

# elektor

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

## Ins Web mit dem Elektor-Linux-Board

- + **Android & Arduino**  
Smartphone mit der realen Welt verbinden
- + **Nixie-VU-Meter**  
Retro-Leistungsanzeige für Röhrenverstärker
- + **ElektorBus-C-Bibliothek**  
Hardware-unabhängige  $\mu$ C-Programmierung

## Kühl Bleiben!



**Effizient & Einstellbar**  
**Ersatz für 7805 und 7905**

# Beta

LAYOUT

## create : electronics



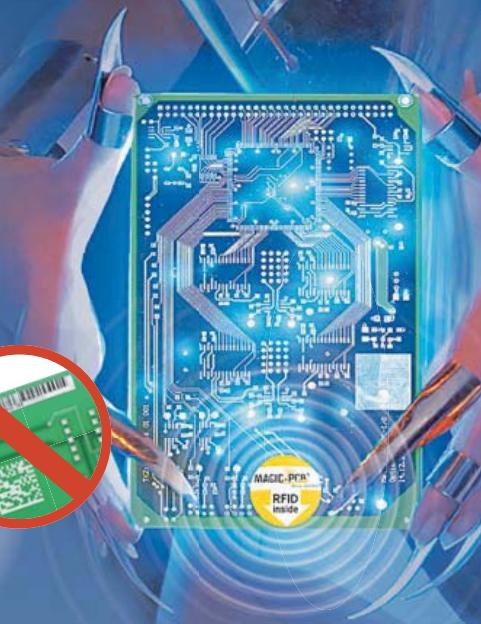
### LIVE Demo auf der electronica:

JEDES Layout-Programm mit IDF output  
kann jetzt 3D, auch EAGLE!



Weltneuheit!

Barcode, DataMatrix... war gestern.  
RFID-embedded PCB -  
jetzt erhältlich!



Entdecke die Chancen:



electronica 2012 | Messe München,  
inside tomorrow | 13.-16. November 2012

Halle C1, Stand 305

[www.beta-layout.com](http://www.beta-layout.com)

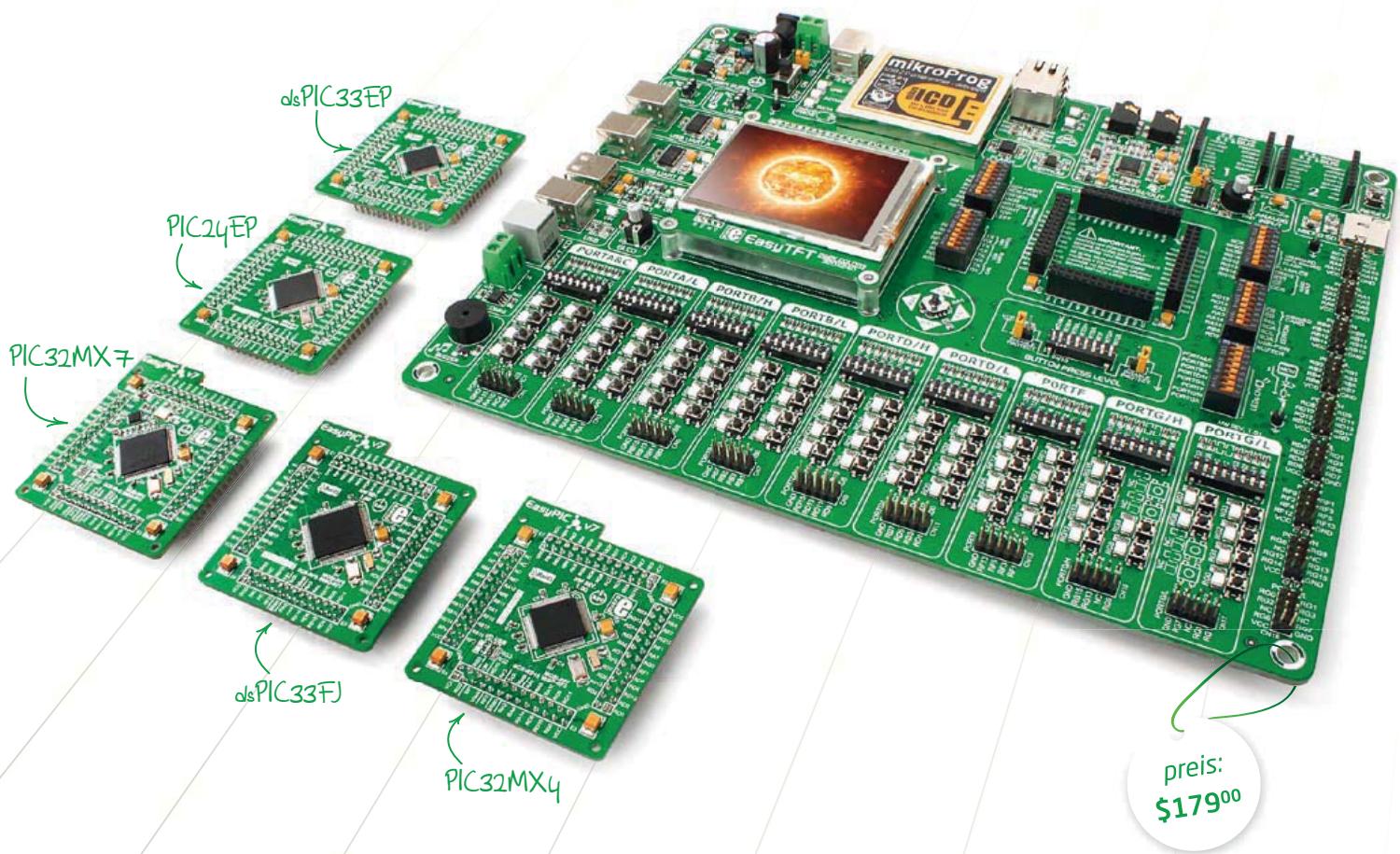
# 16-bit und 32-bit vereint

Intelligenz ist die Fähigkeit zur schnellen Anpassung. Wenn man einfach nahtlos auf einen anderen Mikrocontroller umschwenken kann um die beste Lösung für ein Projekt zu finden, dann ist das schon ein Schlüssel zum Erfolg.

Doch das war nie einfach. Bis heute. **EasyPIC Fusion™ v7** ist das erste Entwicklungssystem der Welt, das drei unterschiedliche Architekturen kombiniert: **dsPIC33®**, **PIC24®** und **PIC32®**. Einfach eine Karte auswählen und loslegen.

Work smart. Not hard.

## EasyPIC Fusion<sup>v7</sup>





## Ihre Wunschliste

Zwischen April und September konnten unsere Leser online die jeweils zehn wichtigsten Artikel der aktuellen Ausgabe bewerten und dazu eigene Wünsche mitteilen. Wir haben Hunderte von Bewertungen und Anregungen bekommen, wofür ich mich an dieser Stelle herzlich bedanken möchte. Die lange, aber natürlicherweise etwas ungeordnete Liste werde ich in den nächsten Tagen aufbereiten und an die Kollegen versenden (nach Redaktionsschluss eines Heftes gibt es manchmal einen Nachmittag Luft für solche Dinge). Sie können sicher sein, dass Ihre Wunschliste auf unsere Auswahl kommender Projekte Einfluss hat. Bei den Artikel-Bewertungen ist mir der überwältigende Zuspruch zu unserem Grundlagen-Kurs aufgefallen. Durchschnittlich rund drei Viertel aller Teilnehmer fanden es interessant, auf unterhaltsame Weise wieder „zurück zu den Wurzeln“ geführt zu werden. Da der Kurs im Dezember enden wird, haben wir mit dem Autor Burkhard Kainka schon eine Fortsetzung angedacht, um in lockerer Folge weitere Grundlagen vermitteln zu können. Sehr gut ist auch unser Embedded-Linux-Kurs angekommen. Und für diese doch etwas komplexere Mikrocontroller- und Software-Materie interessieren sich eine Menge Leute, die sich durchaus auch an einer guten Röhrenschaltung erfreuen können, wie wir nun aus unserer Umfrage wissen. Auf der anderen Seite haben sich viele Leser gewünscht, dass wir wieder mehr einfachere Projekte (gern einmal ohne Controller) veröffentlichen. Auch dies wird in den nächsten „Einsendungs-Besprechungen“ Berücksichtigung finden.

Ab dem nächsten Heft werden Sie eine veränderte Elektor in den Händen halten, sowohl von den Rubriken als auch von der Aufmachung her. Schon jetzt bitte ich auch hier um Feedback (redaktion@elektor.de). Doch das ist noch lange nicht alles, denn es gibt zusätzlich neue Angebote für die Mitglieder unserer Elektronik-Community. Alle zwei Wochen bekommen unsere Mitglieder zum Beispiel einen Extra-Artikel per E-Mail zugeschickt. Mehr dazu finden Sie auf Seite 8!

Jens Nickel

# elektor

### 6 Impressum

Who is who bei Elektor

### 8 Neues von Elektor

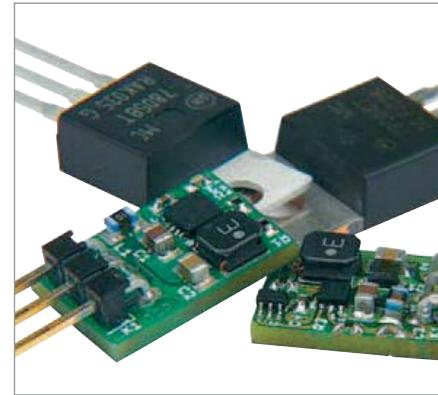
Elektor - Viel mehr als nur eine Zeitschrift!  
Willkommen bei Elektor.LABS!

### 12 Aktuelles

Nachrichten, Leserbriefe, Updates

### 14 Energiebewusste Spannungsregler

Unser 7805- und 7905-Ersatz ist nicht nur effizient, sondern auch einstellbar.



### 20 Embedded Linux leicht gemacht (5)

In dieser Folge des Kurses werfen wir unter anderem den bekannten Webserver „lighttpd“ an.

### 28 Nixie-VU-Meter

Diese Elektor-Schaltung kombiniert ein PSoC (Programmable System on Chip) und IN-9-Nixie-Röhren.

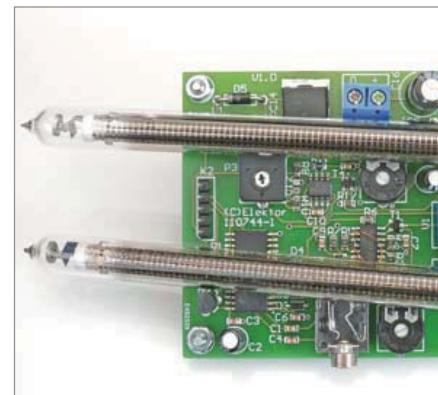


### 34 Eine Lib für den Bus

Zur ElektorBus-C-Bibliothek gibt es natürlich auch Demo-Software: Diesmal steuern wir einen Schrittmotor an.

### 44 Labcenter

Spektrum-Analyser Rigol DSA 815-TG  
Da piept's wohl!

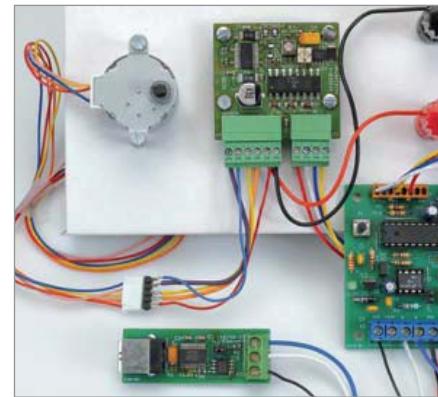


### 46 Bauelemente-Tipps

Manchmal muss man den Energie- oder Leistungsbedarf eines Verbrauchers bestimmen, der am Stromnetz angeschlossen ist. Die Halbleiter-Industrie hat hierfür sogenannte Analog Front Ends (AFEs) auf den Markt gebracht.

