Reihe nach!

## Transfertypen

Der Universal Serial Bus ist – wie der Name schon sagt – universell verwendbar, daher existieren vier unterschiedliche Transfertypen. Eine Sonderstellung nimmt der Control-Transfer ein, der von jedem Gerät unterstützt werden muss (siehe unten). Die drei anderen nehmen auf unterschiedliche Anforderungen der Kommunikation Rücksicht:

- **Bulk-Transfer:** zur Übertragung großer Datenmengen ohne garantierte Geschwindigkeit (Beispiel Laufwerk).
- **Interrupt-Transfer:** mit garantierter Bandbreite für ein geringes Datenvolumen (Beispiel Tastatur).
- **Isochron-Transfer:** mit garantierter Bandbreite, aber ohne Fehlerkorrektur (Beispiel: Audio-Streaming).

Diese drei Transfers sind bezüglich der übertragenen Daten immer

unidirektional. Sie sind wiederum aus kleinsten, zeitlich nicht trennbaren Einheiten aufgebaut, den Transaktionen. Es gibt In-Transaktionen, die Daten vom Gerät zum PC senden, und Out-Transaktionen, die Daten vom PC zum Gerät übermitteln (In/Out aus der Sicht des PCs). Die Transaktionen bestehen wiederum aus drei Paketen. Das Token-Paket gibt an, um welchen Transaktionstyp es sich handelt (In/Out). Das Data-Paket enthält die Daten, während das Handshake-Paket ein Feedback des Empfängers zurückgibt. Folgende Möglichkeiten gibt es hierbei: „Erfolg“ (ACK) oder „ich bin beschäftigt, bitte warten“ (NAK).

## Control-Transfer

So weit so gut, doch ist es gerade das Wesen des USB-Busses, dass sich eine Kommunikation zwischen PC und Gerät geradezu vollautomatisch aufbaut. Der Vorgang, bei dem der Host dem Gerät eine Adresse zuordnet, alle Informationen über das Gerät erfragt, den richtigen Treiber lädt und dann eine Konfiguration auswählt, wird Enumeration genannt. Hinzu kommt, dass jedes Gerät nicht nur über eine Adresse anzusprechen ist, sondern verschiedene „Endpunkte“ mitbringt, die Daten entgegennehmen oder ausgeben können (siehe Kasten). Auch die Zahl und Größe dieser Endpunkte muss der Host/PC erfragen.

Die gesamte Enumeration des Gerätes erfolgt mittels einer speziellen, bidirektionalen Art von Transfers, den sogenannten Control-Transfers (immer über den Endpunkt 0 des Gerätes). Über Control-Transfers lassen sich die Geräte auch steuern. Der Control-Transfer ist wie auch die anderen Transfertypen aus Transaktionen zusammengesetzt, nämlich:

1. Einer Setup-Transaktion.
2. Optional einer oder mehreren Daten-Transaktionen.
3. Einer Status-Transaktion.

Es gibt schreibende und lesende Control-Transfers. Ob der Host Daten in das Gerät schreiben oder von dort auslesen will, teilt er dem Gerät innerhalb der Setup-Transaktion mit.

# Wie das Protokoll funktioniert

## Die Transaktionen

Wie alle Transaktionen besteht auch die Setup-Transaktion aus einem Token-, Data- und Handshake-Paket. Der PC sendet das Token und das Data-Paket, das Gerät antwortet im Erfolgsfall mit dem Handshake-Paket „ACK“ (Bild 2).

Alle Informationen zur Anfrage befinden sich im 8 Byte großen Setup-Data-Paket. Das erste Byte *bmRequestType* bestimmt unter anderem die Richtung der Anfrage (Bit7=0 heißt PC zum Gerät, Bit7=1 heißt umgekehrt) und ob es sich um eine Standard- oder eine herstellerspezifische Anfrage handelt.

Weitere Bytes entfallen auf die Felder *bRequest* (1), *wValue* (2), *wIndex* (2) und *wLength* (2).

Zuerst schauen wir uns den schreibenden Transfer an, siehe hierzu Bild 2. Nach dem Empfang der Daten innerhalb der Daten-Transaktion kann das Gerät den Empfang bestätigen (ACK), den PC auffordern zu warten (NAK) oder Abbrechen (STALL). Tritt beim Token- oder Data-Paket ein Fehler auf, so wird das Paket ignoriert. Wie schon erwähnt, können auch mehrere Datentransaktionen aufeinanderfolgen (in der Grafik ist nur eine eingezeichnet).

In der Status-Transaktion, die den Transfer abschließt, bestätigt das Gerät, dass alles gut gegangen ist. Nun sind die Rollen vertauscht, das Gerät sendet (Status-)Daten, der Host antwortet mit einem Handshake-Paket. War der Transfer erfolgreich, so sendet das Gerät ein „ZeroLength“-Paket an den PC. Der PC antwortet mit ACK. Trat ein Fehler beim Endpunkt auf, so sendet das Gerät STALL. Ist es noch beschäftigt, so sendet es ein NAK.

Beim lesenden Control-Transfer (Bild 3) kann das Gerät innerhalb der Daten-Transaktion auf drei Weisen auf das In-Token des Hosts reagieren. Das Gerät kann die Daten bereitstellen; ist das Gerät jedoch noch beschäftigt, so sendet es ein NAK. Trat ein Fehler beim Endpunkt auf, so sendet es ein STALL. Tritt sonst ein Fehler auf, so wird das Token ignoriert und nichts zurückgeschickt.

Während der Status-Transaktion meldet der PC mit einem „ZeroLength“-Paket, wenn er die Daten erfolgreich erhalten hat, was wiederum vom Gerät mit einem ACK-Handshake-Paket bestätigt werden muss.

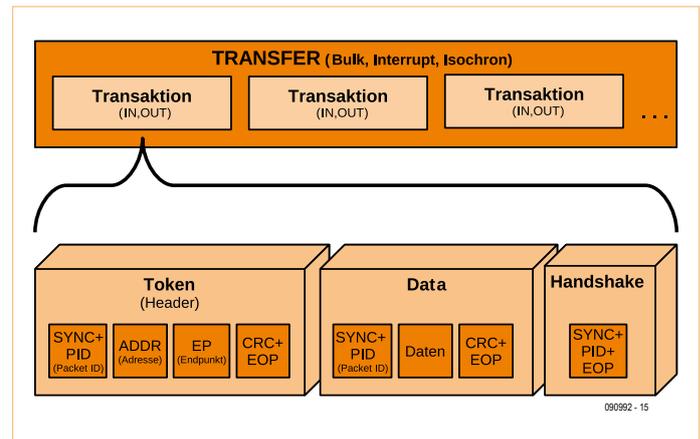


Bild 1. USB-Transfers bestehen aus untrennbaren Transaktionen, die wiederum aus einem Token-, einem Data- und einem Handshake-Paket zusammengesetzt sind (die speziellen Control-Transfers sind hier nicht wiedergegeben).

## Deskriptoren

Während der Enumeration werden vom PC aus mittels Standard-Anfragen (*standard requests*) mehrere sogenannte Deskriptoren (Beschreibungen) erfragt. Diese Geräte-Beschreibungen sind hierarchisch organisiert [1]. Jedes Gerät muss mindestens vier Deskriptoren liefern:

- **Geräte-Deskriptor (device descriptor):** Gibt es pro Gerät nur einmal.
- **Konfigurations-Deskriptor(en) (configuration descriptor):** Ein Gerät kann mehrere Konfigurationen besitzen, zum Beispiel eine Konfiguration für die Stromversorgung per Bus besitzen und eine zweite für eine eigenständige Versorgung. Es ist immer nur eine Konfiguration aktiv.

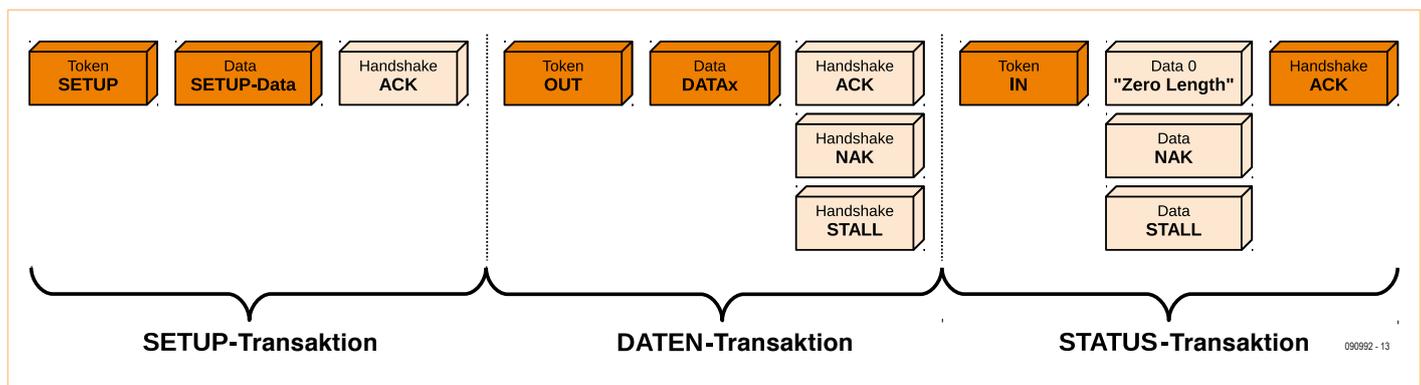


Bild 2. Jeder Control-Transfer (hier schreibend) wird von einer Setup-Transaktion eingeleitet. Daraufhin folgen optional eine oder mehrere Daten-Transaktion(en) und eine Status-Transaktion.

Der Geräte-Deskriptor			
Feldbezeichnung	Byte	Wert in AVR-Library [3]	Beschreibung
bLength	1	18 (0x12)	Größe des Deskriptors in Byte
bDescriptorType	1	1	Geräte-Deskriptor = 1 (Konstante)
bcdUSB	2	0x0110	USB_Spec1_1
bDeviceClass	1	0xFF	Klassencode (hier anbieterspezifisch = 0xFF, siehe [6])
bDeviceSubClass	1	0xFF	Unterklassencode (hier anbietersp. = 0xFF)
bDeviceProtocoll	1	0xFF	Protokollcode (hier anbieterspezifisch = 0xFF)
bMaxPacketSize	1	8	max. Paketgröße Endpunkt 0 (EP0_FS)
idVendor	2	0x03eb	Atmel Code durch usb.org vergeben
idProduct	2	0x0001	Produkt-ID beliebig
bcdDevice	2	0x0001	Release Nummer Gerät
iManufacturer	1	1	Index für String-Deskriptor Hersteller
iProduct	1	2	Index für String-Deskriptor Produkt
iSerialNumber	1	3	Index für String-Deskriptor Seriennummer
bNumConfigurations	1	1	Anzahl möglicher Konfigurationen

Der Konfigurations-Deskriptor			
Feldbezeichnung	Byte	Wert in AVR-Library [3]	Beschreibung
bLength	1	9	Größe des Deskriptors in Byte
bDescriptorType	1	2	Konfigurations-Deskriptor = 2 (Konstante)
wTotalLength	2	39 (0x27)	Länge des Konfigurations-Deskriptors und aller untergeordneter Schnittstellen- und Endpunkt-Deskriptoren
bNumInterfaces	1	1	Anzahl der Schnittstellen
bConfigurationValue	1	1	Nummer, um diese Konfiguration auszuwählen (darf nicht Null sein, sonst geht Gerät in den nicht-konfigurierten Zustand)
iConfiguration	1	0	Index für String-Deskriptor dieser Konfiguration (0 = kein Text)
bmAttributes	1	0x80	Bit7 = 1 Versorgung durch Bus, Bit6 = 1 Selbstversorgung, Bit5 = 1 Remote Wakeup
bMaxPower	1	50	Max. Strombezug vom Bus in 2-mA-Schritten

Der Schnittstellen-Deskriptor			
Feldbezeichnung	Byte	Wert in AVR-Library [3]	Beschreibung
bLength	1	9	Größe des Deskriptors in Byte
bDescriptorType	1	4	Schnittstellen-Deskriptor = 4 (Konstante)
bInterfaceNumber	1	0	Anzahl der Schnittstellen
bAlternateSetting	1	0	Nummer um alternative Einstellungen zu wählen
bNumEndpoints	1	3	Anzahl der Endpunkte außer Endpunkt 0
bInterfaceClass	1	0xFF	Klassencode (hier anbieterspezifisch = 0xFF)
bInterfaceSubClass	1	0xFF	Unterklassencode (hier anbietersp. = 0xFF)
bInterfaceProtocol	1	0xFF	Protokollcode (hier anbieterspezifisch = 0xFF)
iInterface	1	0	Index für String-Deskriptor dieser Schnittstelle (0 = kein Text)

Der Endpunkt-Deskriptor			
Feldbezeichnung	Byte	Wert in AVR-Library [3]	Beschreibung
bLength	1	7	Größe des Deskriptors in Byte
bDescriptorType	1	5	Endpunkt-Deskriptor = 5 (Konstante)
bEndpointAddress	1	0x81	Bit7 = 1 (IN), Bit0..3 Endpunkt-Nummer (andere 0)
bmAttributes	1	2	Transfertyp = Bulk (Contr. = 0, Iso. = 1, Int. = 3)
wMaxPacketSize	2	8	FIFO Größe des Endpunkts in Byte
bInterval	1	0	Polling Interval = 0 (ignoriert für Bulk und Control), 1 für Iso, 1...255 für Interrupt

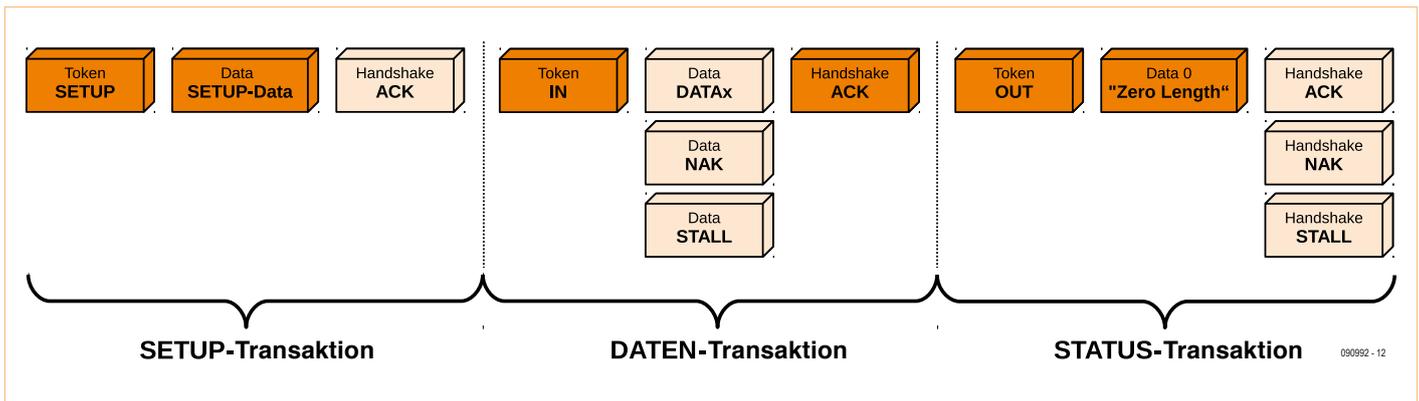


Bild 3. Lesender Control-Transfer.

- **Schnittstellen-Deskriptor(en) (interface descriptor):** Der Schnittstellen-Deskriptor bündelt mehrere Endpunkte zu einer Funktionsgruppe. Es können mehrere Schnittstellen gleichzeitig aktiv sein (Beispiel: Fax-Schnittstelle, Drucker-Schnittstelle und Scanner-Schnittstelle bei Multifunktionsdruckern).
- **Endpunkt-Deskriptor(en) (endpoint descriptor):** Die Endpunkt-Deskriptoren beschreiben die Endpunkte (außer Endpunkt 0). Wichtig ist, dass bei der Endpunktadresse auch die Richtung mittels Bit 7 angegeben werden muss.

Die optionalen String-Deskriptoren sind nicht unbedingt nötig, liefern aber zusätzliche lesbare Informationen. Strings sind in Unicode (16 bit) codiert. Im String-Deskriptor mit dem Index 0 werden die unterstützten Sprachen festgelegt.

### Die Enumeration

Es sollen hier nur die allernötigsten Schritte einer Enumeration vereinfacht beschrieben werden. Wir gehen dabei von nur einer Konfiguration und einem Interface aus.

Nachdem das Gerät an den Bus angeschlossen wurde, ermittelt der PC anhand der Spannungen der beiden Signalleitungen, ob es sich um ein Low- oder Full-Speed-Gerät handelt. Der PC resetet dann das Gerät (beide Datenleitungen auf Low) und findet dabei heraus, ob es sich um ein High-Speed-Gerät handelt. Nach dem Reset des Gerätes ist dieses bereit, über Endpunkt 0 auf der Adresse 0 angesprochen zu werden.

Die (Standard-)Anfragen werden dann mittels schreibenden und lesenden Control-Transfers durchgeführt. Zuerst werden 8 Byte des 64 Byte großen Geräte-Deskriptors erfragt, um die maximale Paketgröße von Endpunkt 0 zu ermitteln (diese kann 8, 16, 32 und 64 Byte betragen). Danach führt der PC einen Reset des Gerätes durch und sendet anschließend eine Geräte-Adresse.

Mit dem dritten Transfer werden 18 Byte des Geräte-Deskriptors erfragt; dann werden mit einem vierten Transfer 9 Byte des Konfigurations-Deskriptors ermittelt. Die Daten zeigen die Gesamtlänge des Konfigurations-, Interface- und aller Endpunkt-Deskriptoren. Ein fünfter Transfer erfragt all diese Deskriptoren in einer Aktion. Der PC lädt jetzt anhand von Vendor- und Product-ID den entsprechenden Gerätetreiber.

Ein letzter Transfer der Enumeration bewirkt das Initialisieren und Aktivieren der User-Endpunkte (Endpunkt 1-15).

Von einer Firmware nicht unterstützte Anfragen des PCs werden vom Gerät einfach mit STALL abgewiesen.

### Weitere Infos

Der Autor dieses Artikels [2] hat zusammen mit Jean-Claude Feltes (ebenfalls Elektor-Autor) eine USB-Minimal-Bibliothek für AVR-Controller entwickelt, die unter [3] heruntergeladen werden kann (Dokumentation in deutscher Sprache). Auf PC-Seite wird die USB-Bibliothek libusb genutzt [4].

Weitere Infos zur USB-Schnittstelle finden sich auch unter [5].

(090992)JN

### Weblinks

- [1] [www.beyondlogic.org/usbnutshell/usb5.shtml#DeviceDescriptors](http://www.beyondlogic.org/usbnutshell/usb5.shtml#DeviceDescriptors)
- [2] [weigu@weigu.lu](mailto:weigu@weigu.lu)
- [3] [www.weigu.lu/b/usb/key/vendor/index.html](http://www.weigu.lu/b/usb/key/vendor/index.html)
- [4] <http://sourceforge.net/projects/libusb/>
- [5] [www.elektor.de/090768](http://www.elektor.de/090768)
- [6] [http://de.wikipedia.org/wiki/Universal\\_Serial\\_Bus](http://de.wikipedia.org/wiki/Universal_Serial_Bus)

## Endpunkte

Die Daten werden zwischen dem PC und einem Geräte-Endpunkt (*endpoint* EP) ausgetauscht. Ein Endpunkt ist ein Datenspeicher (FIFO, Puffer, Speicherbank) im Gerät, der meist nur aus 8 bis 256 Bytes besteht. Jedes Gerät hat mehrere Endpunkte, die über die Endpunktadresse (0-15) angesprochen werden. Diese Adresse enthält in Bit 7 die Richtung der Datenkommunikation (Bit 7 = 0: OUT-Endpunkt; Bit 7 = 1: IN-Endpunkt). Der Control-Endpunkt 0 ist in jedem Gerät vorhanden und wird zum Beispiel für die Enumeration benötigt. Er ist als einziger Endpunkt bidirektional. Bei USB 1.1 besitzt er einen 8 Byte großen FIFO-Speicher. Die Anzahl und mögliche Größe der anderen Endpunkte variieren. Jede Transaktion ist an eine Geräte- und eine Endpunktadresse gebunden.

# Wie man Ideen ins Web XPortiert

Mehr und mehr Mikrocontroller-Systeme besitzen einen Ethernet-Anschluss. Lantronix hat den umgekehrten Weg beschritten und einen Mikrocontroller in eine RJ-Buchse integriert: der XPort Pro war geboren. Schauen wir, was in ihm steckt!

Von Kévin Petit (F), kpet@free.fr



Kern des XPort Pro ist ein Coldfire-Prozessor MCF5483 von Freescale (166 MHz, Ethernet, hardwareunterstützte Verschlüsselung, 8 MB RAM, 16 MB Flash); das Ganze steckt in einem „RJ-Gehäuse“ mit den Abmessungen 3,3 x 1,6 x 1,4 cm. Es gibt fünf Ein-/Ausgänge, von denen zwei für den seriellen Port vorgesehen sind (TXD, RXD), die restlichen drei können ebenfalls hierfür eingesetzt werden (RTS, CTS, DTR) oder stehen als GPIOs (CP1, CP2 und CP3) zur Verfügung. Ein Linux- oder besser *uClinux*-Kernel haucht dem Ganzen Leben ein.

Die Möglichkeiten des XPort Pro lassen sich ohne eigenen Platinenentwurf ausloten, indem man entweder das Demo-Board von Lantronix einsetzt (**Bild 1**) oder die XPort(Pro)-Erweiterung aus unserem Embedded Special des Dezemberhefts 2010 [9].

Die Installation und Konfiguration eines Software Development Kits (SDK) können bei Ungeübten unter Umständen zum Haareraufen führen. Bei einem Linux-SDK wie diesem handelt es sich meist um eine Kombination von Source-Code (Linux-Kernel, C-Bibliothek, Applikationen usw.) begleitet von „Konstruktions-Regeln“ (*makefile*) und diversen Tools (compilation chain, Bilderzeugung usw.). Aber: Einem SDK darf man niemals blind vertrauen! Sehr oft können Sie eine Vielzahl von Dingen einstellen – vielleicht sogar zu viele, falls Sie noch Einsteiger sind. Doch seien Sie beruhigt, hier wartet eine Kurzanleitung auf Sie.

## Installation des SDK

Das von Lantronix ausgelieferte SDK ist ausschließlich unter Linux lauffähig.

Eine Lösung auf der Basis einer virtuellen Maschine ist für Windows-Anwender verfügbar (oder jene, die auf etwas Leistung verzichten wollen, um sich nicht mit der Installation herumschlagen zu müssen). Die Downloads finden Sie unter [2]; je nachdem, was Sie (herunterladen) wollen, kann es notwendig sein, sich anzumelden.

Falls Sie sich gegen den Einsatz der virtuellen Maschine entschlossen haben, müssen Sie das SDK selbst installieren. Dazu muss man entweder das ISO-Image des SDKs auf eine CD brennen oder man muss es (nach Linux-Terminologie) „mounten“. Der Autor hat sich für den zweiten Weg entschieden. Zuerst müssen Sie ein Verzeichnis erstellen, in welchem Sie das ISO-Image mounten (um dies zu tun, müssen Sie *root* sein), z.B. mit `mkdir /mnt/iso`. Nun können Sie das Image mounten mittels des Kommandos `mount -o loop /path/to/iso/image /mnt/iso`. Erstellen Sie danach ein Verzeichnis, um dort das SDK zu installieren (`mkdir ~/xport-sdk` zum Beispiel). Wechseln Sie in dieses Verzeichnis mit `cd ~/xport-sdk` und rufen Sie schließlich das Installations-Script auf mit `/mnt/iso/install.sh`.

Falls Ihre Linux-Distribution auf Debian oder Red Hat basiert, sollte jetzt alles ganz einfach sein. Falls nicht (dies war beim Autor der Fall), müssen Sie den gesamten CD-Inhalt in ein beliebiges Verzeichnis kopieren, um dort das Script `install.sh` modifizieren zu können. Letzteres ruft nämlich ein weiteres Script (`scripts/host_depend.sh`) auf, welches für das SDK notwendige Programme installiert. Das Problem besteht

darin, dass dieses Script nur bei Distributionen funktioniert, welche `rpm` oder `dpkg` zur Paketverarbeitung nutzen. Man muss also diejenigen Zeilen von `install.sh`, die für den Aufruf von `host_depend.sh` zuständig sind, auskommentieren (indem man ein „#“ am Zeilenanfang einfügt). Für die SDK-Version 2.0.0.0 handelt es sich um die Zeilen 167 bis 171, fügen sie dort „#“ am Zeilenanfang ein. Nun können Sie die Installation wieder aufnehmen. Bei diesem Vorgehen müssen Sie selbst die für das SDK notwendigen Programme installieren, je nachdem, welche Fehlermeldungen Sie erhalten. Sobald die Installation beendet ist, können Sie das Image mit `umount /mnt/iso` entfernen.

## Elektor Patches

Sobald das SDK einmal installiert ist, fügen wir die Demo-Applikationen für diesen Artikel hinzu (diese befinden sich dann in `/home/elektor`). Beginnen Sie mit dem Download des benötigten Archivs von der Elektor-Seite [3] und entpacken Sie es in ein beliebiges Verzeichnis (z.B. `/tmp`). Dadurch wird ein Ordner namens „Elektor-Xport“ erzeugt. Gehen Sie in diesen Ordner und geben Sie das Kommando `./elektor_install.sh dossier_d'installation_du_SDK` ein. Im Erfolgsfall erscheint die Meldung „SUCCESS“.

## Partitionierung des Flash-Speichers

Der Flash-Speicher des XPort kann auf verschiedene, von Lantronix vorgegebene (oder selbst definierte) Arten partitioniert werden. Der Autor hat `Kernel + romfs + JFFS2` gewählt. Details zu

# Entwicklung mit dem XPort Pro Modul

diesen Angaben finden sich in der Dokumentation des SDK. Eine Partition enthält dabei das Kernel-Image ebenso wie ein rootfs und romfs. Das rootfs ist ein Speicherort, an dem alle Systemdateien abgelegt sind, in etwa mit „C:“ unter Windows vergleichbar. romfs ist ein Verzeichnis-system, welches nur Lesezugriffe erlaubt, wodurch die Systemdateien nicht so leicht zerstört werden können. Die andere Partition nutzt JFFS2 (ein Dateisystem speziell für Flash-Memories), hier können beliebige Daten abgelegt werden. Der Vorteil dieser Vorgehensweise liegt darin, dass man die Daten auf der JFFS2-Partition belassen kann, während man eine neue Firmware flasht. Das ist sehr praktisch.

## Konfiguration und Einsatz des SDK

Bevor man ein Image erzeugen kann, welches dann in den XPort Pro geladen wird, muss das SDK konfiguriert werden. Rufen Sie zunächst das Verzeichnis auf, in welchem Sie das SDK installiert haben. Geben Sie nun das Kommando `source env_m68k-uclinux` ein, um die notwendigen Umgebungsvariablen für das SDK zu setzen. Bei jedem Öffnen eines neuen Terminal-Fensters muss dieses Kommando eingegeben werden. Jetzt kann man die Konfiguration des SDK mit `make menuconfig` starten. Bevor das Menü erscheint, werden verschiedene Fragen gestellt, geben Sie hier jedesmal **Enter** (für den Default-Wert) ein. Danach sollte das Menü angezeigt werden.

Gehen Sie nun in das erste Untermenü, bezeichnet mit *Vendor/Product Selection*. Prüfen Sie, ob *Lantronix* als *Vendor* eingetragen ist und unter *Lantronix Products* der *XPort\_Pro* eingestellt ist. Eine letzte Option erlaubt es Ihnen, das verwendete Profil auszuwählen. Jedes Profil entspricht einem Satz von Parametern für den Linux-Kernel, uClinux und die zu installierenden Applikationen. Wir wählen das Profil *DEVELOPMENT*. Die zu jedem Profil gehörenden Optionen sind in der Datei `900-548c_Linux_SDK_UG.pdf` dokumentiert, welche sich im Ordner mit der SDK-Dokumentation befindet. Rufen Sie nun wieder das Hauptmenü auf, wählen und bestätigen Sie *Exit* mit der Taste **Enter**. Gehen Sie nun in das zweite Menü,

genannt *Kernel/Library/Defaults Selection*, und wählen sie *Customize Kernel Settings* und *Customize Application/Library Settings* mittels der Leertaste aus. Verlassen Sie nun das Konfigurations-Utility durch zweimal *Exit*. Antworten Sie mit *Yes* bei der Frage, ob die Konfiguration gespeichert werden soll.

Nach einigen Sekunden öffnet sich ein zweites Menü zur Konfiguration des Linux-Kernels. Gehen Sie zu *Processor Type and features* und stellen Sie dort die Option *Lantronix CP Manager* auf *General Purpose I/O* ein, um die drei Ein-/Ausgänge direkt nutzen

verläuft befinden sich die flash-fähigen Images im Ordner `linux/images` des SDK. Die Datei `image.bin`, welche den Linux-Kernel enthält, und das Dateisystem `romfs` sind für uns von besonderem Interesse.

Um das Image in den XPort Pro zu laden, benötigen wir einen TFTP-Server. Es soll hier nicht in allen Details erklärt werden, wie ein solcher Server für jedwede Distribution konfiguriert werden muss, hier hilft Google weiter. Der Server muss so eingestellt werden, dass sein `root` in das Verzeichnis `linux/images` des SDK zeigt. Ich empfehle dringend den Einsatz von `tftpd-hpa`; andere Server

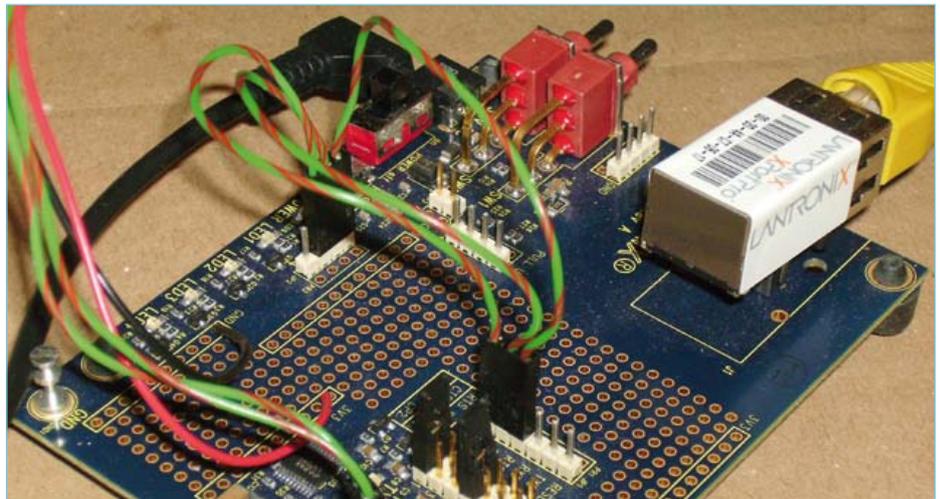


Foto 1. Das XPort Pro-Modul auf dem Demo-Board.

zu können. Zwei von ihnen sind per default als Handshake-Leitungen für den seriellen Port (RTS und CTS) vorgesehen. Falls Sie das Demo-Board von Lantronix verwenden, vergessen Sie nicht, alle Jumper außer 1 und 7 zu ziehen. Markieren Sie auch die Option *Reclaim page before process loading* (um die Fragmentierung des Speichers zu reduzieren) und verlassen Sie das Utility (wie gehabt mit Abspeichern).

Ein drittes und letztes Menü öffnet sich nun, in dem wir nichts verändern müssen – doch auch hier lassen wir die Einstellungen speichern.

Sie können nun das Compilieren durch Eingabe von `make` starten und in aller Ruhe einen Kaffee trinken gehen... Falls alles glatt

haben mir in Zusammenarbeit mit dem XPort Kopfzerbrechen bereitet.

## Jetzt geht's los - alles in den XPort

Wer an diesem Punkt angelangt ist, hat ein via TFTP-Server flash-fähiges Image erzeugt. Für alle, die keine Lust/keine Zeit haben, haben wir ein flash-fertiges Image im von der Elektor-Website downloadbaren Zip-Archiv bereitgestellt [3].

Nun kommen wir endlich zu ersten Praxisübungen. Zuerst wird der *bootloader* dBug konfiguriert. Stellen Sie dazu eine serielle Verbindung mit dem XPort (115.000 Baud, 8 Datenbits, 1 Stoppbit, keine Parität, kein Handshake) über ein Terminal-Programm her. Setzen Sie das Board dann unter Span-

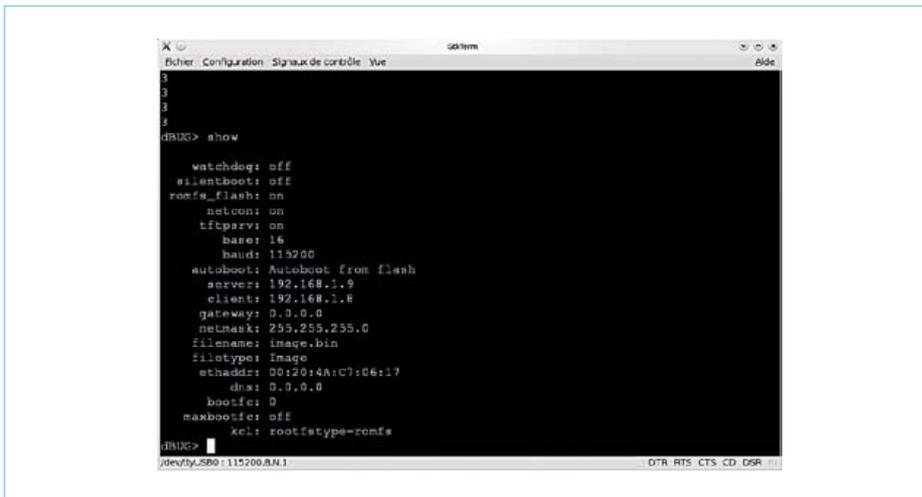


Bild 1. Konfiguration des Bootloaders.

nung. Jetzt heißt es schnell zu sein und innerhalb von drei Sekunden eine Taste zu drücken, um in die Kommando-Ebene des Bootloaders zu kommen (erkennbar durch **dBUG>** auf dem Schirm). Das Kommando *help* tut wie ihm geheißen, *show* zeigt die aktuelle Konfiguration an und *set* die möglichen Optionen. **Bild 1** zeigt die verwendete Konfiguration:

**set watchdog off:** Der Watchdog wird deaktiviert.

**set silentboot off:** Das System startet schneller, aber man muss sehr schnell Ctrl+X drücken, um in das Bootloadermenü zu gelangen (nicht sehr komfortabel während der Entwicklung).

**set romfs\_flash on:** Das Verzeichnissystem romfs wird nicht in das RAM geladen, son-

dern direkt innerhalb des Flashs verwendet. Dies spart viel Speicher und kostet nur wenig Performance (für die Applikationen weiter unten habe ich keinerlei Unterschied feststellen können).

**set autoboot flash:** Systemstart erfolgt automatisch vom Flash aus (es ist auch möglich, über das Netz zu starten).

**set server ip\_server\_TFTP:** Geben Sie eine Adresse in der Form 192.168.0.1 ein.

**set client ip\_client:** Hier wird eine gültige IP-Adresse eingetragen, die der Bootloader für Netz-Operationen benutzt. Diese IP hat nichts mit derjenigen zu tun, die nach dem Start von Linux benutzt wird.

**set netmask Ihre\_Netzwerkmaske:** Falls Ihnen hier nichts einfällt, hilft bestimmt

255.255.255.0 weiter ;-).

**set filename image.bin:** Die in das Flash zu ladende Datei heißt image.bin.

**set kcl rootfstype=romfs:** Dies zeigt dem Linux-Kernel an, dass rootfs vom Typ romfs ist.

Sobald die Konfiguration abgeschlossen ist, können Sie das Kommando dnfl (für *Download from Network & Flash*) eingeben. Die Übertragung der Datei image.bin sollte nun starten. Ist dies nicht der Fall, überprüfen Sie die korrekte Funktion Ihres TFTP-Servers, z.B. mit einem TFTP-Client auf einem anderen Netzrechner. Nach der Übertragung werden Sie gefragt, ob das Flash überschrieben werden soll; antworten Sie hier mit „Ja“. Das Schreiben in das Flash beginnt nun, eine Meldung zeigt an, ob alles gutgegangen ist. Ist dies der Fall, dann drücken Sie Reset. Linux sollte jetzt starten und einige Sekunden später eine Eingabeaufforderung anzeigen. Bravo! Das Schlimmste ist überstanden. Lassen Sie uns jetzt anhand von drei Anwendungen sehen, was man mit dem XPort alles machen kann.

### Der Klassiker: Die blinkende LED

Lantronix stellt ein Utility zur Verfügung, mit dem man die Pins des XPort Pro steuern kann: **cpm** (für *Configurable Pin Manager*). Über eine Datei wird ihm die Konfiguration der drei IO-Pins mitgeteilt. Bezüglich der Syntax verweise ich auf die Dokumentation von Lantronix, Sie können aber einfach auch mal in die mitgelieferte Datei reinsehen.

Unser kleines Demo benutzt ein *Shell*-Script, welches das Utility von Lantronix aufruft, um die IO-Pins zu steuern. Ein Script ist nichts anderes als eine Textdatei, die eine Liste von Befehlen enthält, die der Reihe nach abgearbeitet werden.

Vor dem Start der Demoanwendung sollten Sie unbedingt zwei LEDs an CP1 und CP2 anschließen. Wechseln Sie nun in das Verzeichnis `/home/elektor/ledcpm` (mittels des Kommandos `cd`). Geben Sie `./ledblink.sh` ein. Nun sollten die beiden LEDs abwechselnd blinken.

### Ein bisschen Web

Eine blinkende LED ist ganz nett, aber der XPort verfügt nun einmal über eine Netz-

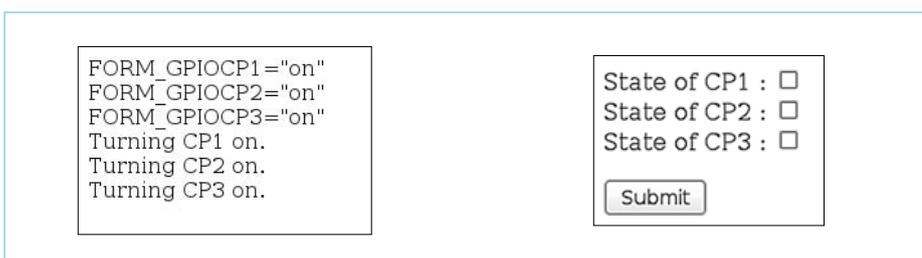


Bild 2. Screenshot der I/O-Steuerung (GPIO).

## Compilieren Ihrer Applikationen

Dieser Artikel enthält nicht das traditionelle *Hello World*, hier soll daher das Compilieren einer Applikation für den XPort erläutert werden.

Sie können Ihre Applikationen entweder in das SDK integrieren, oder aus einem gesonderten Verzeichnis compilieren. Falls Sie Ihre Anwendungen in die Datei `image.bin` einbinden möchten, müssen Sie das SDK nutzen. Doch welche Wahl Sie auch immer treffen, Sie werden ein `makefile` schreiben müssen. Hilfe finden Sie unter [7], oder Sie lassen sich von den beigefügten Dateien inspirieren. Im einfachsten Fall (etwa *Hello World*) können Sie den Compiler `m68k-uclinux-gcc` direkt aufrufen. Allerdings wird das schnell langweilig.

werkbuchse. Also ist es undenkbar, hier keine kleine Web-Anwendung zu testen. Here we go!