### 48 GPIB-nach-USB-Konverter

Der GPIB (General Purpose Interface Bus, bei Hewlett Packard HPIB) ist ein Parallel-Interface nach IEEE488, das für den Anschluss von Sensoren und Messgeräten an einen Computer eingeführt wurde. Wir stellen einen für rund 40 € realisierbaren Wandler vor, mit dem man GPIB-Equipment an den modernen USB eines PCs anschließen kann.



# INHALT

## 14 Energiebewusste Spannungsregler Schaltender Ersatz für 7805 und 7905

Lineare Spannungsregler sind in der Elektronik fast allgegenwärtig. Leider gehen die betagten Klassiker wie der 7805 und 7905 nicht so ökonomisch mit der teuren elektrischen Energie um, wie es heute geboten erscheint. Sparsamer und umweltschonender arbeiten da Schaltregler, doch das Design erfordert mehr Know-how beim Entwickler und oft auch mehr Platz auf der Platine. Hier ist eine Lösung: Ein Schaltregler für eine positive Ausgangsspannung, der die Bauform und Anschlussbelegung eines 7805 hat und sich genauso einfach verwenden lässt.

## 20 Embedded Linux leicht gemacht (5) I/O, ADC, PWM, LAN & Webserver

Das Elektor-Linux-Board bringt eine Vielzahl von Möglichkeiten für die Anwendungsentwicklung mit. In dieser Folge werden wir digitale und analoge Signale einlesen und ausgeben. Dazu kommen eine Netzwerkverbindung und ein kleiner Webserver, der HTML-Seiten dynamisch generiert. Damit kann man sich den Status von LEDs und vielem mehr in einem Browser anzeigen lassen, und auch das Schalten und Walten aus der Ferne ist möglich!

## 28 Nixie-VU-Meter Retro-Leistungsanzeige für Röhrenverstärker

Diese Schaltung ist die Krönung eines (Röhren-)Verstärkers. Mit ihr kommt man zu einer angenehm leuchtenden, kalibrierten Anzeige für Leistung oder Signalpegel. Die Intelligenz der Schaltung steckt in einem PSoC aus Kalifornien, das zwei Nixie-Röhren aus der Ukraine ansteuert.

## 34 Eine Lib für den Bus Schnell zur eigenen ElektorBus-Firmware

Vor genau einem Jahr haben wir Software vorgestellt, mit der man eine PC-Steuerung für die eigene Bus-Anwendung schnell und bequem verwirklichen konnte. Das „Rapid-Development“-Konzept weiten wir nun auf die Firmware der Mikrocontroller-Knoten aus. Dank der ElektorBus-C-Bibliothek müssen sich Entwickler nicht mehr mit den einzelnen Bits der Bus-Protokolle herumschlagen. Die Bibliothek ist dank eines eigenen Hardware-Layers sogar unabhängig vom benutzten Controller. Das eröffnet interessante Perspektiven, die noch über den Bus hinausgehen.

**43. Jahrgang**  
**November 2012**  
**Nr. 503**

- 54 SDN – Software Defined NIC**  
Fertige Netzwerkanbindungen für Mikrocontroller sind nicht ganz billig, die Einarbeitung zeitraubend und die Programmierung aufwändig. Jedoch: Um Netzwerkpakete von einem Controller aus auf die Reise zu schicken, benötigt man eigentlich nur wenige zusätzliche, einfache Bauteile.
- 62 Gleichrichter mit niedrigem Leckstrom**  
Ein Nanopower-Opamp, ein p-Kanal-MOSFET mit niedriger Schwellspannung und zwei Widerstände sind die Ingredienzien eines Gleichrichters mit einer niedrigeren Schleusenspannung als bei einer Diode.
- 64 Arduino auf Kurs (3a)**  
Mit Android und Arduino ist es für jeden möglich, Hardware-Erweiterungen für Smartphones und Tablets zu bauen. Der dritte Teil unserer Serie bietet eine Anleitung, wie man solche Erweiterungen und passende Apps entwickeln kann.
- 72 Zurück zu den Wurzeln (9)**  
Wenn man immer nur bereits vorhandene Signale verarbeiten könnte, wäre die Elektronik um einen Aspekt ärmer: Nämlich das Erzeugen von Schwingungen.
- 76 USB-Wetterlogger mit DCF-Modul**  
Der Datenlogger aus September 2011 zeichnet die von I2C-Sensoren gelieferten Daten für Luftdruck, Temperatur und Feuchte auf. Hier kommt eine erweiterte Version mit einem DCF-Modul.
- 78 Retronik**  
HP-35: Revolution in der Tasche (1972)
- 82 Hexadoku**  
Sudoku für Elektroniker
- 86 Vorschau**  
Nächsten Monat in Elektor
- Aus technischen Gründen mussten wir den in der letzten Vorschau angekündigten Artikel „FPGA-Board“ auf die nächste Ausgabe verschieben.

## Unser Team

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

## Unser Netzwerk



## Internationale Teams

<b>Großbritannien</b> Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	<b>Spanien</b> Eduardo Corral +34 91 101 93 95 e.corral@elektor.es	<b>Indien</b> Sunil D. Malekar +91 9833168815 ts@elektor.in
<b>USA</b> Hugo Vanhaecke +1 860-875-2199 h.vanhaecke@elektor.com	<b>Italien</b> Maurizio del Corso +39 66504755 m.delcorso@inware.it	<b>Russland</b> Nataliya Melnikova +7 (965) 395 33 36 Elektor.Russia@gmail.com
<b>Deutschland</b> Ferdinand te Walvaart +49 241 88 909-0 f.tewalvaart@elektor.de	<b>Schweden</b> Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	<b>Türkei</b> Zeynep Köksal +90 532 277 48 26 zkoksal@betti.com.tr
<b>Frankreich</b> Denis Meyer +33 46 4389435 d.meyer@elektor.fr	<b>Brasilien</b> João Martins +55 11 4195 0363 joao.martins@editorialbolina.com	<b>Südafrika</b> Johan Dijk +27 78 2330 694 / +31 6 109 31 926 J.Dijk@elektor.com
<b>Niederlande</b> Harry Baggen +31 46 4389429 h.baggen@elektor.nl	<b>Portugal</b> João Martins +351 21413-1600 joao.martins@editorialbolina.com	<b>China</b> Cees Baay +86 21 6445 2811 CeesBaay@gmail.com

### IMPRESSUM

43. Jahrgang, Nr. 503 November 2012  
Erscheinungsweise: 11 x jährlich (inkl. Doppelheft Juli/August)

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

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

Anzeigen (verantwortlich): Irmgard Ditgens  
ID Medienservice  
Tel. 05 11/61 65 95-0 - Fax 05 11/61 65 95-55  
E-Mail: service@id-medienservice.de  
Es gilt die Anzeigenpreisliste Nr. 42 ab 01.01.2012

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

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

## Die Elektor Community

264660

Mitglieder in

83

Ländern...

### Noch kein Mitglied?

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



#### 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)!



## Unsere Partner und Sponsoren



ams AG  
[www.ams.com](http://www.ams.com) . . . . . 27



AudioXpress  
[www.audioamateur.com](http://www.audioamateur.com) . . . . . 83



Batronix  
[www.batronix.com](http://www.batronix.com) . . . . . 13



Beta Layout  
[www.beta-layout.com](http://www.beta-layout.com) . . . . . 2



Cadsoft  
[www.element14.com](http://www.element14.com) . . . . . 19



Circuit Design  
[www.circuitdesign.de](http://www.circuitdesign.de) . . . . . 57



DesignSpark  
[www.designspark.com](http://www.designspark.com) . . . . . 67



ELV  
[www.elv.de](http://www.elv.de) . . . . . 42, 43



Eurocircuits  
[www.elektorpcbservice.com](http://www.elektorpcbservice.com) . . . . . 47



FTDI Chip  
[www.ftdichip.com](http://www.ftdichip.com) . . . . . 33



Jackaltac  
[www.jackaltac.com](http://www.jackaltac.com) . . . . . 83



LeitOn  
[www.leiton.de](http://www.leiton.de) . . . . . 57



Linx Technologies  
[www.linxtechnologies.com](http://www.linxtechnologies.com) . . . . . 63



MikroElektronika  
[www.mikroe.com](http://www.mikroe.com) . . . . . 3



National Instruments  
[www.ni.com/multisim/d](http://www.ni.com/multisim/d) . . . . . 61



reichelt  
[www.reichelt.de](http://www.reichelt.de) . . . . . 88

### Sie möchten Partner werden?

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

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

und der elektrischen Sicherheit sind unbedingt zu beachten. Eine Haftung des Herausgebers für die Richtigkeit und Brauchbarkeit der veröffentlichten Schaltungen und sonstigen Anordnungen sowie für die Richtigkeit des technischen Inhalts der veröffentlichten Aufsätze und sonstigen Beiträge ist ausgeschlossen.



Elektor macht mobil für die Zukunft. Ab dem kommenden Monat wird es einige Neuerungen geben. Das Fundament bleibt das, was in den zurückliegenden Jahrzehnten gewachsen ist. Elektor entwickelt sich Schritt für Schritt zu einem weltweiten Portal, zur Plattform und zum Forum für angewandte Elektronik. Und das über Ländergrenzen und Sprachbarrieren hinweg, auf (fast) allen Kommunikationswegen, die heute offen stehen.

Die internationale Elektor-Redaktion:

Harry Baggen (Niederlande), Jan Buiting (UK und USA), Denis Meyer (Frankreich) und Jens Nickel (Deutschland)

Elektor ist viel mehr als die Zeitschrift, die vor Ihnen liegt. Schon vor über 40 Jahren, als alles begann, war Elektor nicht nur ein gedrucktes Fachblatt. Von Anfang an wurden unsere Leser durch einen Platinen-Service und einen technischen Leserdienst unterstützt. In den letzten beiden Jahrzehnten gab das Internet dem Entstehen einer internationalen Elektor-Community starken Auftrieb. Aus Lesern werden Community-Mitglieder, heute auch *Member* oder *Follow-er* genannt. Vom Newcomer bis zum Profi ist dort alles vertreten, gegenseitige Information und Hilfe wird großgeschrieben.

Die Zeichen der Zeit stehen auf Expansion dessen, was Elektor seiner Community bieten kann. Wahrscheinlich durften wir Sie schon einmal auf unserer Website begrüßen, vielleicht haben Sie in unserem Shop eine Platine, ein Modul oder ein Kit bestellt. Elektor-Bücher stehen vermutlich in Ihrem Regal, und es ist gut möglich, dass Sie an einem Elektor-Webinar oder an einem Elektor-Contest teilgenommen haben. Vielleicht hat Elektor sogar Beiträge, Projekte oder Ideen veröffentlicht, deren Urheber Sie sind.

Auch die Industrie hat längst erkannt, dass in Elektor ein hohes Potential steckt. Die Zahl der Kontakte wächst stetig, der intensive Erfahrungsaustausch, die Platzierung von Veröffentlichungen oder auch Beta-Tests von Produkten sind nur einige Themen. Auch Stellenangebote von Personalleitern, die händeringend hochqualifizierte Mitarbeiter suchen, gehen bei uns ein.

An einer ersten Berührung mit Elektor ist heute weniger die Printausgabe beteiligt, unsere Online-Präsenz und das dort verfügbare Leistungsangebot haben dem gedruckten Medium nach und nach den Rang abgelaufen. Der Online-Community gehören heute mehr als 300.000 Mitglieder in über 80 Ländern der Erde an. Die Abrufstatistik weist aus, dass die Elektor-Website von Surfern aus 243 Ländern besucht wird.