Wir verwenden `boa`, einen kleinen Web-Server, der im SDK enthalten ist. Dieser benötigt eine Konfigurations-Datei (`boa.conf`). Die Syntax wird in [4] beschrieben. Sobald diese Datei editiert worden ist, kann man den Server mit dem Befehl `boa -c rep_config` starten, wobei `rep_config` für den Ordner steht, der die Datei `boa.conf` enthält. Unsere kleine Anwendung steuert die Ein-/Ausgänge nun über eine Webseite (siehe Bild 2).

Um den XPort Aktionen ausführen zu lassen, habe ich CGI [5] benutzt. Der XPort kennt weder Perl noch Python noch eine andere Sprache, die man üblicherweise für CGI-Scripts einsetzt. Ich hatte aber weder Lust, die HTTP-Requests manuell zu verarbeiten, noch wollte ich „mal eben“ ein C-Programm schreiben. Eine kurze Internet-Recherche führte zu `proccgi` [6], einem kleinen C-Programm (welches geringfügig modifiziert werden musste, um es mit dem auf dem XPort laufenden Kommando-Interpreter kompatibel zu machen). Dieses behandelt die Requests und erlaubt es, Variablen zu setzen, die man direkt in den Shell-Scripts nutzen kann.

Genug der Theorie, starten wir jetzt unsere kleine Applikation. Vor dem Start des Web-Servers ist eine IP-Adresse für den XPort festzulegen. Falls ein DHCP-Server im Netzwerk existiert, besitzt der XPort möglicherweise schon eine, welche Sie mit dem Befehl `ifconfig` anzeigen lassen können. Im anderen Falle müssen Sie ihm eine Adresse durch `ifconfig eth0 up adresse_ip` manuell zuweisen. Gehen Sie in den Ordner `/home/elektor/webgpio` und geben Sie `boa -c` ein. Der Webserver ist nun bereit.

Sie können ihn über jeden beliebigen Browser durch Eingabe von `http://ip_adresse_xport` in der Adressleiste aufrufen. Wählen Sie die gewünschten Pegel für die Ausgänge aus und klicken Sie auf den Button. Das Script auf dem XPort decodiert die Anfrage und ruft das Utility `cpm` zur Steuerung der Ausgänge auf.

### Temperatursensor im Web

Für diese letzte Applikation hatte ich noch etwas Handfesteres im Sinne. Die Idee

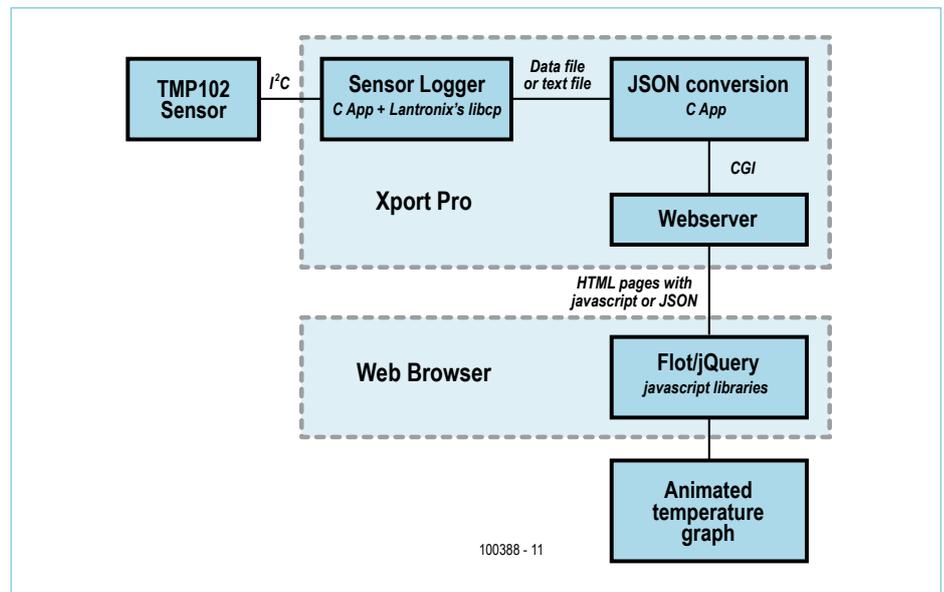


Bild 3. Aufbau der Applikation *webtemp*.

besteht darin, einen Temperatursensor an den XPort anzuschließen und die gemessene Temperatur in grafischer Form über einen Browser zugänglich zu machen. Die Architektur des Ganzen wird als Blockschema in Bild 3 dargestellt. Der eingesetzte Sensor ist ein TMP102. Er ist über einen I<sup>2</sup>C-Bus (SDA->CP1, SCL->CP3) an den

XPort angeschlossen; ADD0 (Adresswahl auf dem I<sup>2</sup>C-Bus) wird mit Masse verbunden. Anfangs dachte ich daran, `rrdtool` zum Erstellen der Grafiken zu verwenden. Allerdings arbeitete es nicht zuverlässig (Probleme mit Beschränkungen durch den XPort). Also bin ich auf die Idee verfallen, das Erstellen der Grafiken auf den Client zu

## µCLinux: Kann man ohne MMU leben?

In einem Datenverarbeitungssystem ist die MMU (*Memory Management Unit*) der Bestandteil, der für die Verwaltung des Speicherzugriffs zuständig ist (heute oftmals in den Prozessor integriert). Das Fehlen einer MMU bedeutet für den XPort Pro gewisse Einschränkungen und bedingt den Einsatz von uCLinux (eine Distribution speziell für Systeme ohne MMU).

Bestimmte Funktionen, die man üblicherweise auf Linux-Systemen findet, kann man daher nicht einsetzen, oder sie bieten nur eine sehr schlechte Performance (`fork`, `mmap`). Was bei der Entwicklung für ein solches System sicherlich am meisten stört, ist das Fehlen von Schutzmechanismen für den Speicher. Falls das Programm abstürzt, wird oftmals das gesamte System unbrauchbar. Die Speicherbereiche für den Kernel und die verschiedenen Prozesse sind nicht voneinander getrennt. Man muss daher bei ungültigen Pointern und anderen Bugs der Speicherverwaltung besonders aufmerksam sein.

## Einsatz der virtuellen Maschine von Lantronix

Laden Sie zunächst das Archiv, welches die virtuelle Maschine (VM) enthält, von [2] herunter und entpacken Sie es in einen Ordner Ihrer Wahl. Installieren Sie danach das Programm **VMware Player**, verfügbar unter [8]. Starten Sie VMware Player und öffnen Sie die Datei **.vmx** von Lantronix. Das Programm fragt, ob die VM verschoben oder kopiert wurde, antworten Sie mit „kopiert“. Die VM startet nun und Sie werden aufgefordert, sich zu identifizieren. Das Passwort lautet PASS. Die Tastatur ist per default auf „QWERTY“ eingestellt, was weitgehend der deutschen Tastaturbelegung entspricht – bei diesem Passwort geht also auch mit einer deutschen Tastatur alles gut. Falls die Tastaturbelegung einmal geändert werden muss, öffnet man das Konfigurations-Fenster in *System->Preferences->Keyboard*.

Auf der Registerkarte *Layout* stellen Sie die Ihrem Keyboard entsprechende Tastaturbelegung ein und verschieben den Eintrag mittels *Move Up* in der Liste nach oben. Schließen Sie danach das Fenster über den Button *Close*. Danach brauchen Sie sich über die Tastatur keine Gedanken mehr zu machen.

Um nun mit der Arbeit zu beginnen, starten Sie ein Terminal via *Applications->Accessories->Terminal*. Das SDK befindet sich in **/home/lantronix/linuxsdk**. Ein letzter Punkt: Wenn Sie den TFTP-Server konfigurieren, vergessen Sie nicht, dass dieser auf der virtuellen Maschine installiert werden muss. Sie müssen daher deren IP-Adresse benutzen, um sich beim Server anzumelden. Jetzt sind Sie am Zug!

übertragen, in diesem Fall den Browser.

Auf Seiten des XPorts emuliert ein C-Programm einen I<sup>2</sup>C-Bus an den IO-Pins. Das Utility *cpm* ist leider nicht schnell genug, daher habe ich die Bibliothek *libcp* von Lantronix verwandt. Die Temperaturdaten werden in regelmäßigen Intervallen in einer Datei gespeichert. Ein Webserver (*boa*) läuft auf dem XPort.

Auf Seiten des PCs setzen wir eine bunte Mischung von HTML, CSS und Javascript ein. **JavaScript** fragt die Temperaturdaten vom Server auf dem XPort ab. Dieser wiederum ruft ein kleines C-Programm auf, welches die Daten des Sensortreibers nach **JSON** umsetzt (ein Format für den Datenaustausch im Web), anschließend schickt er die konvertierten Daten zurück. Der Javascript-Code ruft wiederum die Bibliothek **Flot** auf, um die Temperaturänderung grafisch darzustellen. Ein Button auf der Web-Seite löst eine manuelle Aktualisierung der Grafik aus, über den anderen wird dieser Vorgang automatisiert (Intervall 1 s, siehe **Bild 4**).

Alle (zumindest ein wenig) kommentierten Quellcodes stehen auf der Elektor-Website zur Verfügung [3].

Wechseln Sie zum Ordner **/home/elektor/webtemp**, um diese Anwendung zu starten. Beginnen Sie mit dem Start des Web-Servers durch **boa -c . &**, starten Sie dann die Kommunikation mit dem Sensor durch Eingabe von **tmp102 -q -F /tmp/temp.dat -n0**. Rufen Sie dann einen Browser Ihrer Wahl auf ([http://ip\\_adresse\\_xport](http://ip_adresse_xport)) und klicken Sie einen der Buttons an.

Dieser Artikel kann nicht erschöpfend sein,

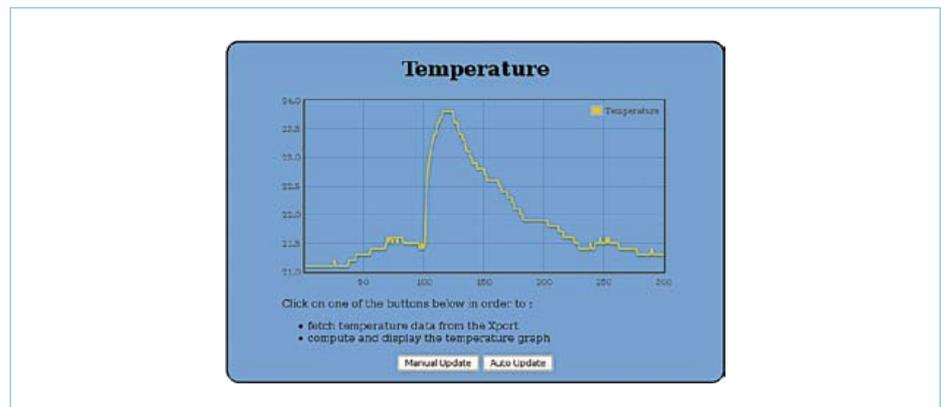


Bild 4. Screenshot der Temperaturmessung.

aber für einen Start mit dem XPort Pro genügen die Infos bestimmt. Trotz seiner Schwächen kann der XPort vieles. Es kommt

nur darauf an, dass Sie Ihre eigenen Ideen „XPortieren“!

(100388)

### Weblinks

- [1] [www.lantronix.com/device-networking/embedded-device-servers/xport-pro.html](http://www.lantronix.com/device-networking/embedded-device-servers/xport-pro.html)
- [2] <http://forums.lantronix.com/forumdisplay.php?f=20>
- [3] [www.elektor.de/100388](http://www.elektor.de/100388)
- [4] [www.boa.org/documentation/boa-2.html#ss2.3](http://www.boa.org/documentation/boa-2.html#ss2.3)
- [5] [http://de.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://de.wikipedia.org/wiki/Common_Gateway_Interface)
- [6] [www.fpx.de/fp/Software/ProcCGI.html](http://www.fpx.de/fp/Software/ProcCGI.html)
- [7] [www.makelinux.net/make3/](http://www.makelinux.net/make3/)
- [8] [http://downloads.vmware.com/d/info/desktop\\_downloads/vmware\\_player/3\\_0](http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0)
- [9] [www.elektor.de/100475](http://www.elektor.de/100475)

„Elektor ist Wissen für Profis und für die, die es werden wollen. Der perfekte Einstieg in mein Studium!“

– Christian, 19 Jahre, Abiturient –



**30%**  
Preis-  
Vorteil

**Elektor ist Wissen für Profis.**

Sichern Sie sich jetzt Ihren Wissensvorsprung durch ein Abo!

**Ihre Vorteile im Abo:**

- Preisvorteil: Schüler und Studenten sparen **30%** gegenüber dem Einzelkauf
- Abonnenten erhalten exklusiv auf spezielle Elektor-Produkte bis zu **40% Rabatt**
- Keine Ausgabe verpassen: Elektor kommt pünktlich, bequem und zuverlässig frei Haus
- Immer up-to-date: Lesen Sie jede Elektor-Ausgabe vor allen anderen



[www.elektor.de/studentenabo](http://www.elektor.de/studentenabo) • Tel. 02 41/88 909-0

**elektor**

# Blaulicht für Modellautos

## Freie Fahrt im Notfall



Von Ludwig Libertin (A)

Nur wenige Bauteile genügen für ein blaues Wechsellicht, das der Polizei, dem Notarzt und der Feuerwehr auf der Straße Vorrang verschafft. Die Modelle der Einsatzfahrzeuge, die damit ausgerüstet sind, wirken fast so realistisch wie ihre Vorbilder.

Notarztwagen, Polizeifahrzeuge und Feuerwehren bahnen sich im Notfall ihren Weg durch den Straßenverkehr, indem sie Blaulicht und Martinshorn einschalten. Auch Fahrzeuge im Modellbau sollen sich möglichst detailgenau an ihren großen Vorbil-

dem orientieren. Das blaue Wechsellicht, hier aufgebaut mit zwei blauen LEDs, ist für Einsatzfahrzeuge ein charakteristisches Ausstattungsmerkmal. Das Martinshorn, das ebenfalls dazugehört, wird vielleicht in einem späteren Miniprojekt realisiert werden.

In den einzelnen Ländern unterscheiden sich die blauen Rundumleuchten an Einsatzfahrzeugen in ihrer Anzahl und ihrem Blinkrhythmus. Wir haben eine Variante gewählt, bei der zwei blaue Leuchten wechselweise doppelblinken. Eine Leuchte blinkt zwei Mal kurz nacheinander, danach blinkt die andere Leuchte ebenfalls zwei Mal kurz nacheinander. Die beiden Blaulichter wechseln einander ab, solange das Fahrzeug im Einsatz ist. Diese Variante lässt sich auf Modell-Einsatzfahrzeuge ohne großen Aufwand übertragen.

Die Schaltung besteht normalerweise ausschließlich aus Standard-Bauelementen, SMDs kommen nicht vor. Auch Einsteiger, die ihre ersten Schritte in die Welt der Elektronik unternehmen, können sich an dieses Miniprojekt wagen.

### Schaltung

Die Steuerung des Blaulichts übernimmt ein digitaler CMOS-Baustein 4060 (siehe Bild 1), er gehört seit Jahrzehnten zum Standard-Sortiment. Da der 4060 einen Oszillator mit nachgeschalteter Teilerkette an Bord hat, müssen extern nur noch wenige passive Bauelemente hinzugefügt werden. Die Blinkfrequenz hängt von C2, R2 und R3 ab, mit den in der Schaltung angegebenen Werten hat das Signal am niedrigen-

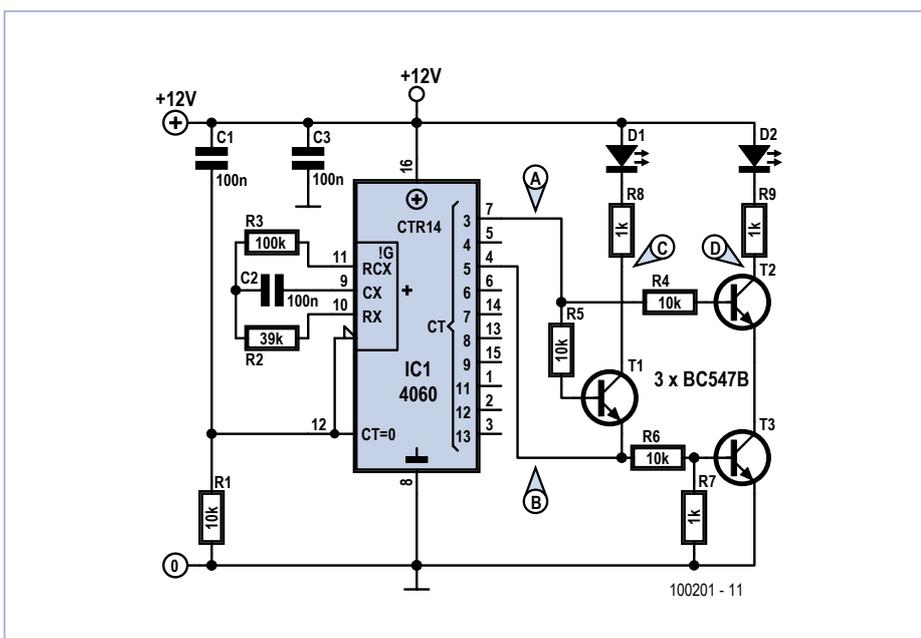


Bild 1. Taktgeber ist ein CMOS-Zählerbaustein 4060, der auch einen Oszillator an Bord hat.

ten Teiler Ausgang (Q3, Anschluss 7) die Frequenz 8 Hz. Die Frequenz des Signals an Q5 (Anschluss 4) beträgt ein Viertel, also 2 Hz. Über die Transistoren T1, T2 und T3 werden die Ausgangssignale des 4060 so verknüpft, dass eine LED zweimal blinkt, während die andere LED pausiert. Dann tauschen die LEDs ihre Rollen, die andere LED blinkt zweimal, die erste LED hat Pause. T2 und T3 bilden gemeinsam ein NAND-Gatter für LED D2. Diese LED blinkt zweimal im Rhythmus des Signals an Ausgang Q3, vorausgesetzt das Signal an Ausgang Q5 ist hoch. Transistor T1, der LED D1 steuert, hat die Funktion eines NAND-Gatters für die Signale der Ausgänge Q3 und Q5, gleichzeitig invertiert er das Signal, das von Q5 kommt. In **Bild 2** sind die Signale als Zeitdiagramm dargestellt. Die mittlere Stromaufnahme beträgt bei

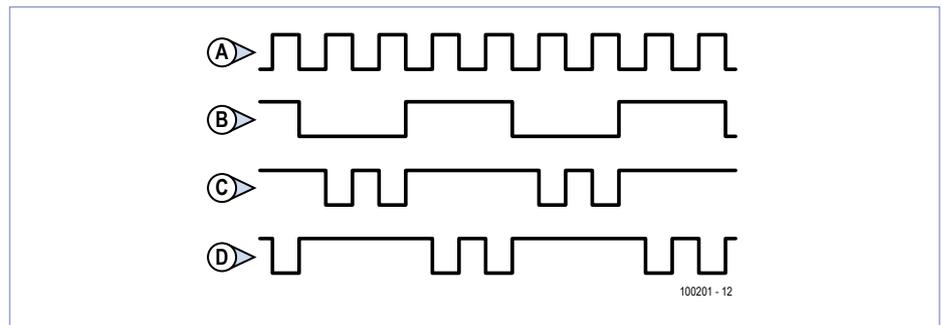


Bild 2. Das Diagramm stellt die Signalabläufe und die Blinkphasen der LEDs dar.

12 V ungefähr 6 mA. Solange die Betriebsspannung über 7,5 V liegt, arbeitet die Schaltung wunschgemäß. Allerdings leuchten die LEDs bei niedriger Spannung weniger hell, und wegen der Reihenschaltung von T2 und T3 macht sich ein geringerer Helligkeitsunterschied bemerkbar.

Es kann aufschlussreich sein, ein wenig zu experimentieren. Wenn das Signal an Ausgang Q4 das Signal an Q5 ersetzt, entsteht

ein einfaches Blinklicht. Wird das Signal an Q5 gegen das Signal an Q6 getauscht, blinken die LEDs vier Mal nacheinander.

Falls der Platz im Modellfahrzeug ausreicht, kann die Schaltung auf einem Stück Lötpunktraster-Platine aufgebaut werden. Nur wenn es extrem klein sein muss, ist der Entwurf einer SMD-Platine eine Überlegung wert.

(100201)gd

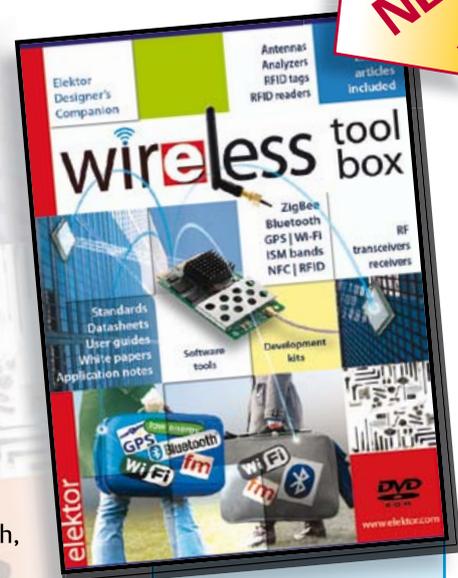
Anzeige

# Wireless-Toolbox

➔ Alles Wissenswerte zum Thema Wireless auf DVD-ROM

Kontaktlose Übertragungssysteme haben in allen Bereichen der modernen Gesellschaft (Transport, Handel, Gesundheitswesen, Haushalt u. a.) Einzug gehalten. Diese DVD-ROM umfasst zum Thema „Wireless“ sorgfältig zusammengetragene technische Dokumentation (1,4 GB) in englischer Originalsprache sowie über 90 originale Elektor-Artikel in deutscher Sprache, die es Ihnen ermöglichen, die eigenen Schaltungen drahtlos miteinander kommunizieren zu lassen.

Je nach zu überwindender Signaldistanz für den kontaktlosen Datenaustausch, wählen Sie den hierfür geeigneten Übertragungsmodus: Bei nur wenigen Zentimetern ist es Near Field Communication (NFC) oder Radio Frequency Identification (RFID), im 10-m-Bereich ist es Bluetooth, Wi-Fi oder ZigBee und bei über 1000 km sind es GPS-Daten.



NEU!

ISBN 978-90-5381-268-6  
€ 32,50 • CHF 44,20

Weitere Infos & Bestellung unter [www.elektor.de/shop](http://www.elektor.de/shop)

# PC-Oszilloskop

## Neue Modellreihe PicoScope 3000 von Pico

Von Harry Baggen (Redaktion NL)

Der britische Messgeräte-Spezialist Pico Technology hat die Mittelklasse seiner PC-Oszilloskope runderneuert. Die Reihe PicoScope 3000 umfasst leistungsstarke, am USB-Bus betriebene Oszilloskope, die sich durch hohe Sample-Frequenzen und integrierte Signalgeneratoren auszeichnen. Wir haben das PicoScope 3206B unter die Lupe genommen.

Die Leistung und Flexibilität der PC-Oszilloskope (auch „USB-Oszilloskope“ genannt) ist in den letzten Jahren sprunghaft gestiegen. PC-Oszilloskope sind fast immer kostengünstiger als klassische Geräte mit Display, darüber hinaus haben sie in mancher Hinsicht die Nase vorn. Abgesehen vom PC besteht die Hardware aus einem unscheinbaren Kästchen, während das Messergebnis in voller Größe auf dem PC-Bildschirm erscheint. Weil PC-Oszilloskope auch zusammen mit Note- oder Netbooks einsatzfähig sind, ist die galvanische Trennung vom 230-V-Stromnetz kein Problem. Andererseits haben auch die Klassiker unter den Oszilloskopen ihre starken Seiten, sie kommen insbesondere bei hohen Sample-Frequenzen zum Tragen. Die Stand-alone-Geräte können für extrem hohe Arbeitsgeschwindigkeiten konstruiert werden, während PC-Oszilloskope wegen der relativ langsamen USB-Datenübertragung bei schnellen Aktivitäten benachteiligt sind. Der Geschwindigkeitsengpass lässt sich umgehen, wenn im Messsystem genügend Hochgeschwindigkeitsspeicher verbaut wird. Die damit einhergehende Komplexität der Hardware hat allerdings zur Folge, dass der Kostenvorteil schrumpft.

### PicoScope 3000

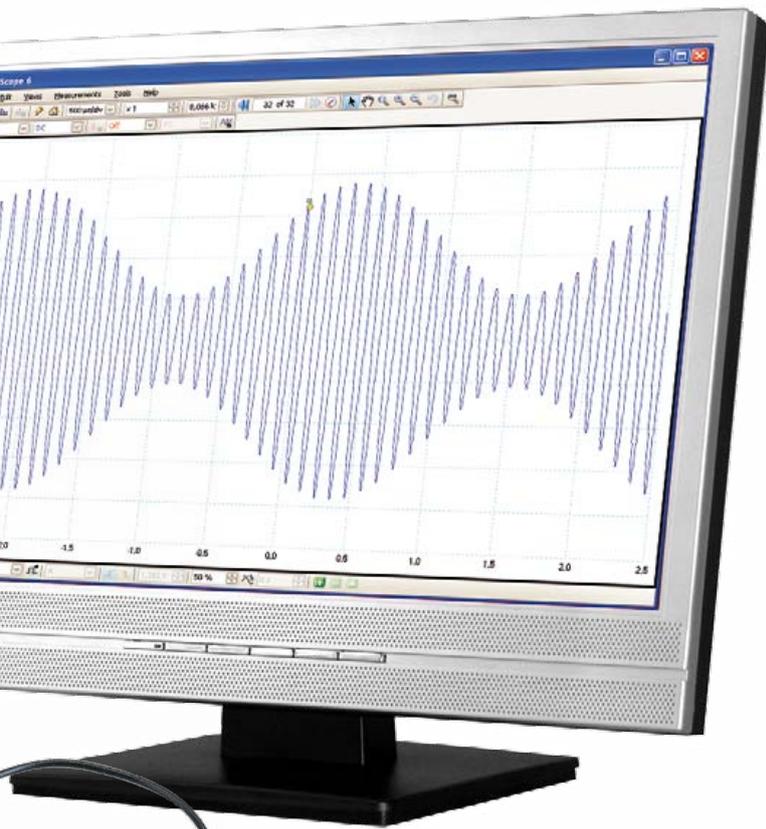
Pico Technology ist ein britisches Unternehmen, das sich auf die Daten-Akquisition und Messtechnik in Verbindung mit dem PC spezialisiert hat. Die Produktpalette der PC-Oszilloskope reicht vom Einsteiger-Modell für 150 € bis zum superschnellen System, das über 15.000 € kostet. Dazwischen liegt die absatzstarke Mittelklasse, die unlängst eine Rundum-Modellpflege erfuhr. Der Reihe PicoScope 3000 gehören sechs PC-Zweikanal-Oszilloskope an, hier liegt die Preisspanne zwischen 500 € und 1100 €. Bei allen Modellen dieser Reihe beträgt die maximale Sample-Frequenz 500 MSamples/s (Megasamples pro Sekunde). Die Bandbreite hängt vom Modell und Preis ab, 60 MHz, 100 MHz oder 200 MHz stehen zu Wahl. Alle Modelle werden in einer A- und B-Version angeboten, der Unterschied liegt in der Art des integrierten Signalgenerators. Wäh-



rend die A-Versionen Signale mit festen Signalformen liefern, sind in den B-Versionen so genannte AWG-Signalgeneratoren eingebaut (AWG=Arbitrary Waveform Generator). AWG-Signalgeneratoren können beliebige, vom Benutzer gestaltbare Signalformen erzeugen. Alle Modelle verfügen über interne Pufferspeicher, die Kapazitäten liegen modellabhängig zwischen 4 MSamples und 128 MSamples. Für unsere Begutachtung überließ uns Pico das Flaggschiff der PicoScope-Reihe 3000, das PicoScope 3206B.

### Hardware

Verständlich, dass wir uns erlaubt haben, das Gehäuse des PicoScope 3206B zu öffnen, um das Innenleben zu inspizieren (Bild 1). Der erste Blick fiel auf die perfekt abgeschirmten Eingangsstufen, der zweite Blick auf den FPGA Spartan-6 von Xilinx. Der FPGA verarbeitet die digitalisierten Messsignale, außerdem generiert er die Daten für die Signale, die das Gerät als Signalgenerator liefert. Die Sample-Daten legt der FPGA in einem DDR3-Speicherchip ab, der als Messdaten-Puffer dient. Ein 8-bit-A/D-Wandler ADS5010 digitalisiert die Eingangssignale beider Kanäle. Dieser ADC-Typ ist für vier Kanäle ausgelegt, bei Einkanal-Betrieb kann er aus dem Eingangssignal bis zu 1000 MSamples/s generieren. Im PicoScope 3206B wird nur die halbe Leistung genutzt, vermutlich weil mit der Wandlerrate auch der Strombedarf sinkt. Der Gerätebetrieb am USB-Port ist ohne externe Energiequelle nur bei niedriger Stromaufnahme möglich. Die vom FPGA in der



Funktion als Signalgenerator kommenden Daten werden von einem 12-bit-DAC AD9706 in analoge Signale umgesetzt, er kann bis zu 175 MSamples/s verarbeiten. Die Kommunikation mit dem PC übernimmt ein USB-Transceiver CY7C68013A von Cypress.

### Software

Die Software, die zu den PC-Oszilloskopen von Pico gehört, ist bei allen Modellen der 3000er-Reihe identisch. Eventuelle Funktionseinschränkungen sind nur durch die Eigenschaften der Hardware bedingt. Die typische Applikation ist der Oszilloskop-Bildschirm, auf dem die Messsignale grafisch in Echtzeit und Farbe auf weißem Hintergrund dargestellt werden. Eine zweite Betriebsart simuliert einen nachleuchtenden Oszilloskop-Bildschirm auf dunklem Hintergrund. Die Triggerung bietet viele Einstellmöglichkeiten und Feinheiten. Der Inhalt des Pufferspeichers ist abschnittsweise zugänglich, das gespeicherte Messsignal kann sehr detailliert betrachtet werden. Weitere Highlights sind der XY-Modus sowie der FFT-Analyser, der das Messsignal kontinuierlich als Spektrum darstellt. Zur Wahl stehen mehrere Analyse-Fenster wie beispielsweise Hamming oder Blackman. Die Software kann auch Daten serieller Busse darstellen, zum Beispiel I<sup>2</sup>C, RS232, SPI oder CAN. In diesem Modus erscheinen gleichzeitig das originale Messsignal und die zugehörigen decodierten Daten auf dem Bildschirm. Nach einem Mausklick auf das Generator-Symbol öffnet sich das Hauptfenster des integrierten Signalgenerators, dort sind die Signalform und die Signalspannung einstellbar. Die B-Versionen der PicoScope-3000-Familie bieten darüber hinaus die Möglichkeit, mit der Maus beliebige Signalformen zu gestalten. Die Signalformen können in CSV-Dateien gespeichert und von dort abgerufen werden.

Zu den zahlreichen weiteren Extras, von denen hier nur wenige erwähnt werden können, gehören das Anlegen spezieller Messprofile und Masken, zum Beispiel für Toleranz-Messungen in Produktionsprozessen. Auch Messungen an den vom Benutzer gestalteten Signalen oder Signalverläufe nach mathematischen Funktionen sind möglich.

### Kästchen mit Zubehör

Der Lieferumfang des PicoScope 3206B schließt zwei Messfühler, ein USB-Kabel, eine CD und eine Schnellstart-Anleitung ein. Das außen eher unscheinbare Gerät im Format eines Taschenbuchs ist mit vier BNC-Buchsen auf der Stirnseite und einem USB-Anschluss auf der Rückseite ausgestattet. Die Software lässt sich von der CD in wenigen Minuten installieren. Nachdem das PicoScope über das USB-Kabel mit dem PC verbunden ist, muss nur noch das Programm gestartet werden.

Das Programm macht schon auf den ersten Blick einen übersichtlichen und aufgeräumten Eindruck. Die Bedienelemente sind auf Symbol- und Menüleisten oberhalb und unterhalb des großflächigen Bereichs verteilt, auf dem die Signale abgebildet werden. Die ungewohnte Größe der dargestellten Signale, die natürlich auch von der Größe des PC-Bildschirms abhängt, ist beeindruckend. Wir haben sie nach kurzer Zeit als selbstverständlich und unentbehrlich empfunden.



Bild 1. Blick in das PC-Oszilloskop 3206B von Pico. Auffällig sind die Abschirmungen der Eingangsstufen.

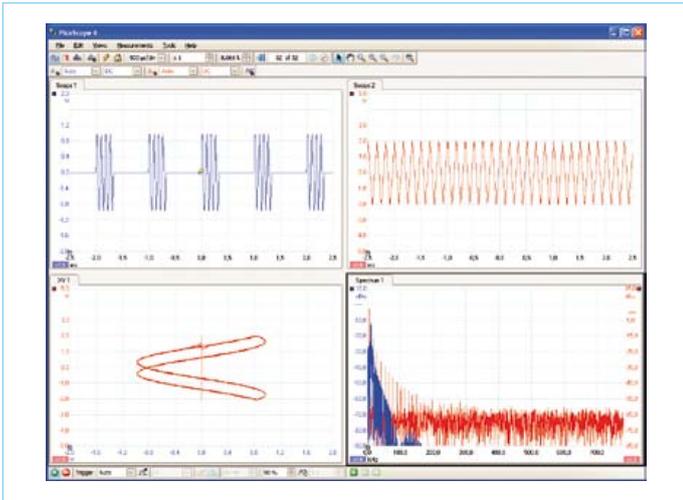


Bild 2. Hier sind vier Programmfenster mit unterschiedlichen Signal-Darstellungen geöffnet.

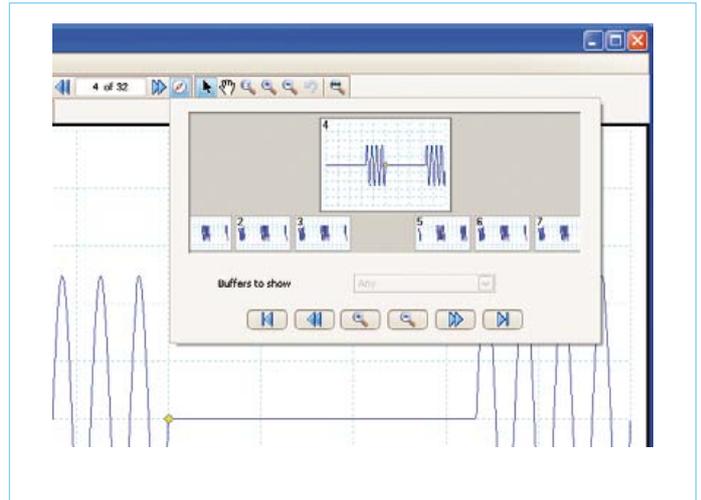
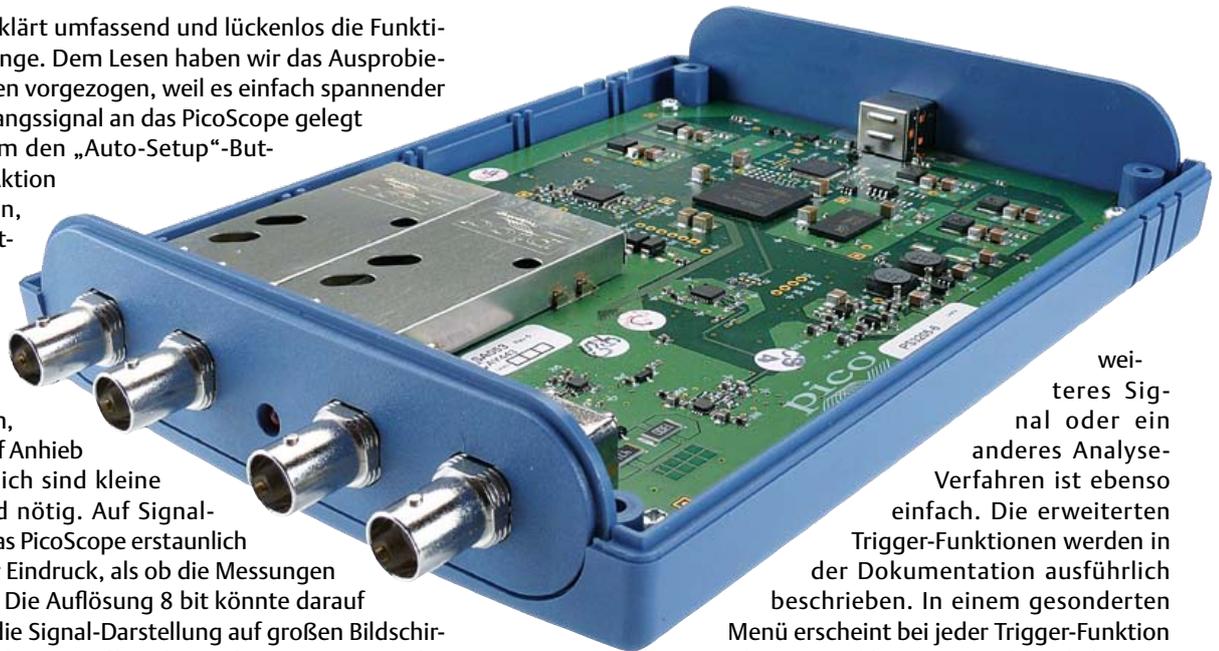


Bild 3. Der Inhalt des Signal-Pufferspeichers wird abschnittsweise in vergrößerbaren Miniatur-Ansichten dargestellt.

Die Dokumentation erklärt umfassend und lückenlos die Funktionen und Bedienvorgänge. Dem Lesen haben wir das Ausprobieren und Experimentieren vorgezogen, weil es einfach spannender ist. Wir haben ein Eingangssignal an das PicoScope gelegt und auf dem Bildschirm den „Auto-Setup“-Button angeklickt. Diese Aktion weist das Programm an, die Einstellungen selbstständig zu finden, bei denen das Bild steht und das Signal optimal abgebildet wird. Versuche mit weiteren Signalen bestätigten, dass dies fast immer auf Anhieb gelingt, nur gelegentlich sind kleine Korrekturen von Hand nötig. Auf Signal-Änderungen reagiert das PicoScope erstaunlich schnell, es entsteht der Eindruck, als ob die Messungen in Echtzeit stattfinden. Die Auflösung 8 bit könnte darauf schließen lassen, dass die Signal-Darstellung auf großen Bildschirmen leicht „holprig“ wirkt. Das trifft nicht zu, die interne Interpolation der Sample-Werte, einstellbar zwischen 8 bit und 12 bit, glättet den dargestellten Signalverlauf. Trotzdem bleibt die Signal-Darstellung auch in den Details dem Original treu. Selbst Anwender, die ausschließlich mit konventionellen Oszilloskopen gearbeitet haben, sind schnell mit dem Konzept der Bedienung vertraut. Hersteller Pico hat nicht versucht, die Bedienelemente klassischer Oszilloskope nachzuempfinden. Wir halten dies für eine gute Entscheidung, denn die Kombination konventioneller Drehknöpfe und Schalter mit den für manche Funktionen unentbehrlichen Menüs beeinträchtigt die Übersichtlichkeit. Die Bedienung des PicoScope macht zwar ein generelles Umdenken nötig, doch das gelingt ohne viel Mühe in kurzer Zeit. Die obere Menü-Ebene ist minimalistisch gestaltet, hier sind nur die wichtigsten Funktionen mit einem Mausklick erreichbar. Alle übrigen Funktionen liegen in unteren Ebenen, die Wege dorthin erklären sich selbst. Zum Umschalten zwischen Einkanal- und Zweikanal-Betrieb genügt ein Mausklick, das Hinzufügen eines Fensters für ein



weiteres Signal oder ein anderes Analyse-Verfahren ist ebenso einfach. Die erweiterten Trigger-Funktionen werden in der Dokumentation ausführlich beschrieben. In einem gesonderten Menü erscheint bei jeder Trigger-Funktion eine knappe Erklärung einschließlich einer erläuternden Darstellung. Orientierung über den Inhalt des Signal-Pufferspeichers geben Miniatur-Ansichten, die in einem separaten Fenster erscheinen. Nach dem Anklicken wird der gewählte Signal-Abschnitt im Großformat dargestellt. Der in allen Modellen der PicoScope-Reihe 3000 integrierte Signal-generator kann in vielen Fällen einen separaten Signalgenerator ersetzen. In den B-Modellen sind AWG-Signalgeneratoren eingebaut, die beliebige, in einem Mini-Editor gestaltbare Signalformen erzeugen können. Das Entwerfen eines Sinus-Bursts, mit dem das Ein- und Ausschwingverhalten von Audio-Filtern und Lautsprechern getestet werden kann, dauerte nur wenige Minuten. Obwohl der Sinus-Burst ein komplexes Signal ist, das an die Triggerung erhöhte Anforderungen stellt, hatten wir die perfekte Einstellung erstaunlich schnell gefunden. Die Sample-Frequenz der PicoScope-3000-Modelle beträgt 500 MSamples/s im Einkanal-Betrieb und 250 MSamples/s im Zweikanal-Betrieb. Die hohe Geschwindigkeit macht das Analysieren von

Signalfanken und Ausgleichsvorgängen auch bei höherfrequenten Signalen möglich. Zusammen mit dem Signal-Pufferspeicher eröffnen sich viele Möglichkeiten der Signal-Analyse.