Weil Elektor heute eine internationale Gemeinschaft von Menschen ist, die sich rund um den Globus über das Thema Elektronik austauschen, nennen wir diese Menschen nicht mehr Leser oder Abonnenten, sie sind zu Mitgliedern oder *Members* geworden. Wir wollen

der globalen Vernetzung der Mitglieder ebenso wie der Vielfalt der Elektor-Angebote und -Dienste Rechnung tragen, indem wir unsere Online-Präsenz umstrukturieren und zukünftig ELEKTOR.COMMUNITY nennen.

## elektor<sup>e</sup>labs

Unter diesem Label können Sie die Phasen und den Werdegang der Projekte verfolgen. Vertreten sind dort auch bereits entwickelte und getestete Projekte. Auch Ihre eigenen Aktivitäten können Sie dort vorstellen. Elektor.LABS ist bereits in einer Beta-Version unter dem Link [www.elektor-projects.com](http://www.elektor-projects.com) online.

## elektor<sup>e</sup>community

Die zentrale Website, die Homepage des Portals, heißt Elektor. COMMUNITY. Von hier sind der Shop Elektor.STORE, das Archiv Elektor.FILES und die Foren der Elektor-Community nur einen Mausklick entfernt. Zum Aufspüren erschienener Artikel, zum Bestellen im Elektor-Shop oder zum Verlängern der Mitgliedschaft ist Elektor. COMMUNITY die richtige Ausgangsbasis. Natürlich sind alle Zweige des Portals auch direkt erreichbar.

## elektor<sup>e</sup>magazine

Unsere Zeitschrift Elektor rangiert zukünftig unter Elektor. MAGAZINE. Die Zeitschrift erscheint zehn Mal jährlich. Acht Mal im gewohnten Format, mit neuer Titelgestaltung und neuem Logo, mit neuen Rubriken, aber mit mindestens der vertrauten Vielzahl an Projekten. Hinzu kommt außer der traditionellen Sommer-Doppel-

# Viel mehr als nur eine Zeitschrift!



ausgabe  
eine zweite  
umfangstarke Ausgabe am  
Anfang des Jahres. Alle Ausgaben können Sie  
als Mitglied der Elektor-Community auch als umweltschonende  
PDF-Version aus dem Netz herunterladen.

## elektor post

Der bisherige wöchentliche Newsletter e-Weekly räumt das Feld für Elektor.POST. Dieser Dienst wird noch mehr bieten als sein Vorgänger. Wir werden Sie nicht nur regelmäßig über Neues aus der Welt der Elektronik auf dem Laufenden halten, sondern die Informationen auch mit Filmclips anreichern. Auch Einladungen zu Webinaren und zu Events vor Ort kommen über Elektor.POST ins Haus. Unsere Community-Mitglieder bekommen außerdem etwa jede zweite Woche ein exklusives, unveröffentlichtes Projekt per E-Mail zugesandt!

### Machen Sie mit!

Wir haben uns entschlossen, das konventionelle Zeitschrift-Abonnement durch die neue Mitgliedschaft in der Elektor-Community zu ersetzen. Das bedeutet für Sie:

- Zehn Mal im Jahr erhalten Sie die runderneuerte Zeitschrift Elektor, mit der vertrauten Vielzahl an informellen Beiträgen und praxisbezogenen Projekten, aber auch mit neuen Rubriken und Themenbereichen.
- Zwei Mal im Jahr ist dies eine seiten- und inhaltsstarke Elektor-Doppelausgabe.
- Etwa 25 Mal im Jahr bekommen Sie einen exklusiven redaktio-

nellen Beitrag, zum Beispiel eine unveröffentlichte Projektbeschreibung, zugesandt. Wenn Sie diese Beiträge ausdrucken und archivieren wollen, können Sie bei uns einen kostenlosen Sammelordner anfordern.

- Im Elektor-Shop räumen wir Ihnen über Elektor.STORE auf alle Produkte und Leistungen mindestens 10 % Rabatt ein.
- Platinen sind zukünftig für sämtliche Projekte in Elektor (mit Ausnahme der „Leserprojekte“) verfügbar, für Mitglieder der Elektor-Community mit mindestens 10 % Nachlass.
- Sie erhalten unmittelbaren Zugang zu Elektor.LABS.
- Elektor.POST bringt Ihnen regelmäßig News und Infos aus der Welt der Elektronik in Ihre Mailbox.
- Sie erhalten (jährlich) eine GOLDCARD, einen als USB-Karte gestalteten Mitgliedsausweis, auf der als Bonus attraktive „Goodies“ gespeichert sind.
- Auf der USB-Karte ist auch der Inhalt der bisherigen Elektor-Jahrgangs-DVD gespeichert.

Wenn Ihnen die Umwelt Herzenssache ist, könnte unsere GREENCARD für Sie eine Alternative sein. Im Rahmen dieser Mitgliedschaft erhalten Sie alle digitalisierbaren Produkte und Leistungen (auch das Elektor.MAGAZINE) ausschließlich online.

Wenn Ihr bisheriges Abo ausläuft, erhalten Sie einen persönlichen Brief, in dem diese beiden attraktiven Angebote noch einmal ausführlich dargestellt werden.

# Willkommen bei elektor labs !

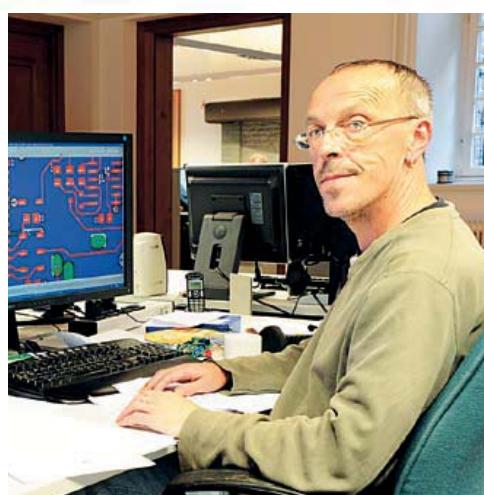
Hallo! Schön, dass Sie die Elektor e-LABS besichtigen wollen. Bis Ende dieses Jahres ist .LABS für jeden offen. Wenn Sie später noch einmal reinschauen möchten, dann brauchen Sie dazu eine Mitgliedskarte oder aber eine Einladung. Unter den Kollegen nennt man mich CPV, und ich führe Sie nun durch dieses neue Labor. Bitte folgen Sie mir!

Elektor e-LABS ist der Ort, an dem man Elektronik in Aktion erleben kann. Vergessen Sie Notizzettel, Dokus und Ähnliches. Ab jetzt ist .LABS - gesprochen „Dot Labs“ - das Zentrum einer lebendigen Elektronik-Community, die auch unter dem Namen Elektor bekannt ist.

In .LABS tummeln sich ganz besondere Menschen: Sie kennen sich nämlich sehr



ist ein wandelndes Lexikon und hat massig Erfahrung. Wenn Sie also eine Audio-Frage haben, dann fragen Sie am besten Ton. Ich mache das immer so.



gut in vielen Winkeln des Elektronik-Universums aus. Hier sehen Sie zum Beispiel Ton G. Da er so bescheiden ist, sage ich das: Ton ist ein hoch talentierter Entwickler von Audioverstärkern. Er

Gegensatz zu Ton und Luc ununterbrochen von den allerneuesten Technologien schwärmt und die frischen Chips schon kennt, wenn sie noch warm aus dem Ätzbad kommen. Er ist immer über alles informiert. Suchen Sie ein bestimmtes ABCXYZ? Dann fragen Sie Ray.

Weiter geht es mit Luc. Bei Elektor nennen wir ihn Lucky Luc, denn irgendwie kann es ja nur pures Glück sein, dass ausgerechnet seine Projekte immer prächtig funktionieren, oder? Fragen Sie mich nicht, wie er das macht. Es glückt ihm eben. Luc ist ein ruhiger Charakter, und stille Wasser sind tief. Wenn Sie sich mit Luc austauschen, dann werden Sie von seinem schon fast unheimlich fundierten Wissen beeindruckt sein.

Nun zu Raymond, den wir neckisch Raving Ray nennen. Der Grund dafür ist, dass er im





Sie müssen unbedingt auch noch Sunil, Sunil und Sunil aus Indien kennenlernen. Dieses indische Trio kocht Projekte mit einer Prise Curry. Und Elektronik mit Pepp ist immer interessant. Mit Zugriff auf .LABS können Sie mit einem der drei Sunils Kontakt aufnehmen. Hinterlassen Sie einfach einen entsprechenden Kommentar auf der Webseite.

Elektor  LABS ist kein obskures Labor, irgendwo tief versteckt im Keller eines Schlosses. Es handelt es nicht einmal um *ein* Lab, sondern um *viele* Labs. Diese findet man überall auf der Welt: In Garagen, Hobbykellern, Firmen, Schulen und Universitäten. Elektor  LABS ist als virtueller Ort nicht geographisch beschränkt, sondern überall da zu finden, wo es Internet gibt. Z.B. bei Ivan aus St. Petersburg, der sich auf Mikrocontroller-Anwendungen für den Sport spezialisiert hat. Und auf anderen Kontinenten geht es weiter: Gina in Peru beschäftigt sich mit „wearable electronics“ und Paikan baut in Japan an Multicoptern. Wenn Sie sich also für Elektronik interessieren, dann sind Sie bei .LABS genau richtig. Vielleicht würden Sie ja gerne bei der Entwicklung eines Windgeschwindigkeitsmessers ohne bewegte Teile mitmachen. Genau für solche Projekte ist Elektor  LABS da. Starten Sie dort ein Projekt und bringen Sie Ihre Ideen und Vorschläge ein. Erstellen Sie eine kurze Beschreibung. Erklären Sie was die Schaltung tun soll, damit die Community Ihre Ziele versteht und mitmachen kann. Sparen Sie nicht an Fotos, Skizzen und Schaltplänen – sie machen ein Projekt erst richtig interessant.

Elektor  LABS wurde dazu konzipiert, Menschen mit ähnlichen Interessen zusammenzubringen. Man kann hier gemeinsam Probleme lösen oder im Team Projekte entwickeln. Kooperativ macht es einfach mehr Spaß, denn hier werden ihre Fähigkeiten gewürdigt. Haben Sie schon bei .LABS reingeschaut? Ein vollelektronisches

Messgerät für die Geschwindigkeit und die Richtung des Windes gibt es da schon. Und noch viele andere interessante Projekte zu allen möglichen Themen.

Wenn man Elektor  LABS besucht,

**Android style capacitive sensing pattern lock**



kann man Projekte nicht nur anschauen, sondern auch bewerten! Man muss lediglich auf einen Stern auf der Projektseite klicken. Diese Bewertungen sind wichtig, denn Elektor respektiert die Einschätzung der Community

[www.elektor-labs.com](http://www.elektor-labs.com)

und wählt gute Projekte für die Veröffentlichung in der Zeitschrift aus. Das hat etliche Vorteile: Das Projekt profitiert vom Engagement der Elektor-Mitarbeiter, und Sie können Ihren Namen als Autor in der Zeitschrift lesen. Außerdem können Sie als Projekt-Starter auch ein Honorar erhalten. Damit können Sie dann weitere Elektronik-Projekte finanzieren oder aber Ihre Mitstreiter zum Essen einladen, ganz wie Sie wollen.

Doch nicht nur Projekte mit fünf Sternen schaffen es in die Zeitschrift. Schließlich mag nicht jeder Lady Gaga oder die Beatles. Manche Elektroniker haben speziellere Vorlieben. Es kann also passieren, dass auch ein Projekt ohne besondere Aufmerksamkeit bei Elektor auf Resonanz stößt und somit in der Zeitschrift erscheint. Deshalb hat jedes Projekt die Chance auf Veröffentlichung. Im Prinzip kann also jeder Autor für sein Engagement belohnt werden. Elektor  LABS ist mehr als pure Elektronik. Hier geht es um alle Facetten von Elektronik und den Spaß dabei.



Es gibt außerdem Wettbewerbe, gelegentlich sogar Elektronik-Samples gratis oder weitere kleine Geschenke. Wie Sie das schon von der Elektor und den zugehörigen Webseiten kennen, veranstalten wir immer wieder besondere Events, bei denen es zum Teil richtig fette Preise abzuräumen gilt. Aktuell läuft etwas beim PSoCarnoph-Projekt:

Wer bei diesem modernisierten analogen Synthesizer mitmachen möchte, hat die Chance, ein kostenloses PSoC5-Entwicklungssystem zu erhalten. Die besten Beiträge werden belohnt. Noch ein wichtiger Hinweis: Dieses Projekt wird vom weltbekannten Filterexperten Kendall Castor-Perry moderiert. Wenn Sie sich für Musikinstrumente und Klangsynthese interessieren, ist das vielleicht genau Ihr Ding.

Damit kommen wir zum Ende der Tour durch die Elektor  LABS. Wie Sie bemerkt haben, geht es bei Elektor  LABS um mehr als um ein Elektronik-Labor. Es geht um Kooperation und Freude an Elektronik. Wenn Sie einen eigenen Account für Elektor  LABS haben, können sie dort stöbern, so oft Sie wollen.

Wenn Sie gerade etwas diskutieren wollen, dann finden Sie dort einen passenden Gesprächspartner, vielleicht auch einen der hier abgebildeten Elektor-Mitarbeiter.

Stellen Sie einfach einen Beitrag zu einem Projekt ein oder hinterlassen Sie einen Kommentar. Wir würden uns freuen!

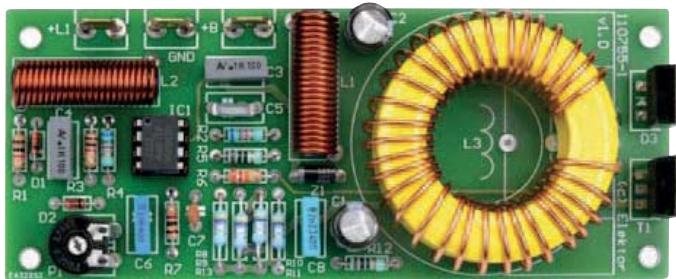
(120608-1)

## Mailbox

### Einspruch eingelegt

„Nicht nachmachen!“, Elektor 09/2012, S. 8 (120434)

In dem Artikel „Nicht nachmachen!“ der Septemberausgabe berichten Sie, dass die Firma Carsig GmbH an Sie herangetreten sei, um Sie über die Existenz eines europäischen Patents zu informieren. Die vorgestellte Lastsimulationsschaltung sei bereits aufgrund dieses bestehenden europäischen Patents geschützt und ein Nachbau für nicht private Zwecke daher unzulässig. Aus meiner Sicht bedarf diese Aussage einer Klarstellung. Ein Blick in die Onlineakte des europäischen Patentamts (<https://register.epo.org/espacenet/application?number=EP09171435&lng=en&tab=doclist>) zeigt, dass das europäische Patent EP 2 173 140 B1 zwar erteilt worden ist, jedoch gegen die Erteilung des Patents innerhalb der neunmonatigen Einspruchsfrist Einspruch eingelegt worden ist. Ein solches Einspruchsverfahren kann - abhängig von der Relevanz der Dokumente, auf die der Einspruch gestützt wird - zu einer Aufrechterhaltung des Patents, zur Einschränkung des Schutzbereichs oder aber zu einem vollständigen Widerruf führen. Es ist daher zum jetzigen Zeitpunkt nicht mit letzter Gewissheit vorhersehbar, ob und gegebenenfalls in welcher konkreten Form ein Patentschutz nach Abschluss des Einspruchsverfahrens bestehen wird.



Wie aus der Onlineakte des europäischen Patentamts zu entnehmen ist, macht die Einsprechende eine fehlende Neuheit sowie einen Mangel an erforderlicher Tätigkeit basierend auf zahlreichen Dokumenten zum Stand der Technik als Widerrufegründe gegenüber dem europäischen Patent geltend und beantragt, das europäische Patent in vollem Umfang zu widerrufen.

Es bleibt damit zumindest bis zum Abschluss des Einspruchsverfahrens ungewiss, ob der Monopolschutz in Form des europäischen Patents tatsächlich Bestand haben wird oder ob die Lastsimulationsschaltung nach dem Abschluss des Einspruchsverfahrens von jedermann frei genutzt werden kann.

Dipl.-Ing. Jan Quellmann  
Patentanwalt

Vielen Dank für Ihren Hinweis! Wie das Einspruchsverfahren in diesem Falle ausgehen wird, wird Sache des europäischen Patentamts sein. Die eigentliche Aussage unseres Artikels wird dadurch in keiner Weise berührt. Eine Recherche, ob für die bei uns veröffentlichten Schaltungen und Projekte ein Patent- oder Gebrauchsmusterschutz besteht, ist für uns und unsere Autoren viel zu aufwendig. Dass es sich um eine rechtlich komplizierte Materie handelt, wird durch Ihre Mail nochmals bestens bestätigt! Wir können nur noch einmal nachdrücklich darauf hinweisen, dass lediglich die private Nutzung der bei uns veröffentlichten Schaltungen frei ist. Für Forderungen und Ansprüche von Dritten, die durch eine (eventuelle) unerlaubte kommerzielle Nutzung entstehen, kann der Elektor Verlag keinerlei Verantwortung übernehmen.

Redaktion Elektor

### Neue Norm

„Stromzähler im Netz“,  
Elektor 10/2012, S. 30  
(110462)

In der Ausgabe  
Oktober 2012  
Ihrer Fachzeitschrift beschreiben  
Sie ein Gateway  
für Stromzähler  
mit S0-Bus. Dieser  
Artikel hat mich  
besonders interessiert,  
da ich gerade



an einem Projekt mit ähnlichen Anforderungen arbeite.  
Sie beziehen sich im 2. Absatz des Kapitels „S0-Interface“ auf die DIN 43864, in welcher einmal der S0-Bus definiert war. Diese DIN 43864 wurde bereits Ende der 1990er Jahre zurückgezogen. Alle heute gültigen Definitionen zum Thema „S0-Bus“ sind seit dieser Zeit in der Europäischen Norm EN 62053-31 zusammengefasst. Die EN 62053-31 ist beispielsweise im Beuth-Verlag unter dem Titel „DIN EN 62053-31; VDE 0418-3-31:1999-04“ in der bis heute gültigen deutschen Fassung aus dem Jahre 1998 (Ausgabedatum April 1999) erhältlich.

Claus Alexander Waßmuth  
Ing.-Büro Waßmuth, Seyweiler

Vielen Dank für Ihren Hinweis!

Redaktion Elektor

### Power-LED-Treiber

Elektor 07-08/2012, S. 56 (120201)

Die Spannungsfestigkeit der Elkos C3 und C4 von 63 V wird nicht benötigt, ein Wert von 16 V reicht gut aus.

### InterZepter

Elektor 06/2010, S. 32 (100174)

Es gibt einen Beschriftungs-Fehler auf der InterZepter-Platine der Version 1.1. Die Labels bei JP2 sind vertauscht. Mit dem Jumper bei den Pins 1 und 2 wird 3,3 V, mit dem Jumper bei den Pins 2 und 3 wird 5 V ausgewählt. Die Platinen der Version 1.0 sind richtig beschriftet.

## Updates und Korrekturen

### Pt100-Simulator

Elektor 07-08/2012, S. 106 (120325)

In der Tabelle ist bei 350 °C ein falscher Norm-Widerstandswert angegeben, der richtige Wert beträgt 229,7 Ω. Die (nominale) Abweichung liegt bei 0,3 %.

### Netzbetrieb für Rauchmelder

Elektor 07-08/2012, S. 98 (120199)

Im Schaltplan ist ein Fehler: Die Drahtbrücke parallel zu R1 muss weg.

Von Phil Knurhahn

## Eine kostengünstige Roboterhand

In den Sandia National Laboratories (Albuquerque, New Mexico) wurde von dem Forscher Curt Salisbury eine neuartige Roboterhand entwickelt, die sich sehr an dem menschlichen Vorbild orientiert. Da sie sehr variabel ist und z.B. auch mit nur vier Fingern auskommen kann, könnte sich so vor allem eine kostengünstige Lösung für diejenigen Menschen ergeben, die eine neue „Greifhand“ brauchen (Bilder: Randy Montoya, Sandia NL). Dabei wird nicht nur die mechanische Struktur der menschlichen Finger übernommen, sondern auch das Hüllmaterial weitgehend elastisch gehalten. Sogar Kartenspiele sind dabei möglich. Heutige Roboterhände kosten oft mehr als 10.000 \$ für jeden Freiheitsgrad der Bewegung (und man braucht für eine funktionierende Hand etwa 12 Freiheitsgrade). Die Sandia-Hand ist nicht nur modular – mit Hilfe von Magneten können verschiedenartige Finger rasch in die eigentliche Hand eingesetzt werden. Man kann aber auch statt der Finger Werkzeuge einsetzen, Schraubenzieher zum Beispiel, oder Kamerahalter. Und die magnetische Verbindung sorgt dafür, dass der Finger abfällt, wenn der Nutzer damit versehentlich gegen eine Wand stößt. Dann wird er wieder aufgesteckt und man kann weiter damit arbeiten. Mehr noch: Wenn diese Panne passiert, kann die Roboterhand mit den verbliebenen Fingern meist sogar den abgefallenen Finger aufheben und selbstständig an der eigenen Hand wieder befestigen. Das Sandia NL hat bei der Entwicklung mit der Stanford Universität zusammen gearbeitet. Auch die Sandia-Hand hat 12 Freiheitsgrade und derzeitige Kostenabschätzungen gehen davon aus, dass bei einer Kleinserien-Produktion jeder der Freiheitsgrade für rund 800 \$ zu fertigen ist – was für die ganze Hand 10.000 \$ bedeutet. Dies wäre eine Kosteneinsparung von bereits rund 90 % – mit der Option, bei Massenproduktion die Kosten weiter zu senken.

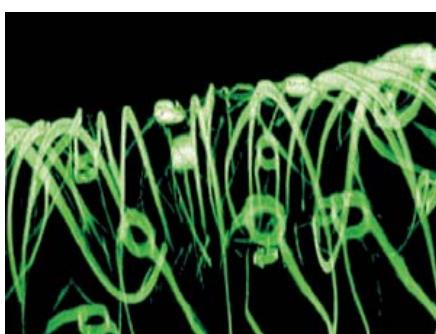
[https://share.sandia.gov/news/resources/news\\_releases/robotic\\_hand](https://share.sandia.gov/news/resources/news_releases/robotic_hand)



## Science Fiction wird Realität

Mit einem Geflecht aus Silizium-Nanodrähten ist es Forschern der Harvard Universität und des benachbarten MIT gelungen, erstmals eine Verbindung zwischen lebenden Zellen und Elektronik herzustellen. In der Online-Ausgabe von „Nature Materials“ wurde beschrieben, wie das geschieht: Dabei werden neue Nerven, aber auch Muskelteile auf einem Nanogerüst aus solchen Silizium-Geflechten generiert (Bild: Charles M. Lieber / Daniel S. Kohane, MIT). Zunächst will man das in Testreihen für neue Medikamente nutzen, aber im Prinzip könnte diese Erfindung zu ganzen Bausteinen weiterentwickelt werden, wie z.B. zu implantierbaren Chips, die Prothesen direkt steuern können. Auch andere Möglichkeiten für die Kommunikation zwischen Nanoelektronik und lebenden Zellen sind denkbar – am Ende vielleicht sogar künstliche Augen.