### Perfekte Duos

Jedes Modell der neuen Reihe PicoScope 3000 von Pico ist eine gelungene Kombination eines PC-Oszilloskops und eines Signalgenerators. Das Modell 3204A, Bandbreite 60 MHz, ist schon für 483 € (plus Mehrwertsteuer) erhältlich. Für diesen Preis bekommt der Kunde ein vollwertiges, vielseitiges und flexibles Oszilloskop mit exzellenter Signal-Darstellung (auf dem PC-Bildschirm), das ohne eigene Stromversorgung auskommt. Zusammen mit einem Laptop oder Notebook entsteht ein perfektes mobiles Duo, wie geschaffen zum Beispiel für den Außendienst. Das Modell 3206B hat seine Bewährungsprobe in unserem Labor-Alltag bestanden. Dem Hersteller Pico können wir bescheinigen, dass ihm mit der neuen Reihe PicoScope 3000 ein guter Wurf gelungen ist.

(110268)gd

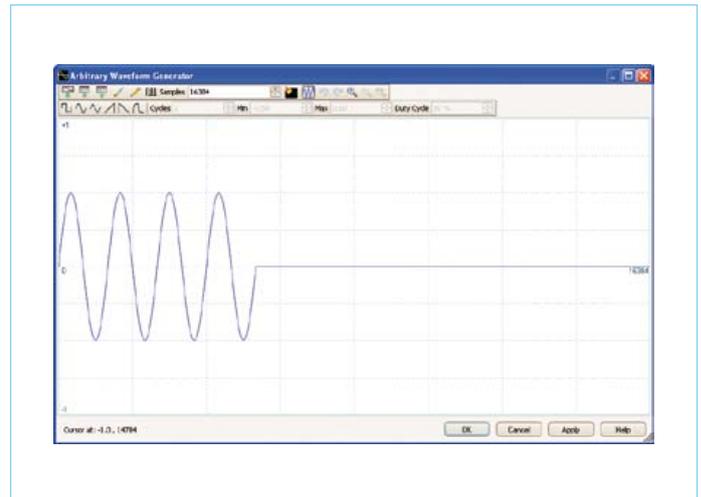


Bild 4. Der Arbitrary Waveform Generator (AWG) erzeugt die in diesem Fenster entworfenen Signalformen.

### Weblink

[www.picotech.com/computer-oscilloscope.html](http://www.picotech.com/computer-oscilloscope.html)

Anzeige

# Lichttechnik für Bühne und Disco

## ➔ Komplett überarbeitete Neuauflage

Auf der Bühne und in der Disco ist Licht nicht nur Beleuchtung, sondern vor allem künstlerisches Gestaltungsmittel. Wie vorzugehen ist, damit Licht beiden Ansprüchen gerecht wird, macht der Autor an vielen Beispielen deutlich.

Das Buch beschreibt, welche Geräte es gibt, wie sie funktionieren und im Rahmen der künstlerischen Arbeit eingesetzt werden. Dabei wird auch auf neuere Entwicklungen wie LED und DMX-RDM eingegangen. Ergänzt wird das mit einer fundierten Aufarbeitung der theoretischen Grundlagen sowie durch zahlreiche Anleitungen für die Praxis, sodass sich dieses Buch auch als Lehrbuch für Veranstaltungstechnik bestens eignet. Einen besonderen Schwerpunkt nimmt das Thema „Sicherheit“ ein. Angefangen von den rechtlichen Grundlagen (VStättVO, BGV C1, DIN-VDE-Normen) über mechanische und elektrische Sicherheit sowie den Brandschutz bis hin zur Notfallorganisation werden alle Aspekte ausführlich abgehandelt.



288 Seiten (kart.)  
Format 17 x 23,5 cm  
ISBN 978-3-89576-228-4  
€ 34,80 • CHF 47,40

Weitere Infos & Bestellung unter [www.elektor.de/shop](http://www.elektor.de/shop)

# E-blocks: RC5 in Flowcode

## Eigene Projekte fernsteuern



RC5 ist ein Kommunikationsprotokoll für Infrarot-Fernsteuerungen. Es wird vor allem bei Fernsehern und häuslicher Elektronik eingesetzt. RC5-kompatible Fernbedienungen gibt es überall preiswert zu kaufen. In diesem Beitrag geht es darum, wie man mit Hilfe der E-blocks eigene Projekte dazu bringen kann, RC5-codierte Infrarot-Signale zu verstehen oder zu generieren.

Von Sean King (UK)

Das RC5-Protokoll wird als Methode der Fernsteuerung von Fernsehern und anderer Consumer-Elektronik schon so lange und breit eingesetzt, dass es eine Art inoffizieller Standard geworden ist. Von daher ist die Frage interessant, wie RC5 funktioniert. Bei der Erläuterung wird die notwendige Hard- und Software beleuchtet, die für den Empfang und die Decodierung von RC5-Signalen erforderlich ist. Doch zunächst einige Aspekte des RC5-Protokolls, die das Verständnis von Hardware und Software erleichtern.

### Protokoll

RC5 ist ein relativ gemächliches Übertragungsverfahren. Jedes Bit benötigt  $1778 \mu\text{s}$  - mit großen tolerierten Abweichungen. Ein normaler RC5-Befehl umfasst 14 bit und benötigt daher etwa 25 ms. In **Bild 1** ist so eine RC5-Bit-Kette aufgeschlüsselt: 2 Start-

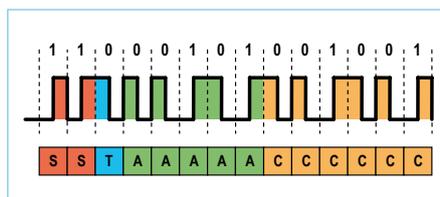


Bild 1. RC5-Message-Struktur.  
Start = 1 + 1. Toggle = 0. Adresse = 5  
(Videorekorder). Kommando = 9.

Bits + 1 Toggle-Bit + 5 Adress-Bits + 6 Kommando-Bits. Die Funktionen der Bits werden nachfolgend beleuchtet.

**Start-Bits:** Die beiden ersten Bits mit dem Wert „1“ stehen für den Beginn des Befehls. Sie dienen gleichzeitig als Zeit-Referenz für die folgenden Bits.

**Toggle-Bit:** Ändert seinen Wert bei jedem neuen Tastendruck auf einer Fernbedienung. Wiederholte Befehle mit gleichem Wert zeigen an, dass die Taste weiter gedrückt gehalten wird. Eine solche Auto-Repeat-Funktion ist bei der Einstellung quasikontinuierlicher Parameter wie der Lautstärke sinnvoll.

**Adress-Bits:** Die fünf Adress-Bits dienen zur Selektion der Empfängers. Für einige Adressen haben sich Konventionen herausgebildet:

Fernseher	= 0
Videorekorder	= 5
Satelliten-Receiver	= 8

**Kommando-Bits:** Mit den sechs Kommando-Bits können  $2^6 = 64$  verschiedene Befehle übertragen werden. Üblicherweise entsprechen die Zahlen einer numerischen

### Elektor Produkte & Service

- E-block RC5-Infrarot-Board (EBo60)
- E-block dsPIC/PIC24-Multiprogrammer (EBo64)

- E-block grafisches LCD (EBo58)
- E-block Switch-Board (EBo07)

Details unter [www.elektor.de/e-blocks](http://www.elektor.de/e-blocks)

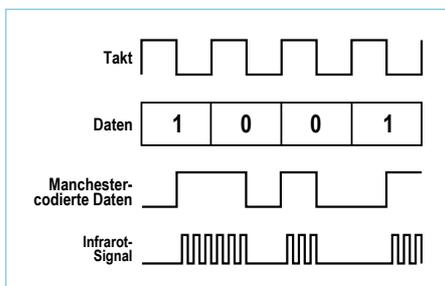


Bild 2. RC5-Codierung.

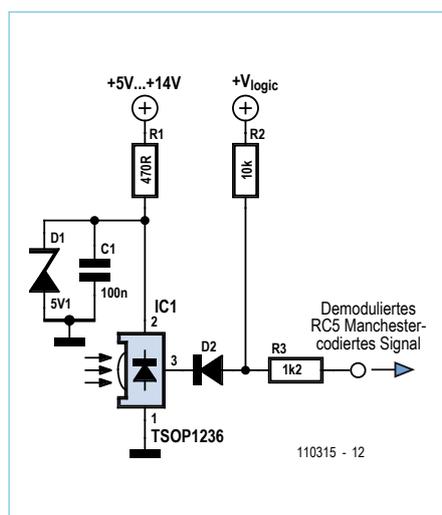


Bild 3. Der IR-Empfänger.

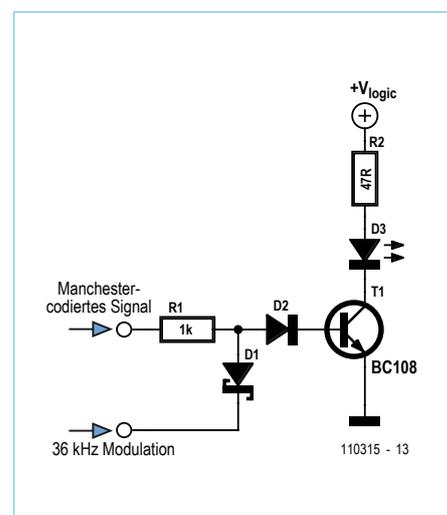


Bild 4. Der IR-Sender.

Tastatur ihrem binär codierten Wert. Für die anderen Werte gibt es ebenfalls etablierte Konventionen.

### Codierung

Im Prinzip hat es der IR-Sender viel einfacher als der IR-Empfänger. Der Sender gibt nämlich das Zeitverhalten und die Amplitude vor, aber der Empfänger muss sich sowohl dem Sender anpassen als auch die Effekte der (wechselnden) Übertragungsstrecke kompensieren. Ein grundlegendes Problem vieler Signalübertragungen ist, dass der Empfänger erkennen muss, wann er mit der Abtastung ankommender Signale beginnen soll. Er muss dazu den Takt des Datenstroms rekonstruieren. Zur Vereinfachung wird der RC5-Code in der so genannten „Manchester-Codierung“ übertragen. Dabei wird der Wert jedes Bits durch die Richtung der Flanke in der Mitte seines Zeitschlitzes bestimmt (siehe **Bild 2**). Dieses Verfahren garantiert, dass zumindest eine Signalflanke pro Bit auftritt, was die zeitliche Synchronisation des Empfängers (auf Kosten einer größeren Bandbreite) erleichtert. Die Werte des vorausgehenden und nachfolgenden Bits bestimmen also darüber, ob am Beginn oder Ende eines Zeitschlitzes eine zusätzliche Flanke eingefügt werden muss.

### Modulation

Das „gemanchesterte“ RC5-Signal muss noch mit einem Modulationssignal von 36 kHz unterlegt werden. Diese Modulationsfrequenz erleichtert es dem Empfänger, störendes IR-Licht anderer Quellen (mit anderen Frequenzen) auszufiltern und eine automatische Verstärkungsregelung durchzuführen.

### RC5-Hardware

Die dank hoher Integration sehr einfache Empfängerschaltung von **Bild 3** basiert auf dem IR-Empfänger-IC TSOP1236, das auf IR-Licht mit 36 kHz reagiert. Trotz seiner nur drei Anschlüsse steckt im Inneren ordentlich Elektronik, denn es entfernt die Modulation aus dem Signal und sorgt für die automatische Verstärkungsregelung unterschiedlich starker IR-Signale. Vereinfacht gesagt verhält sich IC1 wie eine Art optischer Transistor in Emitterschaltung, dessen Basis lichtgesteuert ist, dessen Pin 1 als Pseudo-Emitter an Masse liegt und dessen virtueller Kollektor Pin 2 das fertig demodulierte RC5-Signal liefert. Dieses Signal wird an einen Eingang eines Mikrocontrollers geführt, denn es muss ja noch decodiert werden.

Die Signale werden sogar wie bei einem Transistor invertiert: Wenn ein 36-kHz-IR-Signal empfangen wird, zieht der Ausgang Pin 3 den Pull-up-Widerstand R2 auf Masse. In **Bild 4** ist der noch einfachere Sender zu sehen. Wenn am Mikrocontroller-Ausgang eine „1“ an R1 anliegt, wird die Basis von T1 mit einem 36-kHz-Signal gesteuert, was entsprechende Lichtblitze der IR-Diode D3 auslöst. Das 36-kHz-Modulationssignal muss der Mikrocontroller natürlich ebenfalls bereit stellen.

### Entwicklungssystem

Passend für dieses Projekt sind sowohl Sender als auch Empfänger auf dem neuen RC5-Infrarot-Board EB060 des E-blocks-Systems untergebracht. Praktischerweise ist dort

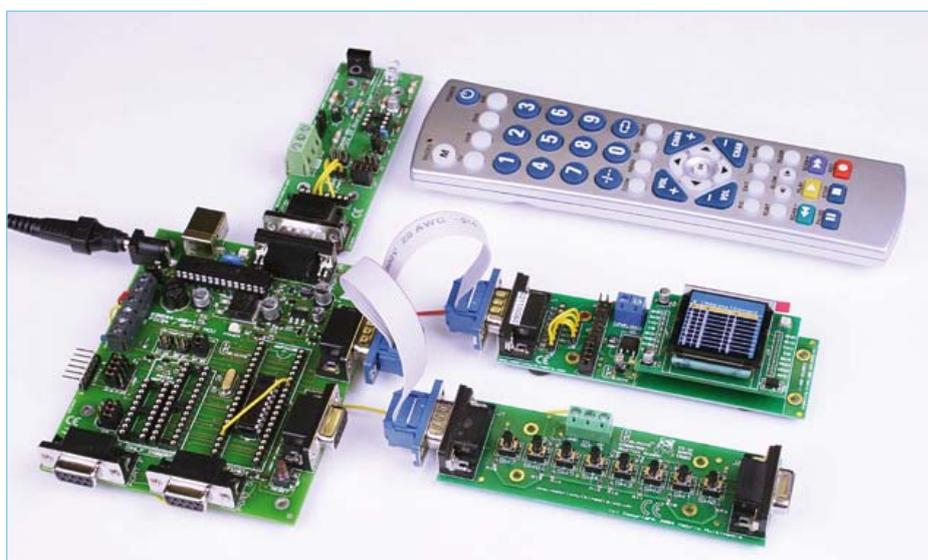


Bild 5. Projekt-Aufbau mit den E-blocks-Modulen EB060, EB064, EB058 und EB007.

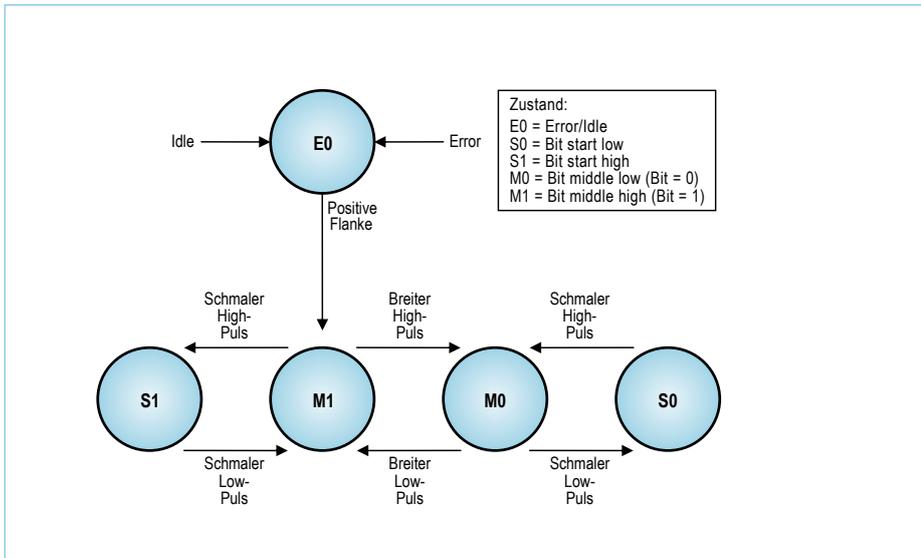


Bild 6. RC5-Decodierung mit State-Machine.

auch ein Sockel für einen extra PIC10F-Chip vorgesehen, den man als Modulator für den Sender verwenden kann, was dem eigentlichen Mikrocontroller des Projekts Arbeit abnimmt. Der komplette Aufbau ist **Bild 5** zu entnehmen. Man erkennt den dsPIC-Multiprogrammer, ein Modul mit grafischem LCD und ein Switch-Board, die sich Port A teilen. An Port E ist das RC5-Infrarot-Board angeschlossen. Der ausgewählte dsPIC ist ein 18-poliger dsPIC30F2011-Chip. Ein konventioneller 8-bit-PIC täte es auch, aber ein modernerer Chip macht sich auf einem neuen Board besser. Nach dem codierenden Sender geht es jetzt weiter mit dessen Gegenstück.

### Signal-Decodierung

Das RC5-Verfahren ist relativ gutmütig gegenüber Timing-Abweichungen. Das war

für die simplen Fernbedienungen früherer Zeiten mit ihren ungenauen Oszillatoren wichtig, da deren Frequenz sogar von der (abnehmenden) Spannung der Batterieversorgung beeinflusst wurde. Mit den beiden Start-Bits eines RC5-Befehls lässt sich eine Synchronisation des Empfängers durchführen, sodass die Zeiten für die folgenden relevanten 12 Bits passend eingestellt sind. Das Verfahren ähnelt dem bei seriellen Schnittstellen. Dennoch können sich hier kleine Zeitfehler über die folgenden Bits akkumulieren und so zu Decodierungsfehlern bei den letzten Bits führen. Aufgrund der garantierten Flanken in der Mitte der Zeitschlitze führt das Manchester-Verfahren zu vier verschiedenen Impulsformen, aus denen die Nachricht zusammengesetzt ist:

1. breiter High-Impuls;

Tabelle 1. Impulsbreiten			
	Min.	Ideal	Max.
Breit (µs)	1334	1778	2222
Schmal (µs)	444	889	1333

2. schmaler High-Impuls;
3. breiter Low-Impuls;
4. schmaler Low-Impuls.

Mit dieser Beschreibung kann man relativ große Zeitabweichungen tolerieren, bevor die Unterscheidung der Impulse problematisch wird. Man braucht nicht einmal die exakte Taktfrequenz zu bestimmen und verhindert so akkumulierende Zeitfehler. Die Software benötigt zur Impulsunterscheidung lediglich eine simple State-Machine. Die wichtigste Aufgabe ist es dabei, einen kurzen von einem langen Impuls zu unterscheiden. Die hierfür benötigten Zeiten haben demnach eine Toleranz von ¼ bit = 444 µs. Die detaillierten Zeiten stehen in **Tabelle 1**.