<http://web.mit.edu/newsoffice/2012/nanoelectronics-and-tissues-0827.html>



## SONDERPREIS AKTION

Exklusiv für die Leser der Elektor und nur bis zum 31.11.2012

### Batronix BX48 Batego II

Weltweit führende Programmietechnik, 5 Jahre Garantie, Made in Germany



nur 470,- \*

### Rigol DS1102E Oszilloskop

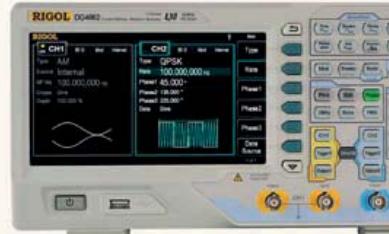
100 MHz, 1 GSa/s, 1 Mpts, FFT, Bestseller, 3 Jahre Garantie



nur 370,- \*

### Rigol DG4062 Arbiträrgenerator

2 Kanäle, 60 MHz, 500 MSa/s, 16 kpts, Bestseller, 3 Jahre Garantie



nur 735,- \*

### Rigol DSA815 Spektrum Analyser

9 kHz bis 1,5 GHz, unschlagbares Preis-Leistungs-Verhältnis, 3 Jahre Garantie



nur 1190,- \*

**electronica**  
Halle A6 Stand 220

Jetzt die Batronix Sonderpreisaktion für Elektor-Leser nutzen und modernste Labortechnik zum Sparpreis bestellen:

[www.batronix.com/go/15](http://www.batronix.com/go/15)

\* Alle Preise bereits inkl. MwSt. und Versand.

# Energiebewusste Spannungsregler

## Schaltender Ersatz für 7805 und 7905

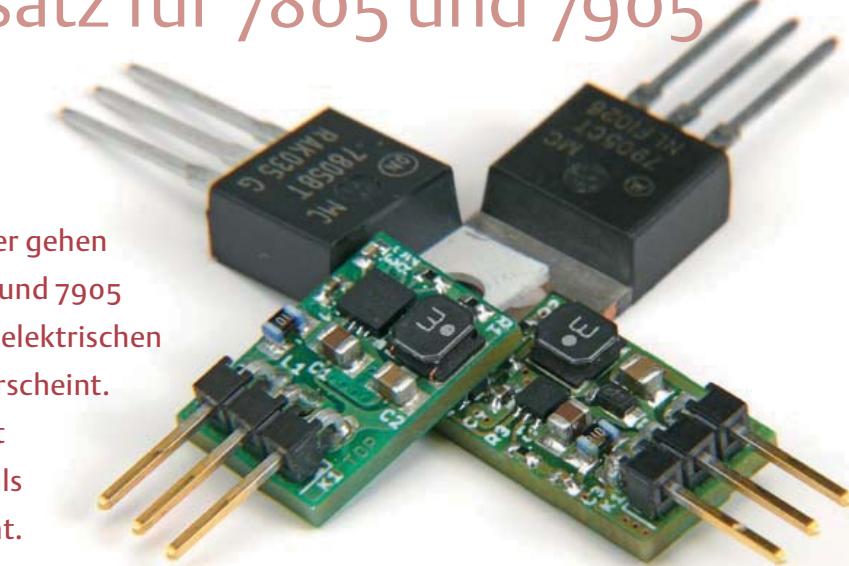
Von Raymond Vermeulen (Elektor-Labor)

Lineare Spannungsregler sind in der Elektronik fast allgegenwärtig. Leider gehen die betagten Klassiker wie der 7805 und 7905 nicht so ökonomisch mit der teuren elektrischen Energie um, wie es heute geboten erscheint.

Das Prinzip der linearen Regelung ist der Grund, dass relativ viel Energie als nutzlose Verlustwärme verloren geht.

Auch wenn die einzelnen Energiemengen

unbedeutend sein mögen, global summieren sie sich zu einem beträchtlichen Posten. Sparsamer und umweltschonender arbeiten Schaltregler, sie erreichen wesentlich höhere Wirkungsgrade.



Ursprünglich war es mein Kollege Ton Giesberts, der vorschlug, für den noch aus den siebziger Jahren des vorigen Jahrhunderts stammenden Spannungsregler 7805 ein schaltendes Äquivalent zu entwickeln. Ich erinnerte ihn daran, dass solche Schaltregler längst auf dem Markt sind. Doch diese Schaltregler, wandte er ein, haben abweichende Bauformen, oder sie können nur niedrige Ströme liefern. Aus der Idee entstand ein Schaltregler für eine positive Ausgangsspannung, ein Abwärtswandler (*Buck Converter*), der die Bauform und Anschlussbelegung eines 7805 hat.

Kurz nachdem das geschafft war, schlug Redakteur Harry Baggen vor, auch eine Variante für negative Ausgangsspannung zu entwickeln. Zwar besaß ich Erfahrungen im Konstruieren von Positiv-nach-Negativ-Wandlern, doch Wandler des Typs „Negativ-nach-weniger-Negativ“ waren für mich Neuland. Bei meinen Recherchen im Internet entdeckte ich einen Schaltungsvorschlag von National Semiconductor, der eine spezielle Art des Abwärtswandlers für negative Spannungen anwendet, *Negative Buck* genannt. Auf dieser Methode beruht

unser Schaltregler für negative Ausgangsspannung, der energiesparende Ersatz für den 7905.

### Positiv-Regler

Der Positiv-Regler arbeitet nach dem Prinzip des Abwärtswandlers, auch *Buck- oder Stepdown Converter* genannt. Wie **Bild 1** zeigt, gehören zum geschalteten Kreis eine Diode, eine Induktivität und ein Kondensator. Wenn der Schalter schließt, wird in der Induktivität Energie gespeichert. Öffnet der Schalter, geht die Energie in den als Puffer dienenden Kondensator und die parallele Last über, der Kreis schließt sich über die Diode. Vom Verhältnis der Einschaltzeit zur Zykluszeit (*Duty Cycle*) hängt die Spannung an der Last ab. Lange Einschaltzeiten haben zur Folge, dass sich die Spannung an der Last der Spannung nähert, die von der Quelle geliefert wird. Bei dieser simplen Konfiguration ist die Ausgangsspannung abhängig von der Größe der Last. Weil dies unerwünscht ist, wird die Schaltung zu einem Regelkreis erweitert. Die Ausgangsspannung wird mit dem Sollwert verglichen, die Spannungsdifferenz steuert die Schaltzeiten des Schalters.

Abwärtsregler, die nach diesem Prinzip arbeiten, werden in elektronischen Systemen vielfältig eingesetzt. Die Halbleiter-Industrie hat integrierte Schaltregler auf den Markt gebracht, die außer dem Schaltelement auch die Regelelektronik sowie Maßnahmen gegen Überlastung und Kurzschluss enthalten. Für unseren Positiv-Regler haben wir den Typ TPS62150 von Texas Instruments gewählt. Dieser synchrone Stepdown-Wandler arbeitet mit der hohen Schaltfrequenz 1,25 MHz, so dass die externen Komponenten mit wenig Volumen auskommen; der entnehmbare Strom beträgt 1 A. Das interne Funktionsschema des TPS62150 ist in **Bild 2** wiedergegeben. In unserem schaltenden 7805-Ersatz, siehe **Bild 3**, ist an die Stelle der Diode aus Bild 1 ein IC-interner MOSFET getreten, was die Verluste der Diode herabsetzt. Abgesehen von der Induktivität sowie einigen Widerständen und Kondensatoren sind die Komponenten auf dem Chip integriert. Vorgeschaltet ist ein LC-Tiefpass (L1/C1), der unvermeidbare Störsignale niedrig hält. Widerstand R3 hilft, eine eventuelle Schwingneigung des Tiefpasses zu unterdrücken. Normalerweise sorgen bereits

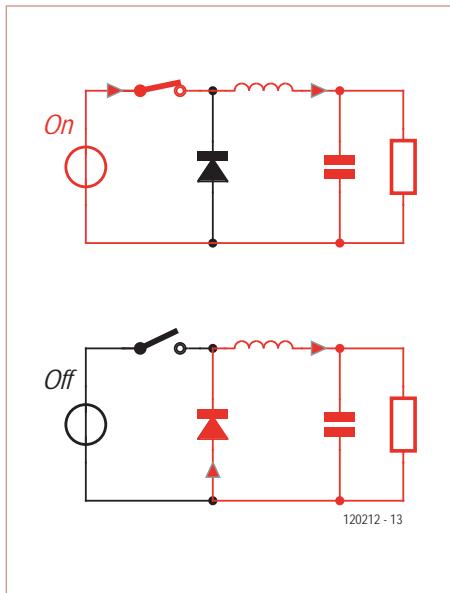


Bild 1. Prinzip eines Step-Down-Schaltreglers.

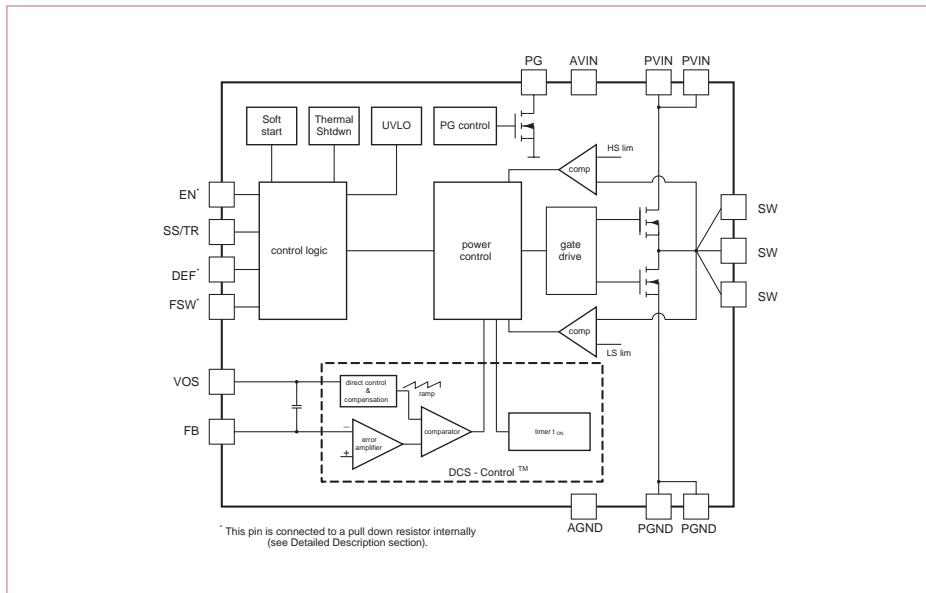


Bild 2. Internes Funktionsschema des TPS62150.

die ESR-Eigenschaften von L1 und C1 dafür, dass der Tiefpass nicht schwingen kann. Wie aus dem Funktionsschema ersichtlich, greifen im TPS62150 mehrere Regelkreise ineinander. Gemeinsam stellen sie die Ausgangsspannung präzise auf den Sollwert ein. Außerdem stellen sie sicher, dass auch größere Lastsprünge nur minimale Spitzen oder Einbrüche auf der Ausgangsspannung zur Folge haben.

Bestimmend für die Wahl der Bauelemente waren die angestrebten Abmessungen des Aufbaus. Die Bauform TO-220 ist nicht streng definiert, das Gehäuse kann abhängig vom Typ und Hersteller in Grenzen variieren. Mit den Abmessungen 15,5 · 10,1 · 4,75 mm haben wir einen Mittelweg gewählt. Dort, wo sich beim originalen 7805 die Kühlfahne befindet, hat die Platine ihren Platz. Die Platine kann nicht mit einem Kühlkörper verschraubt werden, wegen der geringen Wärmeentwicklung unseres Schaltreglers ist dies aber auch nicht notwendig. In **Tabelle 1** sind die Werte der Widerstände R1 und R2 für Ausgangsspannungen angegeben, die von +5 V abweichen.

**Bild 4** zeigt die Platine für den Positiv-Regler. Das Layout eines Schaltreglers bedarf einiger Überlegung, wobei folgende allgemeine Faustregeln gelten: Wo Ströme stark variieren, müssen die Stromwege kurz sein, und wo starke Spannungsänderungen auftreten, ist die Fläche des Kupfers klein zu halten. Da der Positiv-Regler die Abmessungen eines 7805 hat, war es nicht schwierig, diese Regeln einzuhalten.

## Eigenschaften

## Positiv-Regler:

- Ausgangsspannung: 5 V (oder andere, abhangig von  $R_1/R_2$ )
  - Eingangsspannung: 6...17 V
  - Ausgangstrom:  $\leq 1$  A
  - Welligkeit:  $\leq 80$  mV<sub>SS</sub> bei  $U_{in} = 17$  V und  $I_{out} = 1$  A
  - Abmessungen und Anschlüsse kompatibel mit 7805 (TO-220)

## Negativ-Regler:

- Ausgangsspannung:  $-5\text{ V}$  (oder andere, abhangig von  $R_1/R_2$ )
  - Eingangsspannung:  $-6\ldots-17\text{ V}$
  - Ausgangstrom:  $0,85\text{ A}$
  - Welligkeit:  $\leq 36\text{ mV}_{\text{SS}}$  bei  $U_{\text{in}} = -17\text{ V}$  und  $I_{\text{out}} = 0,85\text{ A}$
  - Abmessungen und Anschlüsse kompatibel mit 7905 (TO-220)

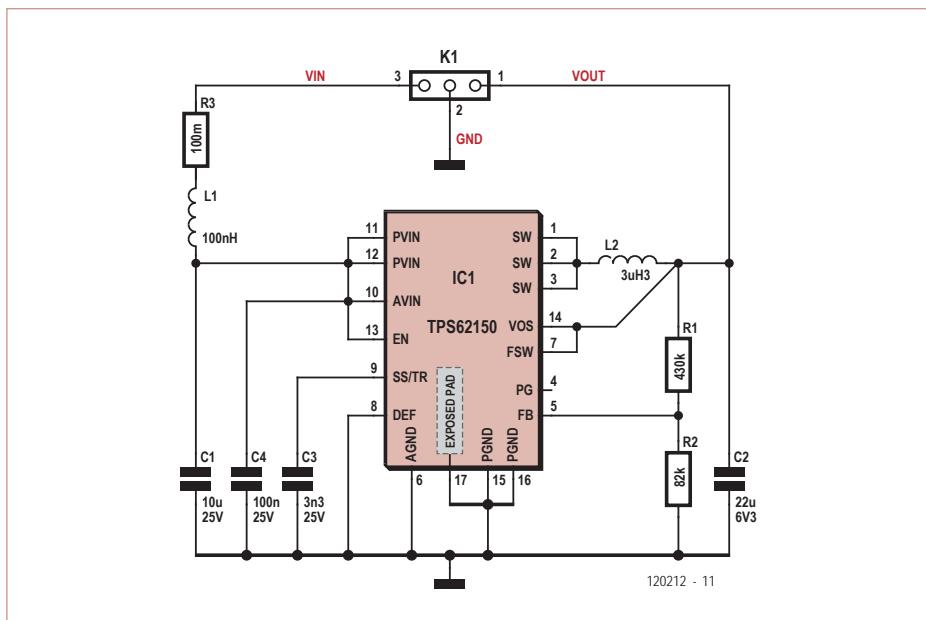


Bild 3. Schaltung des Positiv-Reglers.

## Stückliste Positiv-Regler

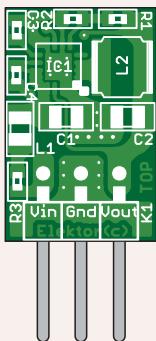


Bild 4. Doppelseitige Platine für den Positiv-Regler.  
(200%)

### Widerstände:

R1 = 430 k, 1 %/0,1 W, SMD 0402  
R2 = 82 k, 1 %/0,1 W, SMD 0402  
R3 = 0Ω1, 1 %/0,1 W, SMD 0402

### Kondensatoren:

C1 = 10 µ/25 V, X5R, SMD 0805 (z. B.  
TMK212BBJ106KG-T)  
C2 = 22 µ/6,3 V, X5R, SMD 0805 (z. B.  
JMK212BJ226MG-T)  
C3 = 3n3/25 V, X7R, SMD 0402  
C4 = 100 n/25 V, X5R, SMD 0402

### Induktivitäten:

L1 = 100 nH, SMD 0805 (z. B.  
MLZ2012DR10D)  
L2 = 3,3 µH, 4 · 4 mm (z. B.  
NRS4018T3R3MDG)

### Halbleiter:

IC1 = TPS62150RGTT (Abwärtswandler),  
QFN-16

### Außerdem:

K1 = Stifteleiste 3-polig abgewinkelt, Raster  
2,54 mm  
Platine 120212-1 (siehe [www.elektor.de/120212](http://www.elektor.de/120212))

Tabelle 1. Ausgangsspannungen  
des Positiv-Reglers.

U <sub>uit</sub>	R1	R2
5,0 V	430 k	82 k
3,3 V	256 k	82 k
2,5 V	174 k	82 k
1,8 V	102 k	82 k
1,5 V	72 k	82 k
1,2 V	41 k	82 k

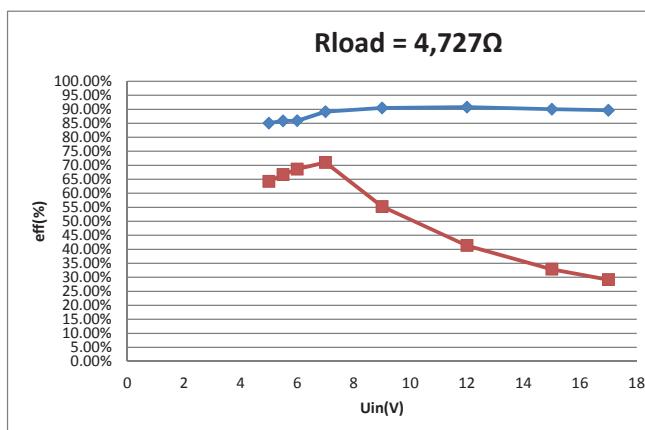
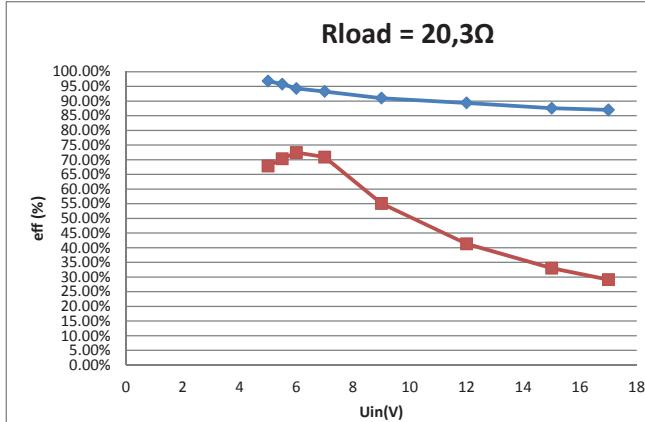


Bild 5. Wirkungsgrad der Positiv-Regler (blau: Schaltregler, rot: linearer 7805) bei den Ausgangsströmen 250 mA (Kurve a) und 1 A (Kurve b).

Unser schaltender Positiv-Regler ist ein vollwertiges Äquivalent des originalen 7805. Die Eingangsspannung kann bis 17 V betragen, der Ausgang ist bis 1 A belastbar. Die Kondensatoren, die normalerweise zum Eingang und Ausgang parallel liegen, können (falls bereits vorhanden) in der Schaltung

bleiben. In Neuentwicklungen können sie problemlos entfallen.

Die Diagramme in Bild 5 stellen den Wirkungsgrad des Positiv-Reglers bei unterschiedlichen Eingangsspannungen und Ausgangsströmen dar. Bei niedrigen Eingangsspannungen beträgt der Wirkungsgrad bis 250 mA ungefähr 95 %, er sinkt bei maximaler Eingangsspannung auf etwa 85 %. Wenn sich der Ausgangsstrom an der Obergrenze bewegt, liegt der Wirkungsgrad über den gesamten Eingangsspannungsbereich nahe 90 %.

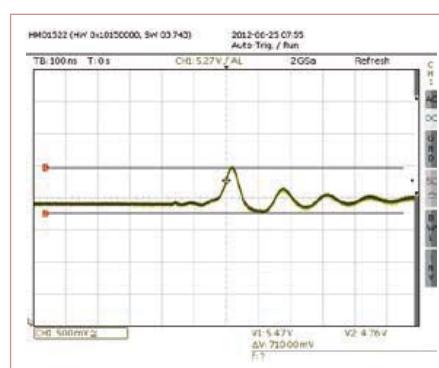


Bild 6. Verlauf der Ausgangsspannung bei impulsförmiger Laständerung.

Eine weitere wichtige Eigenschaft ist das Regler-Verhalten bei sprunghaften Laständerungen (Load Steps). Um diese Eigenschaft zu testen, haben wir die Last mit 50 kHz zwischen 10 % und 85 % der maximalen Last geschaltet. Das Diagramm in Bild 6 zeigt den Spannungsverlauf am Ausgang. Die hohe Spitze wird hauptsächlich durch die Messklemmen und die Zuleitungen verursacht. Unser Testaufbau ist nicht mit einer realen

Situation vergleichbar, er ist eher als „Worst-Case“ zu betrachten. Im durchschnittlichen Betrieb fällt die Spannungsspitze bei Lastwechsel deutlich niedriger aus.

## Negativ-Regler

Das Konzept des Positiv-Reglers und die Dimensionierung seiner Komponenten entsprechen weitgehend einer Schaltung, die Texas Instruments im Datenblatt des TPS62150 vorschlägt. Beim Negativ-Regler ist die Lage anders: Auf dem Markt sind zwar diverse integrierte Bausteine für schaltende Stromversorgungen, die Eingangsspannungen anheben, herabsetzen oder invertieren. Doch ein schaltendes IC, das eine negative Spannung auf eine weniger negative Spannung hoch regelt, ist uns nicht bekannt. Wie schon erwähnt, haben wir im Internet einen Schaltungsvorschlag gefunden, der diese Aufgabe trotzdem beherrscht.

Das bestimmende Bauelement ist ein Aufwärtswandler, der als so genannter *Negative Buck Converter* „verpolt“ angeschlossen ist. Die Anschlussbezeichnungen der Schaltung in **Bild 8** scheinen vertauscht zu sein: Die Schaltungsmasse ist mit Eingang  $V_{in}$  des TPS61170 verbunden, während die negative Eingangsspannung am GND-Anschluss des IC liegt. Diese Konfiguration ist notwendig, um die negative Eingangsspannung auf einen weniger negativen Wert anzuheben. Zu lösen ist hier ein Problem, das die Rückkopplung betrifft. Bei dieser Konfiguration muss das Rückkoppelsignal auf den GND-Anschluss des TPS61170 bezogen werden. Der Teilkoeffizient eines einfachen Spannungssteilers würde nicht mehr den korrekten Wert haben, wenn sich die Eingangsspannung bezogen auf den  $V_{CC}$ -Anschluss ändert. Das hätte zur Folge, dass der Regler nur bei einer definierten Eingangsspannung wunschgemäß arbeitet. Das Problem wird mit einem *Current Shunt Monitor*, einem MAX4073, im Rückkoppelzweig gelöst. Dieses IC setzt die Ausgangsspannung bezogen auf GND in eine Spannung um, die auf  $V_{in}$  bezogen ist.