### State-Machine

Zur Decodierung des Signals benötigt man ein Verfahren, das einen IR-Datenstrom in numerische Werte verwandelt. Ein klassischer Ansatz hierfür ist die so genannte State-Machine - recht unelegant „Endlicher Automat“ genannt. Die prinzipielle Arbeitsweise zeigt das Diagramm von **Bild 6**. Der Automat kennt fünf Zustände. Die Pfeile zwischen den Zuständen zeigen die zulässigen Pfade an. Die Pfeile sind mit den Ereignissen beschriftet, die einen Zustandswechsel auslösen. Die Zustände M0 und M1 repräsentieren die erfolgreiche Deco-

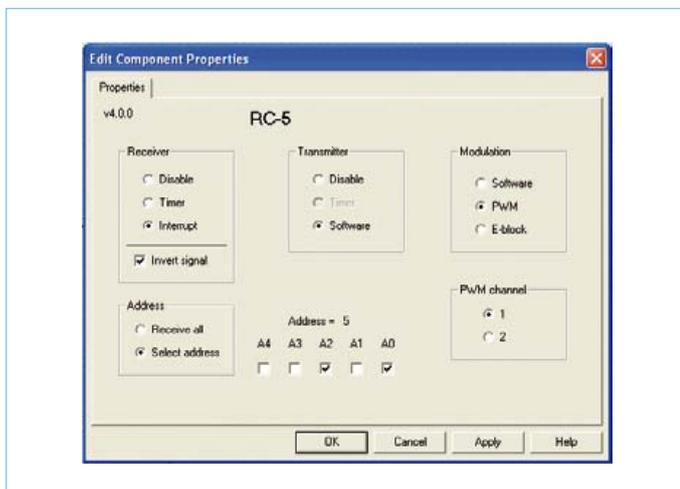


Bild 7. RC5-Einstellungen in Flowcode.

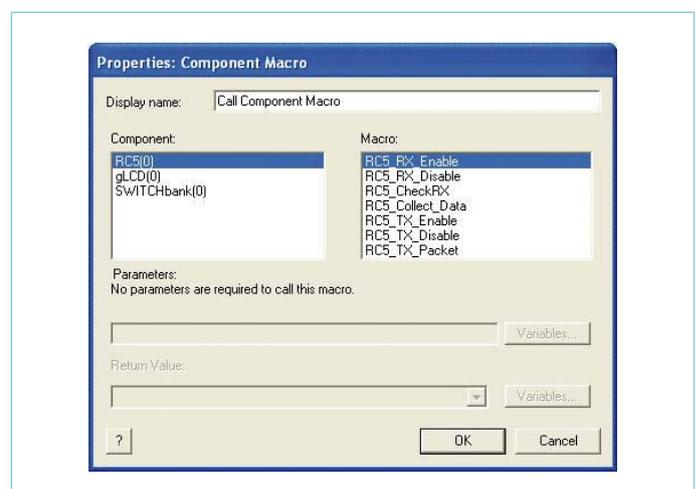


Bild 8. RC5-Einstellungen in Flowcode.

dierung eines Bits (mit den Werten „0“ und „1“). Immer wenn der Automat eines dieser beiden Zustände erreicht, kann das entsprechende Bit in ein Schieberegister geschrieben werden.

### Fehlerfallen

Wenn ein Ereignis auftritt, das nicht zu den gangbaren Pfaden passt, dann wird ein Fehler ausgelöst und zurück zum Zustand E0 gesprungen. Auch bei zu kurzen oder zu langen Impulsen wird ein Fehler erkannt, der zum Zustand E0 führt.

### Beispiel von Bild 6

Wenn kein Signal empfangen wird, befindet sich der Automat via Idle-Pfad (Leerlauf) im Zustand E0. Nur bei einer positiven Flanke wird dieser Zustand verlassen und zu M1 gewechselt. Gleichzeitig wird eine „1“ ins Schieberegister geschoben. Folgt ein kurzer positiver Impuls, wird in S1 gewechselt. Ein weiterer kurzer negativer Impuls versetzt den Automaten wieder in M1 und schiebt eine weitere „1“ in das Register.

Damit sind schon die beiden Start-Bits decodiert. Nun kommt der bedeutungstragende Teil des Befehls an die Reihe.

Ein breiter positiver Impuls führt zum Zustand M0 und schiebt für das Toggle-Bit den Wert „0“ in das Empfangsregister. Die State-Machine decodiert auf diese Weise auch die folgenden 11 bits. Sie ist für dieses Projekt so implementiert, dass sie eine Anzahl von Variablen mit Werten füllt.

### Software in Flowcode für dsPIC/PIC24

Die Projekt-Software wurde mit Flowcode für dsPIC/PIC24 realisiert. Hier ist ein Makro enthalten, das einige verbreitete RC5-Varianten verarbeiten kann. Diese Flowcode-RC5-Komponente enthält Optionen für den Empfang und das Senden von RC5-Signalen. Sogar an die Inversion der Signale bei unserem Empfänger wurde gedacht. Außerdem gibt es einen Filter für die Empfangsadresse (siehe **Bilder 7** und **8**). Die zugehörigen Makros erlauben den Aufbau eines Befehls aus seinen drei Bestandteilen (Toggle, Adresse und Befehlscode), der dann am Stück gesendet werden kann. Ana-

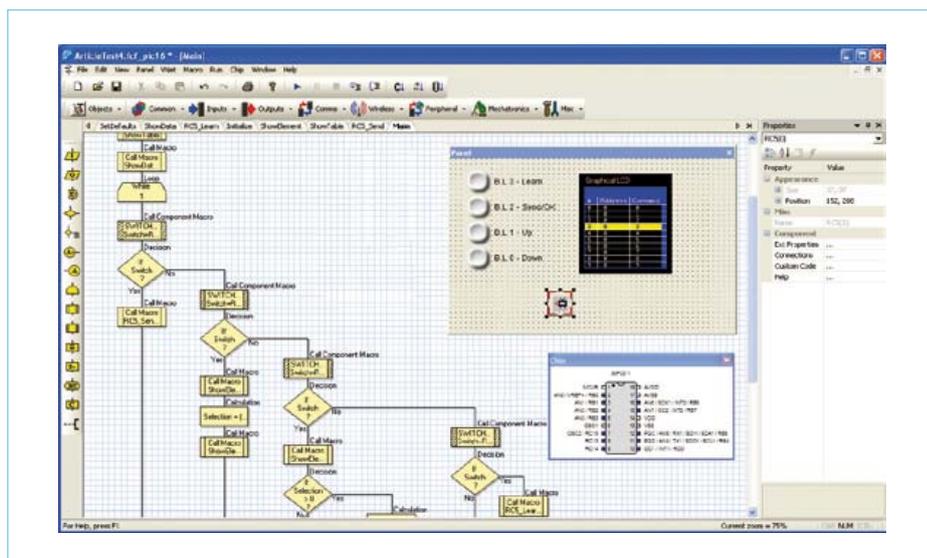


Bild 9. Das Flowcode-Programm.

log erlauben die Makros für den Empfang das Decodieren und Zerlegen der Befehlskette in ihre Teile. Zu Debugging-Zwecken wurde ein grafisches Display als E-blocks-Modul hinzugefügt. Damit kann man sehr schön die empfangenen Bits darstellen. Die vielen möglichen Zeichen machen zudem die Darstellung von decodierter Adresse und Befehlscode möglich, sodass man sich leicht vom ordnungsgemäßen Funktionieren der Software überzeugen kann.

Eigentlich ist es schon erstaunlich, dass ein per IR gesteuertes Gerät in der Sonne stehen kann und dennoch auf eine Fernbedienung reagiert. Das klappt deshalb, weil aus dem Signalgemisch nur IR-Anteile mit 36 kHz gefischt werden. Dank dieser Modulation und der Manchester-Codierung kann ein Mikrocontroller mit einfachstem Empfänger auch aus einem IR-Signalgemisch mit hoher Sicherheit die richtigen Anteile zur Decodierung heranziehen.

### Das Programm

Das mit Flowcode erstellte Programm (siehe **Bild 9**) nutzt die RC5-Komponente zur Realisierung einer RC5-Fernbedienung, die von anderen konventionellen Fernbedienungen Befehle lernen und diese anzeigen kann. Das grafische LCD (EB058) zeigt eine Tabelle von zehn RC5-Befehlen mit der Default-Adresse „0“ (Fernseher) und den Befehls-Codes „0“ bis „9“ (entspricht den Dezimaltasten 0 bis 9). Das Programm wird mit vier Tastern des Switch-Boards EB007 gesteuert, die zusätzlich zum Display an Port BL angeschlossen sind. Die ausgewählte Nachricht wird dann vom RC5-Infrarot-Board EB060 gesendet, wenn der entsprechende Taster (Switch 2)

betätigt wird. Die gelbe Auswahlmarkierung kann mit den Tastern 1 und 0 aufwärts und abwärts bewegt werden. Der Lern-Modus wird mit Taster Nr. 3 aktiviert. Das angezeigte Menü enthält drei Optionen: Die Taster 0 und 1 dienen zur Auswahl und mit Taster 2 wird die Auswahl bestätigt. Im Lern-Modus wird die Tabelle mit einem roten Auswahlbalken dargestellt. Die ausgewählte Position erhält dann die Werte des empfangenen gültigen RC5-Befehls, worauf die Auswahl gelb und der Lern-Modus verlassen wird. Der Lern-Modus kann auch vorzeitig mit Taster 2 abgebrochen werden. Die Drahtverbindung führt von Sockel U1 zu Pin RD0/INT1 an PORTA am Miscellaneous-Steckverbinder und wird für den Empfangs-Interrupt verwendet. Die Signale RC13 und RC14 an diesem Steckverbinder dienen zur Steuerung des Sendens und zur Modulation. Während dieses Programm zur Behandlung von RC5-Signalen dient, kann man jede Menge Spezialfunktionen nach Wunsch einbauen. Das beschriebene Flowcode-Programm lässt sich kostenlos von der Elektor-Webseite zu diesem Artikel [1] herunterladen.

### Fazit

Eigentlich sind Codierung und Decodierung von RC5-Signalen keine einfachen Aufgaben, aber dank einer in Flowcode realisierten State-Machine wird dieses Vorhaben auf ein Stück nicht besonders komplexer Software reduziert.

(110315)

### Weblink:

[1] [www.elektor.de/110315](http://www.elektor.de/110315)

# Hexadoku

## Sudoku für Elektroniker

Wir haben ja jetzt schon ein paar sonnige Tage erlebt, doch könnten wir davon noch viel mehr gebrauchen. Nicht zuletzt deshalb, weil es kaum etwas Schöneres gibt, als in der Sonne auf dem Balkon oder der Terrasse zu sitzen. Ein wenig Unterhaltung gehört natürlich auch dazu, und da kommt unser Hexadoku gerade richtig!

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

durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt.

Wer das Rätsel löst - sprich die Zahlen in den grauen Kästchen herausfindet - kann wie jeden Monat einen Hauptpreis oder einen von drei Trostpreisen gewinnen!

### Mitmachen und gewinnen!

Unter allen internationalen Einsendern mit der richtigen Lösung verlosen wir

- einen **ELEKTOR-Gutschein** im Wert von 100 € und
- drei **ELEKTOR-Gutscheine** im Wert von je 50 €.

### Einsenden

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail, Fax oder Post an:

Elektor – Redaktion Süsterfeldstr. 25 52072 Aachen  
 Fax: 0241 / 88 909-77 E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!  
 Einsendeschluss ist der 30. Juni 2011!

### Die Gewinner des Hexadokus aus dem Aprilheft stehen fest!

Die richtige Lösung ist: B9A65.

Der Elektor-Gutschein über 100 € geht an: M. Borrias aus De Bilt (Niederlande).

Einen Elektor-Gutschein über je 50 € haben gewonnen: Bertrand Danet, Ralf Möckel und Sven Skjenneberg.

Herzlichen Glückwunsch!

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

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

Kontakt: Tel. 0511/61 65 95-0 – Fax 0511/61 65 95-55 – E-Mail: service@id-medienservice.de



Entwicklung industrietauglicher Software und Hardware sowie Elektronik  
**03303/212166** oder [www.jasys.de](http://www.jasys.de)

### Bausätze zu ELEKTOR 1986 bis heute!

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

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



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

Mail: [info@geist-electronic.de](mailto:info@geist-electronic.de)

Shop: [www.geist-electronic.de](http://www.geist-electronic.de)



**VTS – ELEKTRONIK GbR**  
 Preiswerter Leiterplatten Service  
[info@vts-elektronik.de](mailto:info@vts-elektronik.de)  
[www.vts-elektronik.de](http://www.vts-elektronik.de)

[www.anttronic.de](http://www.anttronic.de)



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

### Alles Spule!

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



**www.spulen.com**

**Geist Electronic-Versand GmbH**  
 Tel.: 07720/36673  
 Fax: 07720/36905  
 Mail: [info@geist-electronic.de](mailto:info@geist-electronic.de)  
 Shop: [www.geist-electronic.de](http://www.geist-electronic.de)



Unser Leistungsspektrum:

- Diverse Schrittmotoren ab Lager lieferbar
- Eigene Entwicklung von Schrittmotorendstufen in versch. Ausführungen
- Netzteile, Endschalter, geschirmte Kabel u.v.m.

Kundenspezifische Lösungen entwickeln wir schnell und preiswert

Fon: 0241/4091800 • Fax: 0241/4091803  
[www.mechapro.de](http://www.mechapro.de) • [info@mechapro.de](mailto:info@mechapro.de)



**A Rohde & Schwarz Company**

- Oszilloskope
- Netzgeräte
- RF-Messtechnik
- Systemgeräte

- Konfigurierbare digitale & analoge Schaltaktoren für die Hausautomation
- Seriell ansteuerbar über Modbus von PC, SPS oder µController
- 4,3" TFT Touch-Display mit I²C-Extender



[www.elconeq.de](http://www.elconeq.de)  
 Tel. 02832-9784 301

**Hard- & Softwareentwicklung**

- µController-Module (8051-komp.) z.B. 64kFlash, 2xCAN, 2xUART, I²C, RTC, 32k-FRAM, ID
- I²C-Erweiterungen digital/analog



Bauelemente, Bücher & Bastelideen für Kid's  
[www.kleiner-ingenieur.com](http://www.kleiner-ingenieur.com)



**schlanke Preise - fettes Angebot**  
**Schnellversand ohne Mindestumsatz**  
 Bauteile - Ersatzteile - Zubehör  
 Fernbedienungen - Zeilentrafos  
 HDTV DVB-T Sat Audio/Video  
 Überwachungstechnik und vieles mehr.  
[www.Laehn-Versand.de](http://www.Laehn-Versand.de)



**Preisgünstige Bauteile**  
 zusätzlich über 3.500 Auslaufartikel  
 zum Super-Sonderpreis  
[www.elcomp.at](http://www.elcomp.at)

**Great Value in Test & Measurement**  
[www.hameg.com](http://www.hameg.com)



160 pages of tech audio articles  
 Self, Linkwitz, Cordell, Pass a.o.  
[www.linearaudio.net](http://www.linearaudio.net)

**Baugruppenbestückung**  
 vom Prototypen bis zur Serie  
**FS-ELECTRONIC.de**

**Anzeigenschluss:**  
**19-07-2011**

**Erscheinungstermin:**  
**12-08-2011**

Die Buchung einer Anzeige beinhaltet einen kostenlosen Eintrag auf der Website von Elektor, inklusive eines Links zu Ihrer Seite.

**Reservieren Sie jetzt Ihre Jahresbuchung**

# www.elektor.de

# Kristallines Licht: Die bemerkten

## A Note on Carborundum.

*To the Editors of Electrical World:*

SIRS:—During an investigation of the unsymmetrical passage of current through a contact of carborundum and other substances a curious phenomenon was noted. On applying a potential of 10 volts between two points on a crystal of carborundum, the crystal gave out a yellowish light. Only one or two specimens could be found which gave a bright glow on such a low voltage, but with 110 volts a large number could be found to glow. In some crystals only edges gave the light and others gave instead of a yellow light green, orange or blue. In all cases tested the glow appears to come from the negative pole, a bright blue-green spark appearing at the positive pole. In a single crystal, if contact is made near the center with the negative pole, and the positive pole is put in contact at any other place, only one section of the crystal will glow and that same section wherever the positive pole is placed.

There seems to be some connection between the above effect and the e.m.f. produced by a junction of carborundum and another conductor when heated by a direct or alternating current; but the connection may be only secondary as an obvious explanation of the e.m.f. effect is the thermoelectric one. The writer would be glad of references to any published account of an investigation of this or any allied phenomena.

NEW YORK, N. Y.

H. J. ROUND.

Bild 1. Der Bericht von H. J. Round in der „Electrical World“ von 1907.

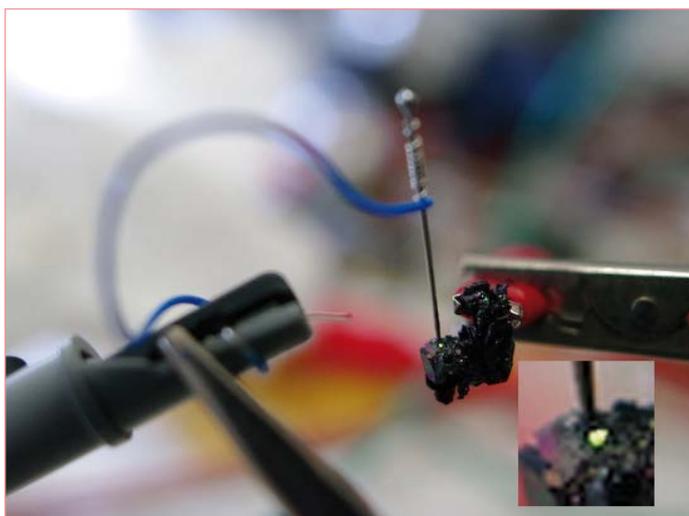


Bild 2. Michael Lippert replizierte die Experimente mit Siliziumkarbid-Kristallen von Round aus dem Jahre 1907. Die negativ gepolte Nadelspitze berührt die Oberfläche des Kristalls, der mit dem positiven Pol der Spannungsquelle verbunden ist. Bei einer Spannung von 9 V und einem Strom von 30 mA konnte er ein grünes Leuchten an der Kontaktstelle beobachten (Wikimedia Commons).

Von Andrew Emmerson (UK)

Wer hat`s erfunden? Gemeint ist kein Kräuterbonbon, sondern die Leuchtdiode, kurz LED. Und die Antwort lautet überraschenderweise: Niemand!

Zurück zu den Anfängen: Wer bemerkte zuerst die Lichtabstrahlung bei Halbleitern? Das war im Jahre 1907 Henry J. Round, ein Assistent von Marconi. Aber Round war mit vielen anderen Dingen beschäftigt, so dass das Phänomen später wiederentdeckt werden musste. Sein Bericht in **Bild 1** entstand, während er mit Spitzendetektoren für Funkzwecke experimentierte. Er bemerkte, dass ein Strom durch zwei Punkte an einem Siliziumkarbid-Kristall (= SiC) den Kristall zum Leuchten bringt. Er meinte, dass entweder die Spannung oder aber der thermische Effekt des Stroms an der Verbindung zwischen Kristall und anderen Materialien für das Licht verantwortlich wäre.

Damals interessierte man sich nicht für Rounds Beobachtungen. Erst 102 Jahre später wiederholte der Biologe und Freizeit-Elektroniker Michael Lippert diese Versuche (siehe **Bild 2**). Das Bild und Nachbau-Tipps finden sich in seinem Blog [1]. Zum Glück sind diese Experimente ungefährlich und recht einfach. Man braucht lediglich eine Spannungsquelle ab 9 V mit einer Strombegrenzung auf 30 mA (Vorwiderstand). Die notwendigen Siliziumkarbid-Kristalle fand Michael Lippert sehr preiswert bei eBay. Wenn man die Nadelspitze über den Kristall bewegt, findet man bestimmt eine Stelle, an der es leuchtet. Das ist zumindest in dunklen Räumen sichtbar. Die Farbe des Lichts hängt dabei von der Stelle am Kristall sowie vom fließenden Strom ab und reicht von orange über gelb bis zu grün. Mit etwas Geschick kann man sogar richtig hell leuchtende Stellen finden.