Als Aufwärtswandler war ein Typ gesucht, der möglichst hohe Eingangsspannungen verträgt und gleichzeitig die nötige kleine Bauform hat. Die Wahl fiel auf den TPS61170 von Texas Instruments, **Bild 7** zeigt das interne Funktionsschema. Die

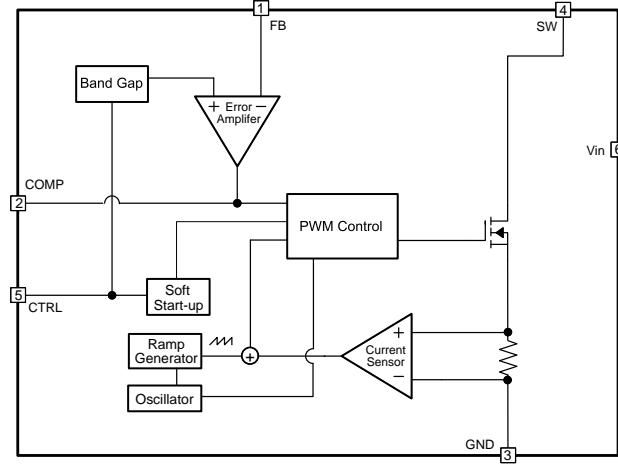


Bild 7. Internes Funktionsschema des TPS61170.

maximale Eingangsspannung  $V_{in}$  bezogen auf GND beträgt 20 V. Trotz der abweichen- den Konfiguration kann die Schaltung ähn-lich wie beim Positiv-Regler dimensioniert werden. Die LC-Tiefpässe am Eingang und Ausgang sind hier ebenfalls vorhanden. Das Ergebnis ist ein 7905-Ersatz (**Bild 9**), des- sen Abmessungen mit dem originalen 7905

identisch sind. Unser Negativ-Regler kann auch Spannungen liefern, die von -5 V abweichen, wenn der Spannungsteiler  $R1/R2$  wie in **Tabelle 2** angegeben modifiziert wird.

Auch die Eigenschaften unseres 7905-Ersatzes sind überzeugend. Aus **Bild 10** geht hervor, dass der Wirkungsgrad verglichen mit

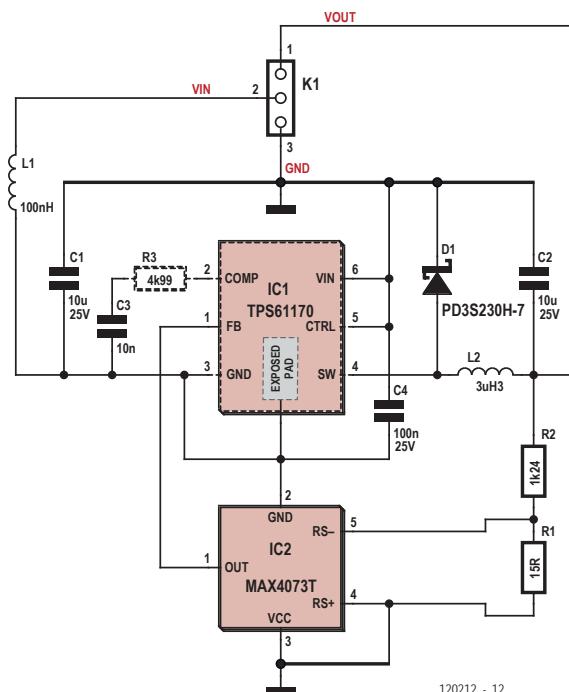


Bild 8. Schaltung des Negativ-Reglers.

## Stückliste Negativ-Regler

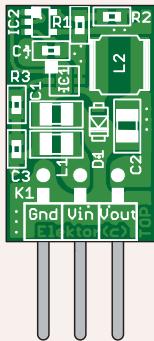


Bild 9. Doppelseitige Platine für den Negativ-Regler.  
(200%)

### Widerstände:

R1 = 15  $\Omega$ , 1 %/0,1 W, SMD 0402  
R2 = 1k24, 1 %/0,1 W, SMD 0402  
R3 = 4k99, 1 %/0,1 W, SMD 0402

### Kondensatoren:

C1 = 10  $\mu$ /25 V, X5R, SMD 0805 (z. B.  
TMK212BB106KG-T)  
C2 = 22  $\mu$ /6,3 V, X5R, SMD 0805 (z. B.  
JMK212BJ226MG-T)  
C3 = 10 n/25 V, X5R, SMD 0402  
C4 = 100 n/25 V, X5R, SMD 0402

### Induktivitäten:

L1 = 100 nH, SMD 0805 (z. B.  
MLZ2012DR10D)  
L2 = 3,3  $\mu$ H, 4 · 4 mm (z. B.  
NRS4018T3R3MDG)

### Halbleiter:

IC1 = TPS61170DRV (Aufwärtswandler),  
6-SON  
IC2 = MAX4073TAXK+T (Stromsensor),  
SC-70-5  
D1 = PD3S230H-7 Schottky-Diode,  
PowerDI323

### Außerdem:

K1 = Stifteleiste 3-polig abgewinkelt, Raster  
2,54 mm  
Platine 120212-2  
(siehe [www.elektor.de/120212](http://www.elektor.de/120212))

Tabelle 2. Ausgangsspannungen  
des Negativ-Reglers.

U <sub>uit</sub>	R1	R2
-5,0 V	15 $\Omega$	1k24
-3,3 V	23 $\Omega$ 2	1k24
-2,5 V	30 $\Omega$ 1	1k24
-1,8 V	42 $\Omega$ 2	1k24
-1,5 V	51 $\Omega$	1k24
-1,2 V	63 $\Omega$ 4	1k24

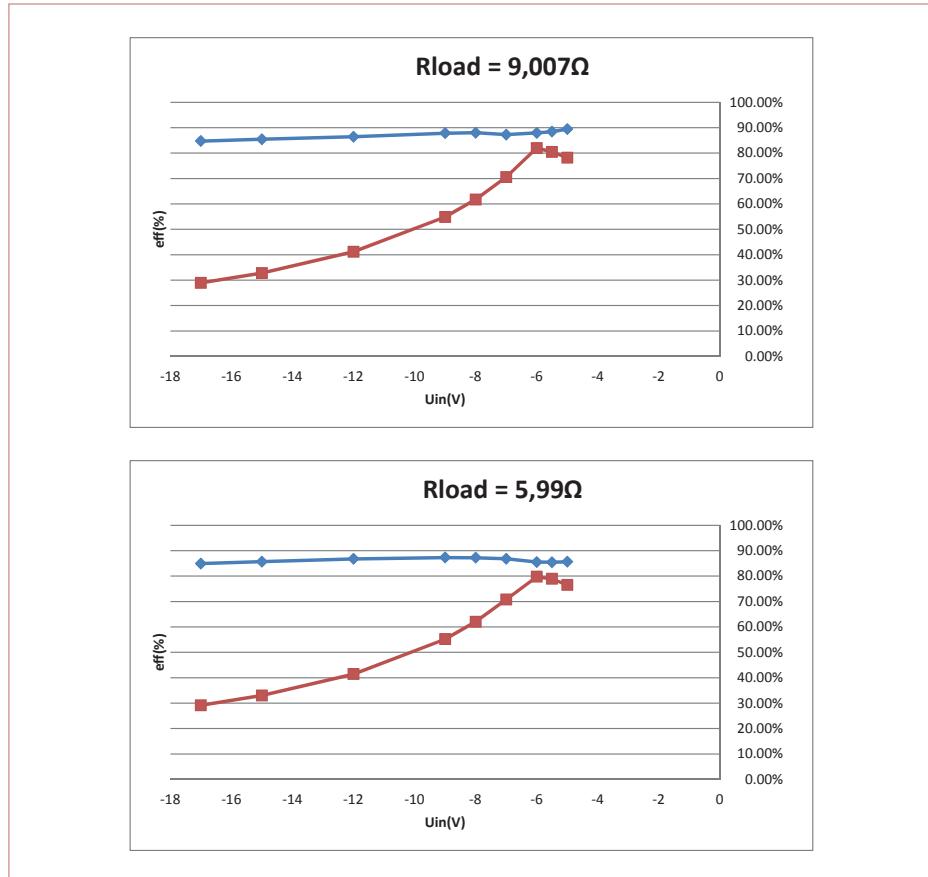


Bild 10. Wirkungsgrad der Negativ-Regler (blau: Schaltregler, rot: linearer 7905) bei den Ausgangsströmen 0,5 A (Kurve a) und 0,85 A (Kurve b).

dem originalen 7905 deutlich höher liegt. Die Belastbarkeit erreicht nicht ganz den Wert des Positiv-Reglers, dies ist auf die interne Strombegrenzung des TPS61170 zurückzuführen. Ferner tritt eine leichte Temperaturabhängigkeit der Strombegrenzung auf. Wenn die Begrenzung einsetzt und ein Luftstrom das IC kühlt (gegen das IC pusten!), bewegt sich die Spannung kurz-

zeitig in Richtung negativer Werte. Den Test mit pulsförmiger Last am Ausgang bestand der Negativ-Regler ebenfalls mit Bravour. Das Diagramm in Bild 11 zeigt, dass der Regler im gesamten Eingangsspannungsbereich stabil arbeitet. Der wiedergegebene Spannungsverlauf ist der „Worst Case“, die Eingangsspannung beträgt -17 V. Schaltende Spannungsregler sind zeitgemäße Alternativen für konventionelle lineare Spannungsregler. Die Welligkeiten der Ausgangsspannungen sind zwar stärker ausgeprägt, der Aufbau ist ein Stück komplexer. Doch wenn Verluste und Wirkungsgrade maßgebliche Kriterien eines Systementwurfs sind, stellen sie eine (fast) ideale Lösung dar.

(120212)gd

## Weblinks

- [1] TPS61170: [www.ti.com/product/tps61170](http://www.ti.com/product/tps61170)
- [2] TPS62150: [www.ti.com/product/TPS62150](http://www.ti.com/product/TPS62150)
- [3] [www.elektor.de/120212](http://www.elektor.de/120212)

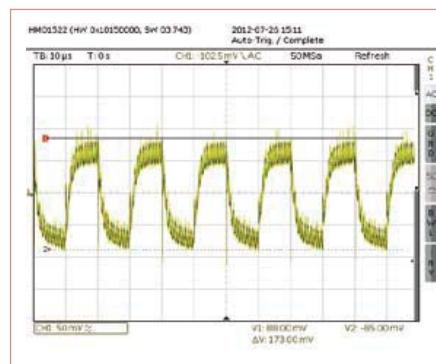


Bild 11. Welligkeit der Ausgangsspannung bei pulsförmiger Last (50 kHz, 60...560 mA,  $V_{in} = -17$  V, AC-Kopplung).

# WIRELESS POWER: CHARGING INNOVATION



Texas Instruments' Qi Compliant Wireless Power bqTESLA Evaluation Modules und die Komplettlösungen von TI & Würth ermöglichen Design-Ingenieuren Geräte für den Endverbraucher, z.B. Smart Phones, MP3-Player und Digital-Kameras, sowie die Applikationsinfrastruktur, z.B. bei Möbeln und Autos, mit Wireless Power Technologie auszustatten. Ganz gleich ob Sie Wireless Charging in ein bestehendes

oder neues Design integrieren wollen, wir haben die Tools, bieten die Unterstützung und verfügen über die notwendige Expertise! Beschleunigen Sie Ihren Design-Zyklus mit unseren Lösungen! Stellen Sie Ihr eigenes Kit aus TI Transmittern und Receiver Modulen sowie den entsprechenden Induktivitäten von Würth Elektronik zusammen! Und nutzen Sie EAGLE, um Ihr Design zu erstellen!