Doch zurück zur Halbleiterlicht-Geschichte. In den 1920er-Jahren machte der Autodidakt und Radiotechniker Oleg Losev (**Bild 3**) die gleiche Entdeckung wie Round etliche Jahre zuvor. Er beobachtete, dass Kristalldioden für Funkempfänger leuchten, wenn man Strom hindurch schickt. Anders als Round ging er der Elektrolumineszenz am Physikotechnischen Institut in Leningrad richtig auf den Grund und veröffentlichte einige Aufsätze zur Strom/Spannungs-Charakteristik von SiC-Dioden und ihrem Verhältnis zur Lichtemission. Sein Artikel „Luminous carborundum detector and detection effect and oscillations with crystals“ in der britischen Zeitschrift „Philosophical Magazine“ vom November 1928 brachte es auf immerhin 21 Seiten.

War die Elektrolumineszenz für Round nur ein Kuriosum, so hatte Losev keine Zweifel an der praktischen Bedeutung seiner Entdeckung. Kleine nichtvakuierte Lichtquellen für niedrige Spannungen (<10 V) konnten mit ihrer schnellen Reaktion Aufgaben abdecken, die mit normalen Glühlampen undenkbar waren. Sein Patent auf ein „Lichtrelais“ von 1927 war gedacht für „schnelle telegrafische und telefonische Kommunikation, Übertragung von Bildern und andere Anwendungen, bei denen ein lumineszierender Kontakt von einem modulierendem Strom gesteuert wird“ — eigentlich

# swerte Vorgeschichte der LED



Bild 3. Oleg Losev war ein russischer Funktechniker, der als zweiter entdeckte, dass Dioden bei Stromfluss Licht aussenden (Wikimedia Commons).



Bild 4. Das sowjetische Patent von Oleg Losev aus dem Jahre 1927 für sein „Lichtrelais“ (Quelle: PatentsFromRU.com).

schon die Vorwegnahme moderner optoelektronischer Kommunikationstechnik. Tragischerweise endeten seine vielversprechenden Untersuchungen vorzeitig, da er bei der deutschen Belagerung von Leningrad 1942 im Alter von 39 Jahren an Hunger starb.

Waren demnach Round und Losev die Väter des Halbleiterlichts? Nicht wirklich, da ihre Untersuchungen nicht weiterführten. Außerdem sind LEDs auf Basis von Siliziumkarbid bis heute nicht sehr effizient. Von daher hätten die Entdeckungen von Round und Losev damals kaum zu praktisch verwertbaren Resultaten oder gar industriell hergestellten Produkten führen können.

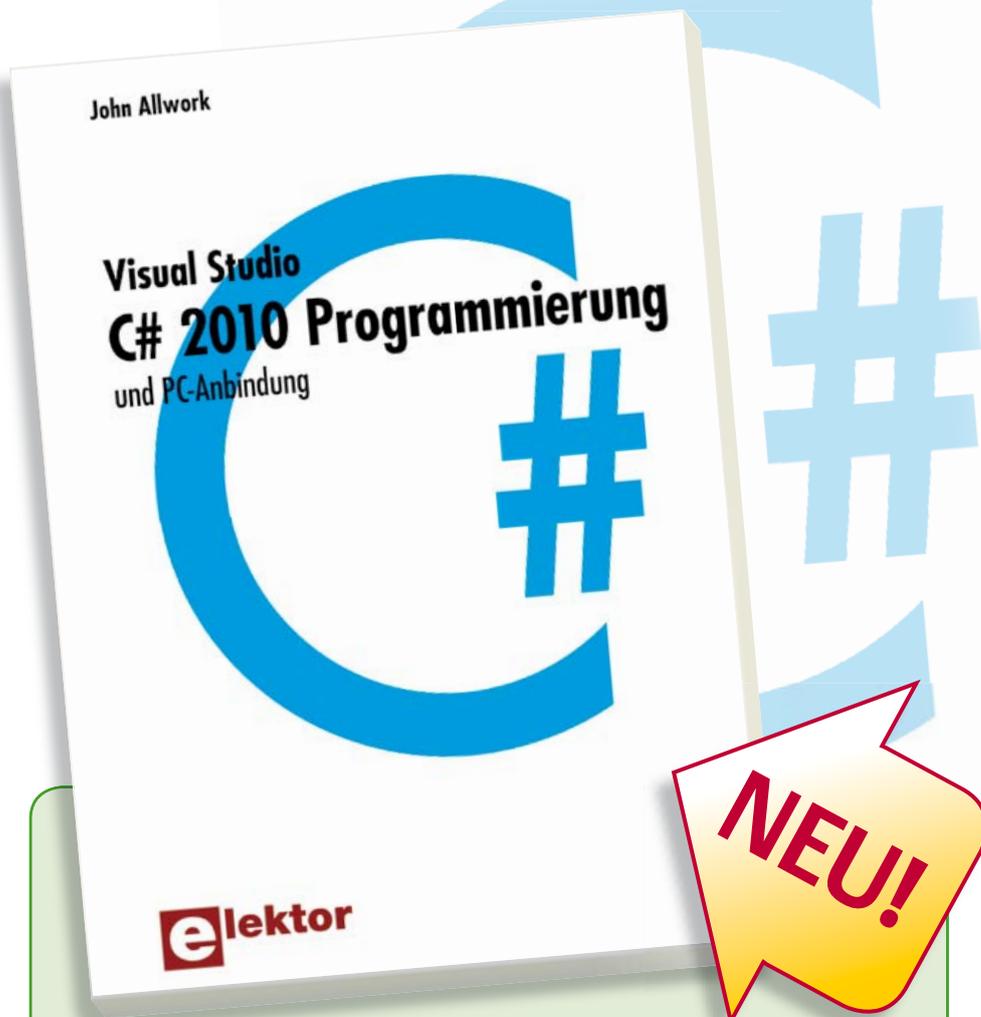
(110021)

## Weblink

- [1] Michael Lippert: Selbstbau einer SiC-LED: [www.dlip.de/?p=99](http://www.dlip.de/?p=99)

# Starke Stücke

Die ganze Welt der Elektronik  
in einem Shop!



Visual Studio

## C# 2010 Programmierung und PC-Anbindung

Ziel dieses Buches ist, auf einfache Weise zu zeigen, wie mit der populären Hochsprache C# ein PC programmiert werden kann. Am Anfang beschreibt das Buch Datentypen und Programmsteuerungen, die dann um fortschrittliche Konzepte wie die objektorientierte Programmierung, Threads, die Internetkommunikation und Datenbanken erweitert werden. Alle verwendeten Code-Beispiele können kostenlos von der Elektor-Webseite heruntergeladen werden. Mit den vorgestellten Konzepten können auf einfache Weise eigene Ergebnisse erzielt und somit der Lernprozess unterstützt werden. Auch professionelle Software-Tools von Microsoft stehen kostenlos zur Verfügung. Die Anwender stellen für Microsoft's Visual Studio 2010 eine Vielzahl von Steuerungen, Eigenschaften, Methoden und Ereignissen bereit. Viele Programmbeispiele, praktische Übungen und Links auf Support-Videos vervollständigen dieses Buch.

349 Seiten (kart.) • ISBN 978-3-89576-244-4 • € 44,00 • CHF 59,90

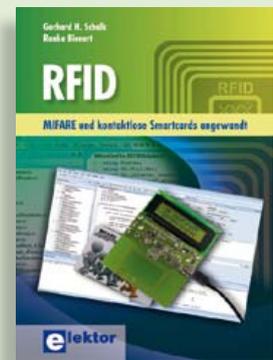


Von Top-Autor und Entwickler  
Burkhard Kainka

## Basiskurs BASCOM-AVR

BASCOM und AVR-Controller sind ein starkes Team! Was immer man entwickeln möchte, meist hat ein ATmega schon das Wichtigste an Board: Ports, Timer, AD-Wandler, PWM-Ausgänge und serielle Schnittstelle, RAM, Flash-ROM und EEPROM, alles ist reichlich vorhanden. Und BASCOM macht die Anwendung zu einem Kinderspiel. Auch komplexe Peripherie wie LCD, RC5 und I<sup>2</sup>C lassen sich mit wenigen Befehlen nutzen.

223 Seiten (kart.) • inkl. Software-CD  
ISBN 978-3-89576-238-3 • € 39,80 • CHF 54,20



EMIFARE und kontaktlose  
Smartcards angewandt

## RFID

MIFARE ist die weltweit meistgenutzte RFID-Technologie. Dieses neue Buch bietet einen praxisorientierten und umfassenden Einstieg in diese Technologie. Die einleitenden Kapitel behandeln u. a. die physikalischen Grundlagen, die relevanten Normen, das RFID-Antennendesign, die Sicherheitsaspekte und die Kryptografie. Das vollständige Hardware- und Softwaredesign eines Readers ist ausführlich beschrieben.

464 Seiten (kart.) • ISBN 978-3-89576-219-2  
€ 56,00 • CHF 76,20

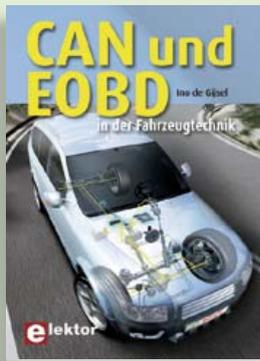


Mikrocontroller selber programmieren

## Das MSP430 Mikrocontroller Buch

Moderne Mikrocontroller werden immer leistungsfähiger und können vielfältige Aufgaben übernehmen, für die vor wenigen Jahren noch ein kompletter Computer nötig gewesen wäre. Dieses Buch eröffnet einen schrittweisen Einstieg in die Welt der Mikrocontrollerprogrammierung und führt mit ausführlichen Anwendungsbeispielen in die Fähigkeiten dieser außergewöhnlichen Prozessorfamilie ein.

296 Seiten (kart.) • ISBN 978-3-89576-236-9  
€ 42,00 • CHF 57,20



Professionelle Kfz-Fehlerdiagnose

## CAN und EOBD in der Fahrzeugtechnik

In diesem Buch werden Funktion und Aufbau des CAN-Protokolls sowohl im Hard- als auch im Software-Bereich ausführlich beschrieben. Dabei kommen die damit verbundenen Themen wie „Aufbau eines CAN-Berichtes“, „Prioritätsregelung“ und „physikalische Verbindung“ nicht zu kurz. Neben dem CAN-Bus ist EOBD-via-CAN ein vorherrschendes Thema. Wie funktioniert EOBD und welche diagnostischen Gegebenheiten sind mithilfe eines Diagnostestesters zugänglich?

275 Seiten (kart.) • ISBN 978-3-89576-242-0  
€ 39,00 • CHF 53,10

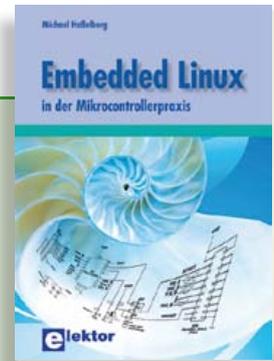


Callbacks, Fließkommafunktionen und BCD-Arithmetik

## Systemprogrammierung II für AVR-Mikrocontroller

Mit jeder neuen Controllergeneration nimmt die Anzahl der bereitgestellten Interrupts zu. Eine Möglichkeit Interrupts zu organisieren, um dieser Zunahme Herr zu werden, nämlich der Einsatz von Callbacks und Callback Chains, wird im ersten Teil des Buches vorgestellt. Es folgt die Ergänzung und Erweiterung der Fließkommaarithmetik aus dem ersten Buch der Reihe „Systemprogrammierung“ mit den wichtigsten mathematischen Funktionen.

319 Seiten (kart.) • ISBN 978-3-89576-243-7  
€ 42,00 • CHF 57,20

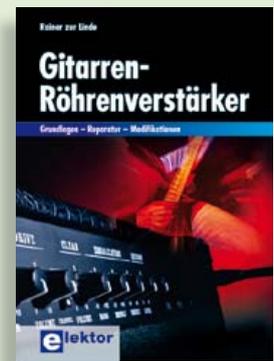


Linux goes Embedded

## Embedded Linux in der Mikrocontrollertechnik

Linux auf dem Desktop – das ist heute leicht und einfach möglich. Embedded Linux ebenso, auch wenn es kompliziert erscheint. Dieses Buch gibt allen Interessierten Hilfestellung, die Linux auf einer Embedded-Plattform installieren und nutzen möchten. Das im Buch vorgestellte System arbeitet mit der Toshiba-ARM9-Familie.

304 Seiten (kart.) • ISBN 978-3-89576-208-6  
€ 44,00 • CHF 59,90



Grundlagen – Reparatur – Modifikationen

## Gitarren-Röhrenverstärker

Obwohl subjektive Qualitätskriterien für „guten Sound“ rational kaum fassbar sind, sind die technischen Zusammenhänge eindeutig. Dieses Buch deckt diese Zusammenhänge in Theorie und Praxis auf. Es richtet sich an den technisch ambitionierten Musiker ebenso wie an denjenigen, der sich mit dieser außerordentlich interessanten Verstärkertechnik näher befassen möchte. Neben unverzichtbaren Funktionsbeschreibungen findet der Leser Anleitungen zur Fehlersuche und Instandsetzungstechnik.

253 Seiten (kart.) • ISBN 978-3-89576-241-3  
€ 42,00 • CHF 57,20

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

[www.elektor.de](http://www.elektor.de)

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](mailto:bestellung@elektor.de)

**e**lektor

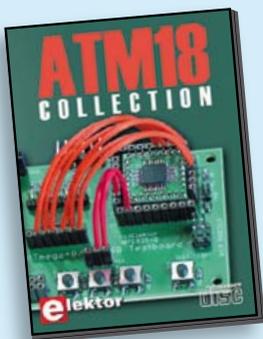


**Kompletter Elektor-Jahrgang 2010 auf DVD**

**Elektor-DVD 2010**

Die neue Elektor-Jahrgangs-DVD enthält alle Artikel des Jahrgangs 2010. Sie verfügt über eine sehr übersichtlich gestaltete Benutzeroberfläche. Mit der Elektor-DVD 2010 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 978-90-5381-267-9 • € 27,50 • CHF 37,40



**Mikrocontroller-Programmierung leicht gemacht**

**ATM18-Collection**

Diese CD-ROM enthält Artikel der populären Elektor-Serie „CC2-AVR-Projekt“ mit mehr als 25 Projekten mit dem ATM18-Board inkl. der benötigten Software und Platinenlayouts sowie weiteren Zusatzinformationen. Des Weiteren umfasst die CD auch den kompletten 6-teiligen Elektor-BASCOM-AVR-Kurs.

ISBN 978-0-905705-92-7 • € 29,50 • CHF 40,20



**Topseller**

**Pico C**

(Elektor April 2011)

HF-Entwickler müssen häufig parasitäre Kapazitäten im Picofarad-Bereich aufspüren. Leider werden auch bessere Digitalmultimeter ziemlich unpräzise, wenn es um das Messen solch kleiner Kapazitäten geht. Gewöhnlich liegt der kleinste Messbereich bei 2.000 pF! Mit dem „Pico C“ klappt das Messen besser, er zeigt die Picofarads sogar mit Nachkommastelle an!

*Bausatz mit allen Bauteilen inkl. 'Elektor Project Case', Platine, progr. Controller und LCD*

Art.-Nr. 100823-71 • € 82,50 • CHF 112,20



**OBD-2-Wireless**

(Elektor April 2011)

Die preiswerteste Lösung für die Autodiagnose ist ein OBD-2-Interface zum Anschluss an einen Notebook/PC mit Diagnosesoftware. Allerdings ist eine solche Kabellösung nicht wirklich praktisch. Ein autonomer OBD-Tester ist hingegen wesentlich teurer und nicht so vielseitig wie eine PC-gestützte Diagnose. Eine interessante Alternative ist dieses kabellose OBD-Interface mit Funkverbindung zum PC (via ZigBee oder Bluetooth).

*(ZigBee-)Bausatz mit allen Bauteilen inkl. Gehäuse + ZigBee-USB-Stick*

Art.-Nr. 100872-71 • € 139,95 • CHF 190,40

*(Bluetooth-)Bausatz mit allen Bauteilen inkl. Gehäuse*

Art.-Nr. 100872-72 • € 139,95 • CHF 190,40



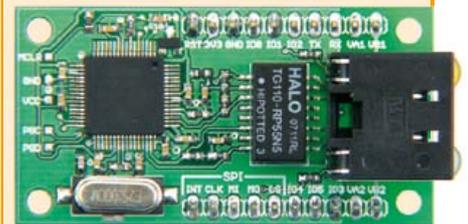
**SatFinder**

(Elektor März 2010)

Wer regelmäßig eine Satellitenschüssel neu ausrichten muss (zum Beispiel, wenn ein Camper oder ein Boot damit ausgestattet ist), steht immer vor dem Problem, die jeweilige Satellitenposition zu ermitteln. Der GPS-basierte SatFinder verfügt über eine Datenbank mit den beliebtesten TV-Satelliten und berechnet mit Hilfe der GPS-Standortdaten alles, was man zum Ausrichten der Antenne auf den Satelliten braucht.

*Bausatz inkl. progr. Controller, Display und Platine (europäische Version)*

Art.-Nr. 100699-71 • € 79,95 • CHF 108,80



**Netzer**

(Elektor Dezember 2010)

Diese Schaltung ist rund um einen PIC 18-Controller aufgebaut, der mit einem integrierten Ethernet-Transceiver ausgestattet ist. Die frei erhältliche C-Software-bibliothek zur Implementierung des TCP/IP-Stacks wurde noch um einige Funktionen erweitert. Außerdem gehört zu diesem Projekt noch ein µC-Webserver, der Zustände an den Pins signalisieren und ändern kann, womit sich über das Internet schalten und walten lässt. Darüber hinaus lässt sich das Modul aber auch als „NetzwerkModem“ für einen anderen Controller nutzen. Für fortgeschrittene User gibt's SPI-Kommunikation, PoE und mehr.

*Bestückte und getestete Platine*

Art.-Nr. 100552-91 • € 59,95 • CHF 81,60

Juni 2011 (Nr. 486) €

+++ Das Lieferprogramm zu dieser Ausgabe finden Sie auf [www.elektor.de](http://www.elektor.de) +++

Mai 2011 (Nr. 485)

**Fernbedienung per Handy**

- 071035-72 ..... Bausatz mit Relais-Platine  
inkl. allen Bauteilen und Relais ..... 49,95
- 071035-91 ..... SMD-bestückte Platine mit allen Bauteilen  
(ATM18-Controllermodul) ..... 12,50
- 071035-92 ..... SMD-bestückte Platine mit allen Bauteilen  
(ATM18-Testboard) ..... 37,50
- 071035-93 ..... SMD-bestücktes LCD-Board mit allen Bauteilen  
und Stiftleisten inkl. 4x20-LCD ..... 29,95
- 071035-95 ..... SMD-bestückte Porterweiterungs-Platine ..... 16,95

**Elektor Proton Robot**

- 110263-71 ..... Komplettbausatz  
(Rumpf + Kopf + Audio + Greifer + PIC-Controller) ..... 1249,00
- 110263-72 ..... Komplettbausatz  
(Rumpf + Kopf + Audio + Greifer + AVR-Controller) ..... 1249,00
- 110263-91 ..... Komplett aufgebauter und getesteter Roboter  
(mit PIC-Controller) ..... 1699,00
- 110263-92 ..... Komplett aufgebauter und getesteter Roboter  
(mit AVR-Controller) ..... 1699,00
- 110263-78 ..... PIC-Board (bestückte und getestete Platine) ..... 39,00
- 110263-79 ..... AVR-Board (bestückte und getestete Platine) ..... 39,00

**Bessere Verständlichkeit bei Online-Konferenzen**

- 100465-1 ..... Platine ..... 9,95
- DMX512-Dimmer**
- EB006 ..... E-blocks PIC-Multiprogrammer ..... 115,95
  - TEFLCST4 ..... Flowcode 4 für PIC (Home/Studenten-Version) ..... 61,50

April 2011 (Nr. 484)

**Pico C**

- 100823-1 ..... Platine ..... 9,95
- 100823-41 ..... Progr. Controller ATtiny2313-20PU ..... 9,95
- 100823-71 ..... Bausatz mit allen Bauteilen inkl. 'Elektor Project Case',  
Platine, progr. Controller und LCD ..... 82,50

**OBD-2-Wireless**

- 100872-71 ..... (ZigBee-)Bausatz mit allen Bauteilen  
inkl. Gehäuse + ZigBee-USB-Stick ..... 139,95
- 100872-72 ..... (Bluetooth-)Bausatz mit allen Bauteilen  
inkl. Gehäuse ..... 139,95

**GPIB/USB-Konverter**

- 080068-91 ..... Bestückte und getestete Controller-Platine ..... 72,00

**Gitarre für Multi-Effektgerät**

- 100923-1 ..... Platine ..... 7,95

**RS-485-Kommunikation mit ATM18**

- 071035-72 ..... Bausatz mit Relais-Platine  
inkl. allen Bauteilen und Relais ..... 49,95
- 071035-91 ..... SMD-bestückte Platine mit allen Bauteilen  
(ATM18-Controllermodul) ..... 12,50
- 071035-92 ..... SMD-bestückte Platine mit allen Bauteilen  
(ATM18-Testboard) ..... 37,50
- 071035-93 ..... SMD-bestücktes LCD-Board mit allen Bauteilen  
und Stiftleisten inkl. 4x20-LCD ..... 29,95
- 071035-95 ..... SMD-bestückte Porterweiterungs-Platine ..... 16,95
- 080213-71 ..... USB/TTL-Konverterkabel 5 V ..... 24,50

**Asteroids & E-blocks**

- EB014 ..... Tastatur ..... 23,45
- EB058 ..... Grafisches LCD ..... 93,00
- EB655S14 ..... dsPIC/PIC24-Bundle (Starterkit Professional) ..... 386,75
- TEDSSI4 ..... Flowcode 4 für dsPIC/PIC24 (Professional-Version) ..... 234,50

**Höhenmesser für Modellraketen**

- 100418-41 ..... Progr. Controller PIC16F88-E/SO (SOIC-18) ..... 9,95

März 2011 (Nr. 483)

**SatFinder**

- 100699-1 ..... Platine ..... 12,95
- 100699-41 ..... Progr. Controller ATMEGA8A-PU (europäische Version) ... 9,95
- 100699-71 ..... Bausatz inkl. progr. Controller, Display und Platine  
(europäische Version) ..... 79,95

# Bestseller

<b>Bücher</b>	1 <b>Basiskurs BASCOM-AVR</b> ISBN 978-3-89576-238-3 ..... € 39,80 ..... CHF 54,20
	2 <b>RFID</b> ISBN 978-3-89576-219-2 ..... € 56,00 ..... CHF 76,20
	3 <b>Embedded Linux in der Mikrocontrollertechnik</b> ISBN 978-3-89576-208-6 ..... € 44,00 ..... CHF 59,90
	4 <b>Das MSP430 Mikrocontroller Buch</b> ISBN 978-3-89576-236-9 ..... € 42,00 ..... CHF 57,20
	5 <b>Systemprogrammierung II für AVR-Mikrocontroller</b> ISBN 978-3-89576-243-7 ..... € 42,00 ..... CHF 57,20
<b>CD- &amp; DVD-ROMs</b>	1 <b>ECD 6</b> ISBN 978-90-5381-258-7 ..... € 29,50 ..... CHF 40,20
	2 <b>ATM18-Collection</b> ISBN 978-0-905705-92-7 ..... € 29,50 ..... CHF 40,20
	3 <b>Elektor-DVD 2010</b> ISBN 978-90-5381-267-9 ..... € 27,50 ..... CHF 37,40
	4 <b>Wireless-Toolbox</b> ISBN 978-90-5381-268-6 ..... € 32,50 ..... CHF 44,20
	5 <b>The Audio Collection 3</b> ISBN 978-90-5381-263-1 ..... € 21,50 ..... CHF 29,30
<b>Bausätze &amp; Module</b>	1 <b>Pico C</b> Art.-Nr. 100823-71 ..... € 82,50 ... CHF 112,20
	2 <b>SatFinder</b> Art.-Nr. 100699-71 ..... € 79,95 ... CHF 108,80
	3 <b>OBD-2-Wireless (Bluetooth)</b> Art.-Nr. 100872-72 ..... € 139,95 ... CHF 190,40
	4 <b>Bluetooth für OBD-2-Analyser NG</b> Art.-Nr. 090918-71 ..... € 29,95 ..... CHF 40,80
	5 <b>Netzer</b> Art.-Nr. 100552-91 ..... € 59,95 ..... CHF 81,60

Bestellen Sie jetzt einfach und bequem online unter [www.elektor.de/shop](http://www.elektor.de/shop) oder mit der portofreien Bestellkarte am Heftende!

**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](mailto:bestellung@elektor.de)



# Halbleiterheft 2011

## Die große Sammlung von Schaltungen, Ideen und Tipps

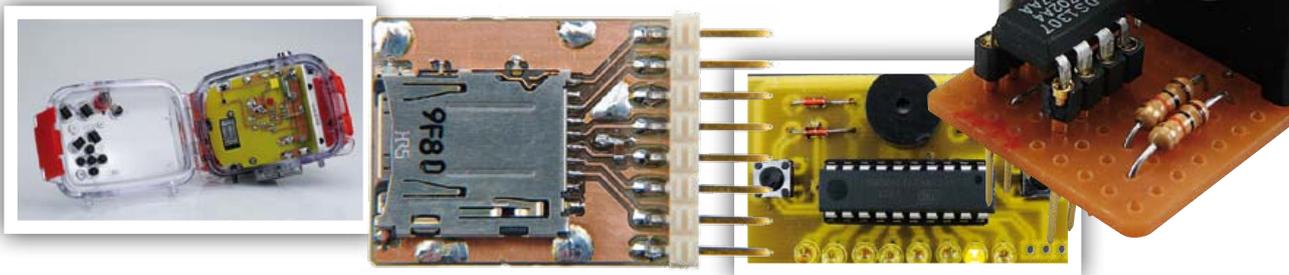
Unsere nächste Ausgabe ist extrastark: Das bekannte Sommer-Doppelheft Juli/August ist eine einzigartige Inspirationsquelle für Elektronik. Die Redakteure und Entwickler des Elektor-Labors haben wieder eine Vielzahl interessanter Schaltungen gesammelt. Darüber hinaus sind Anwendungen neuer ICs, Schaltungstricks sowie Software- und Entwicklungstipps für Mikrocontroller zahlreich vertreten. Diese besonders gehaltvolle Elektor-Ausgabe gibt es nur einmal im Jahr - also nicht verpassen!

### Vorgeschmack auf den Inhalt:

- |                          |                             |                 |
|--------------------------|-----------------------------|-----------------|
| Bewegungssensor          | Backup-Stromversorgung      | Ringoszillator  |
| Dip-Meter                | Multiflasher                | DC/DC-Konverter |
| Tandem-Türglocke         | Universeller Bauteiletester | Schwebende      |
| Chaos-Generator          | Wasserdetektor              | Stromversorgung |
| Audion                   | Elektronische Hundeflöte    |                 |
| Serieller AVR-Programmer | Lauflicht                   |                 |

### Extra im Halbleiterheft:

Mittlerweile schon Tradition hat das große Halbleiterheft-Projekt. Dieser Artikel erhält gerade seinen letzten Schliff – mehr wird noch nicht verraten. Lassen Sie sich überraschen – in ein paar Wochen wird der Vorhang gelüftet!



Das Elektor-Halbleiterheft Juli/August 2011 erscheint am 22. Juni 2011!

Elektor gibt es im Bahnhofsbuchhandel, Elektronik-Fachhandel, an ausgewählten Kiosken und garantiert beim Presse-Fachhändler. Sie können Elektor auch direkt bei [www.elektor.de](http://www.elektor.de) bestellen.

Änderungen vorbehalten!



**Abo-Service:** Marleen Brouwer  
E-Mail: [abo@elektor.de](mailto:abo@elektor.de)

**Bestellannahme und Bestellservice:** Peter Custers  
E-Mail: [bestellung@elektor.de](mailto:bestellung@elektor.de)  
Tel. +49 241 88 909-66

#### Geschäftszeiten

Montag – Donnerstag von 08:30 bis 17:00 Uhr  
Freitag von 08:30 bis 12:30 Uhr  
Tel. +49 241 88 909-0  
Fax +49 241 88 909-77

Unser Kundenservice berät Sie bei allen Fragen zu Bestellungen, Lieferterminen und Abonnements. Änderungen, Reklamationen oder besondere Wünsche (wie z. B. Geschenkabonnement) richten Sie ebenfalls an den Kundenservice. Vergessen Sie bitte nicht, Ihre Kundennummer anzugeben – falls vorhanden.

#### Einzelheft

Deutschland	€ 7,40
Österreich, Belgien, Luxemburg	€ 8,20
Schweiz	CHF 14,30

#### Jahresabonnement-Standard

Deutschland	€ 77,50
Österreich, Belgien, Luxemburg	€ 84,50
Schweiz	€ 97,50
Andere Länder	€ 102,50

#### Jahresabonnement-PLUS

Deutschland	€ 95,00
Österreich, Belgien, Luxemburg	€ 102,00
Schweiz	€ 115,00
Andere Länder	€ 120,00

#### Probeabonnement

Alle Länder (zzgl. Porto)	€ 14,90
---------------------------	---------

#### Studentenabo-Standard

Deutschland	€ 62,00
Österreich	€ 67,60
Schweiz	€ 78,00

#### Studentenabo-PLUS

Deutschland	€ 79,50
Österreich	€ 85,10
Schweiz	€ 95,50

#### Upgrade zum Abo-PLUS

Alle Länder	€ 17,50
-------------	---------

Jahres- und Studentenabonnements (11 Hefte) dauern immer 1 Jahr und verlängern sich automatisch um weitere 12 Monate, wenn nicht spätestens 2 Monate vor Ablauf schriftlich gekündigt wird.

Änderungen und Irrtümer vorbehalten.

#### Bankverbindungen

**Commerzbank Aachen**  
Konto 1 201 102 (BLZ 390 400 13)  
IBAN: DE89 3904 0013 0120 1102 00  
BIC: COBADEFFXXX

**Postgiro Köln**  
Konto 229 744-507 (BLZ 370 100 50)  
IBAN: DE17 3701 0050 0229 7445 07  
BIC: PBNKDEFF



# Elektor-PCB-Service

Die Adresse für Platinen, Prototypen und Multilayer

Möchten Sie Ihre selbst entworfene Platine schnell und zuverlässig geliefert bekommen? In Kleinserie und dabei zu einem unschlagbar günstigen Preis? Bestellen Sie jetzt Ihre individuelle Platine beim Elektor-PCB-Service!



Überzeugen Sie sich selbst vom Elektor-Leiterplatten-Service – jetzt unter

[www.elektorpcbservice.de!](http://www.elektorpcbservice.de!)

Elektor-PCB-Service ist der Leiterplatten-Service von Elektor! Über die Website [www.elektorpcbservice.de](http://www.elektorpcbservice.de) können Sie Ihren Entwurf als professionelle Platine herstellen lassen. Der Elektor-PCB-Service ist die richtige Adresse für Prototypen von neu entwickelten Platinen und für die Produktion modifizierter Elektor-Platinen.

Brauchen Sie kurzfristig einige Muster (Protos) oder eine Kleinserie (Batch), bestehend aus 5 bis 50 Exemplaren? Der Elektor-PCB-Service bietet jetzt beides zu einem günstigen Preis. Sie müssen uns nur über unsere Website Ihr Platinenlayout zusenden.

- Höchste Präzision und Industrie-Qualität zum günstigen Preis
- Kein Mindestbestellwert
- Keine Film- oder Einrichtungskosten
- Keine versteckten Kosten
- Online-Preisrechner
- Versand innerhalb von 5 Werktagen

Hier ist meine Anschrift:

Firma \_\_\_\_\_

Vorname \_\_\_\_\_

Name \_\_\_\_\_

Straße, Nr. \_\_\_\_\_

PLZ, Ort \_\_\_\_\_

Land **DE**

Kunden-Nr. \_\_\_\_\_

E-Mail \_\_\_\_\_

Antwort

**Elektor-Verlag GmbH**  
**Süsterfeldstr. 25**  
**52072 Aachen**

Innerhalb  
Deutschlands  
kein Porto  
nötig!

Hier ist meine Anschrift:

Firma \_\_\_\_\_

Vorname \_\_\_\_\_

Name \_\_\_\_\_

Straße, Nr. \_\_\_\_\_

PLZ, Ort \_\_\_\_\_

Land **DE**

Kunden-Nr. \_\_\_\_\_

E-Mail \_\_\_\_\_

Antwort

**Elektor-Verlag GmbH**  
**Süsterfeldstr. 25**  
**52072 Aachen**

Hier ist meine Anschrift:

Firma \_\_\_\_\_

Vorname \_\_\_\_\_

Name \_\_\_\_\_

Straße, Nr. \_\_\_\_\_

PLZ, Ort \_\_\_\_\_

Land **DE**

Kunden-Nr. \_\_\_\_\_

E-Mail \_\_\_\_\_

Antwort

**Elektor-Verlag GmbH**  
**Süsterfeldstr. 25**  
**52072 Aachen**

Innerhalb  
Deutschlands  
kein Porto  
nötig!

→ **shop.embedded-projects.net**  
HARDWARE FOR YOUR PROJECTS – ONLINESHOP

**ONLINESHOP**  
mit vielen  
**Mikrocontrollerboards  
und Starterkits zum  
Einsteigen mit Demo  
ab € 39,95**  
(inkl. 19 % MwSt., zzgl. Versand)

**Schauen Sie vorbei!**

**AUS UNSEREM PRODUKTANGEBOT:**  
**AVR-Starterkit  
Komplettpaket  
mit USBprog**  
Schnelleinstieg in die  
AVR-Welt mit Anleitung  
und Demo  
Best.-Nr.: 700098



**Aktionspreis  
€ 59,90\***  
inkl. 19 % MwSt.  
zzgl. Versand

\* Nur solange der Vorrat reicht!

Holzbachstraße 4, D-86152 Augsburg  
Tel +49 (0) 821 279599-0  
Fax +49 (0) 821 279599-20  
shop@embedded-projects.net



**embedded projects GmbH**  
HARDWARE FOR PROJECTS

**www.  
elektor.  
de**

**INSERENTENVERZEICHNIS JUNI 2011**

Beta Layout . . . . .	www.beta-layout.com . . . . .	41
Embedded Projects . . . . .	www.embedded-projects.net . . . . .	87
Eurocircuits . . . . .	www.eurocircuits.de . . . . .	9
Good Will Instruments . . . . .	www.gwinstek.com . . . . .	29
Hameg . . . . .	www.hameg.com . . . . .	88
Jackaltac . . . . .	www.jackaltac.com . . . . .	11
LeitOn . . . . .	www.leiton.de . . . . .	49
Markt . . . . .		77
mikroElektronika . . . . .	www.mikroe.com . . . . .	3
NXP Product . . . . .	www.NXP.com/microcontrollers . . . . .	2
PICO Technology . . . . .	www.picotech.com/scope3102 . . . . .	35
Reichelt . . . . .	www.reichelt.de . . . . .	31

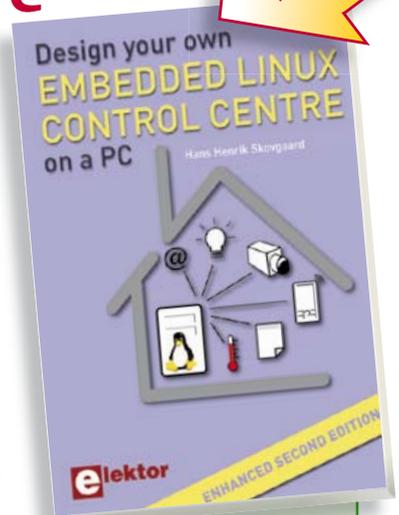
## Design your own Embedded Linux Control Centre on a PC



➔ **Neues Fachbuch in englischer Originalsprache**

Wenn man heute das eigene Heim elektronisch steuern will, gibt es dazu viele Möglichkeiten. Doch dieses Buch ist anders: Es geht eben nicht um X10, ZigBee, Z-Wave oder eine der vielen eingeführten und technisch tollen Systeme, die man heute kaufen kann. Dieses Buch beschreibt ein Do-it-yourself-System aus Recycling-Technik. Warum das? Na, weil die noch funktionsfähige ältere Technik zu schade ist, um im Keller zu verstauben. Außerdem ist dieser Ansatz äußerst preisgünstig.

Das in diesem Buch beschriebene System besteht aus einem wieder zum Leben erweckten alten PC, einer Funksteckdose mit drei Ausgängen plus Sender sowie einer USB-Webcam. Alle Teile sind mit Linux zu einer Einheit verwoben und Linux gibt es bekanntlich kostenlos. Dieses Buch zeigt Ihnen **alle** Grundlagen für die Einrichtung eines Linux-Systems inklusive einer Software-Entwicklungsumgebung, sodass die Hardware als Control Center verwendet werden kann.



412 Seiten (kart.)  
Format 17 x 23,5 cm  
ISBN 978-1-907920-02-8  
€ 39,50 • CHF 53,80

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



ab  
1.148 €

## 70MHz...200MHz 2/4 Kanal Mixed Signal Oszilloskope HM072x | HM0102x | HM0152x | HM0202x

- ✓ 2GSa/s Real Time, rauscharme Flash A/D Wandler (Referenz Klasse)
- ✓ 2MPts Speicher, Memory Zoom bis 50.000:1
- ✓ MS0 (Mixed Signal Opt. H03508) mit 8 Logikkanälen
- ✓ Serielle Busse triggern und hardwareunterstützt dekodieren, I<sup>2</sup>C, SPI, UART/RS-232 (Opt. H0010, H0011)

Kostenfreie Serielle Bus Trigger- & Dekodieroption H0011 über die Analogkanäle bei Bestellungen bis 31.12.2011

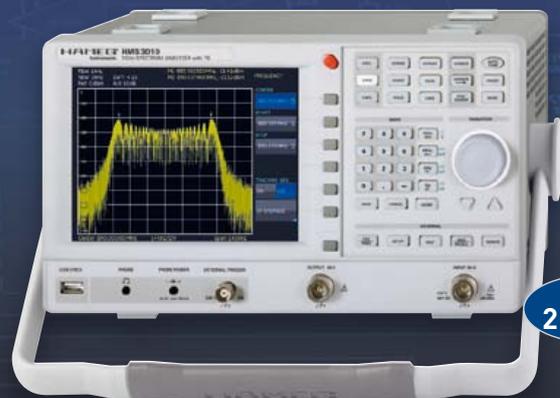
### PROGR. 2 [3] KANAL HOCHLEIST. NETZGERÄT HMP2020 [HMP2030]



ab  
998 €

- ✓ 1x 0...32V/0...10A    1x 0...32V/0...5A    188W max.  
[3x 0...32V/0...5A    188W max.]
- ✓ 188W Ausgangsleistung durch intelligentes Powermanagement
- ✓ Geringe Restwelligkeit: <150µV<sub>eff</sub> durch lineare Nachregelung
- ✓ Hohe Stell- und Rückleseauflösung von 1mV/0,1mA
- ✓ Galvanisch getrennte, erdfreie und kurzschlussfeste Ausgänge
- ✓ Komfortabler Parallel- und Serienbetrieb durch U/I Tracking

### 1GHZ [3GHZ] SPEKTRUMANALYSATOR HMS1000 [HMS3000]



ab  
2.500 €

- ✓ Frequenzbereich 100 kHz...1 GHz [3 GHz]
- ✓ Tracking Generator HMS1010 [HMS3010] -20 dBm/0 dBm
- ✓ Amplitudenmessbereich -114...+20 dBm  
DANL -125 dBm [-135 dBm] mit Preamp. Option H03011
- ✓ Sweepzeit 20 ms...1000 s