# Embedded Linux leicht gemacht (5)

## I/O, ADC, PWM, LAN & Webserver

Das Elektor-Linux-Board bringt eine Vielzahl von Möglichkeiten für die Anwendungsentwicklung mit. In dieser Folge werden wir digitale und analoge Signale einlesen. Dazu kommen eine Netzwerkverbindung und ein kleiner Webserver, der HTML-Seiten dynamisch generiert. Damit kann man sich den Status von LEDs und vielem mehr in einem Browser anzeigen lassen, und auch das Schalten und Walten aus der Ferne ist möglich.

Von Benedikt Sauter [1]

Inzwischen laufen Bootloader und Kernel, und viele Leser haben auch schon erste Erfahrungen mit dem Dateisystem und der SD-Karte gemacht (siehe hierzu auch den Textkasten „SD-Karten-Image“). In diesem Artikel geht es nun stetig auf die erste „richtige“ Anwendung zu. Da Embedded-Linux-Applikationen meist dort zum Einsatz kommen, wo Maschinen gesteuert und Daten erfasst

werden, zeigen wir in diesem Kursteil zuerst, wie man digitale und analoge Signale einliest und ausgibt. Danach richten wir eine Netzwerkverbindung ein. Hiermit bekommen wir einen Fernzugang zum Board, das sich nun zum Beispiel über eine Webseite steuern lässt.

### Digitale I/O-Pins

Im zweiten Teil dieser Serie haben wir bereits die LED1 auf dem Board ein- und ausgeschaltet. Die LED hängt am GPIO3-Pin des Prozessors. Diese GPIO-Pins können als Ein- oder Ausgang, aber auch als Interrupeingang verwendet werden. Die Vorgehensweise bei der Initialisierung solch eines I/O-Pins dürfte für die meisten Elektor-Leser nichts Neues sein:

- Aktivierung des Pins als GPIO.
- Initialisierung der Datenrichtung.
- Ausgabe eines Wertes bzw. Einlesen des aktuell anliegenden Signalpegels.

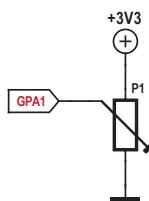


Bild 1. Mit einem Poti kann man den A/D-Wandler testen.

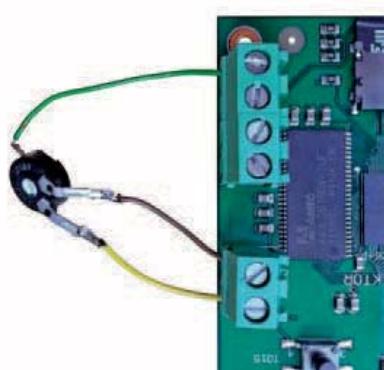


Bild 2. Der analoge Eingang ist auf eine Schraubklemme herausgeführt, so fällt das Anschließen leicht.

Unter Embedded Linux können wir die GPIO-Pins über die Gerätetreiber von der Konsole aus ansprechen.

Zuerst wechseln wir in den Ordner für die Kommunikation mit dem GPIO-Treiber:

```
cd /sys/class/gpio
```

Dann folgt die Aktivierung des mit der LED verbundenen Pins als GPIO (siehe Schaltplan in [2]):

```
echo 3 > export
```

Nun wollen wir gleich auch den Pin, der mit dem Taster verbunden ist, als GPIO aktivieren:

```
echo 15 > export
```

Dann konfigurieren wir die Pins als Aus- bzw. Eingang:

```
cd gpio3
echo "out" > direction
```

```
root@gnublin:~# echo 1 > /dev/lpc313x_adc
set res to 10
root@gnublin:~# cat /dev/lpc313x_adc
0x21f
root@gnublin:~# cat /dev/lpc313x_adc
0x11d
```

```
cd ../gpio15
echo „in“ > direction
```

Nun können wir die LED (wie schon einmal gezeigt) mit den folgenden Befehlen ein und ausschalten:

```
cd ../gpio03
echo 1 > value

echo 0 > value
```

Der Zustand des Tasters steht immer in einer (virtuellen) Datei namens „value“. Deren Inhalt kann man einfach mit dem Tool cat ausgeben:

```
cd ../gpio15
cat value
```

Jetzt sollte man in der Lage sein, auch das Relais auf dem Board zu schalten. Das Relais hängt an GPIO18, der Pin ist natürlich als Ausgang zu konfigurieren.

### Analog/Digital-Wandler

Der LPC3131 bietet vier analoge Eingänge mit je maximal 10 bit Auflösung. Der Wertebereich des Ergebnisses ist dann 0..1023 (bzw. 0..0x3FF in Hex-Darstellung). Als Spannungsreferenz wird 3,3 V verwendet, die Versorgungsspannung der I/O-Bank, auf der die Pins liegen.

Der Zugriff auf die gewandelten Werte läuft ähnlich wie beim Taster. Für die A/D-Wandler gibt es einen eigenen Treiber, der aber die Werte nur immer einzeln pro Kanal ausgeben kann. Zuvor muss man einstellen, welchen Kanal man abfragen möchte.

Am einfachsten kann man das Verhalten des A/D-Wandlers mit einem einfachen (Trimm-)Poti testen. Die Schaltung ist in **Bild 1** dargestellt. Das Potentiometer wird direkt an die 3,3-V-Versorgungsspannung angeschlossen, den Schleifer verbinden wir mit dem Pin GPA1, der auf eine Schraubklemme herausgeführt ist.

Das Ganze sollte dann grob wie in **Bild 2** aussehen. Jetzt können wir den A/D-Wandler initialisieren und nacheinander Werte ausgeben lassen, siehe hierzu **Bild 3**.

Möchte man zum Testen nicht immer wieder den gleichen Befehl eingeben, kann man dies mit dem Programm „watch“ automatisieren. Nach Eingabe von:

```
watch -n 1 cat /dev/lpc313x_adc
```

ruft das Tool ein Mal pro Sekunde den gewünschten Befehl auf. Mit Strg-C bricht man dies wieder ab.

Neben GPA1 sind auf dem Elektor-Linux-Board auch die ADC-Kanäle GPA0 und GPA3 verfügbar, und zwar an der Stiftleiste J5 (GPA2 ist nicht herausgeführt).

Möchte man den Eingang des A/D-Wandlers noch (in gewissen

Bild 3. Initialisierung des ADCs und Ausgabe von zwei Werten.

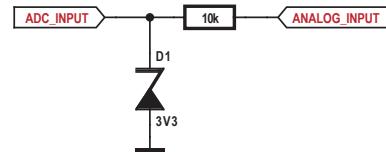


Bild 4. Schutz des ADC-Eingangs.

Grenzen) gegen Überstrom und Überspannung schützen, so kann man einen 10-k-Widerstand in Serie schalten und zusätzlich eine 3,3-V-Zener-Diode gegen Masse anschließen (**Bild 4**).

### PWM-Signal erzeugen

Die so genannte Puls-Weiten-Modulation hat viele Anwendungen: Spannungen erzeugen, Servo-Ansteuerung, Audio und vieles mehr [3]. Auf dem Elektor-Linux-Board ist ein PWM-Ausgang des Controllers auf die Stiftleiste J5 herausgeführt. Für einen Test schließen wir

#### Listing 1: PWM.

```
#include <stdio.h>
#include <stdlib.h>

#ifndef abs
#define abs(x) ((x) < 0 ? -(x) : (x))
#endif

int pwm(int value) {
    FILE* f = fopen(„/dev/lpc313x_pwm“, „wb“);
    fputc(value & 0xff, f);
    fputc((value >> 8) & 0xff, f);
    fclose(f);
}

int main() {
    int value = 0;
    int b;

    while(1) {
        b = abs(63 - 2*value);
        pwm(b * b);
        value = (value + 1) % 64;
        usleep(1000);
    }
}
```

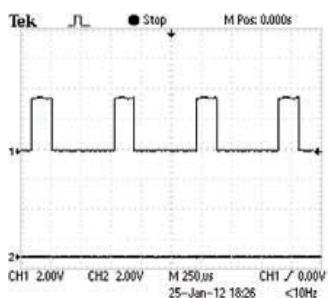


Bild 5. PWM-Ausgabe nach Eingabe von 1000 als Zähler-Vergleichswert.

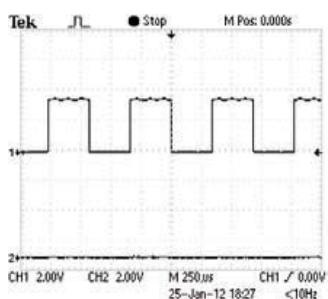


Bild 6. PWM mit 50 % Tastverhältnis.

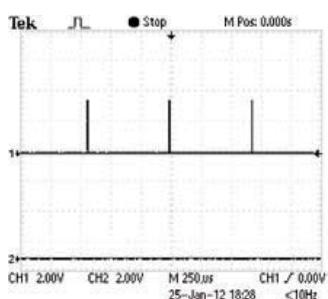


Bild 7. PWM mit 1 % Tastverhältnis.

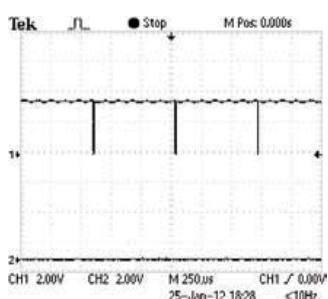


Bild 8. PWM mit 99 % Tastverhältnis.

dort am besten ein Oszilloskop an.

Im PWM-Modus inkrementiert der Controller einen 12-bit-Zähler pro Zählerakt. Sobald ein bestimmter Vergleichswert erreicht ist, wechselt die Flanke am PWM-Pin von High nach Low (wenn der Zähler überläuft, wird er zurückgesetzt und gleichzeitig der Pin wieder auf High gezogen). Als Vergleichswert kann man einen beliebigen 12-bit-Wert heranziehen, das entspricht einem Bereich von 0..4095. Wird als Wert eine 0 eingegeben, so wechselt das PWM-Signal sofort auf Low. Bei einer Eingabe von 2000 bekommen wir ein Tastverhältnis von etwa 50 %.

Da der PWM-Treiber im Gegensatz zu den I/O- und ADC-Treibern einen binären Eingabewert benötigt, können wir nicht einfach mit echo und cat arbeiten, da diese eine Eingabe automatisch als Zeichen (ASCII) interpretieren. Wir benötigen daher ein kleines Hilfsprogramm.

In C haben wir uns dies recht schnell auf dem Board selbst geschrieben. Doch man kann es noch etwas bequemer haben, denn solch ein Programm gibt es bereits (**Listing 1**). Im Home-Ordner, in dem man sich automatisch nach dem Anmelden befindet, liegt die Datei „pwm.c“. Im Code muss nur noch der Name der Gerätedatei angepasst werden.

Zuerst öffnen wir die Datei mit einem Editor auf dem Board:

```
nano pwm.c
```

Mit den Pfeiltasten geht man nun zur folgenden Zeile ...

```
FILE* f = fopen("/dev/pwm", "wb");
```

... und passt sie entsprechend an:

```
FILE* f = fopen("/dev/lpc313x_pwm", "wb");
```

Jetzt kann man die Datei mit Strg-o speichern und den Editor mit Strg-x beenden. Übersetzen kann man den Code auf dem PC, aber auch direkt auf dem Linux-Board, da wir dort ebenfalls über einen Compiler verfügen:

```
gcc -o pwm pwm.c
```

Ist das Programm übersetzt (dies dauert einen Moment) kann man es im Anschluss direkt ausführen:

```
./pwm
```

Im Oszi sieht man nun, wie sich das Tastverhältnis zwischen den Impulsen und Pausen stetig ändert.

Möchte man ein Signal mit einem festen Puls-/Pausenverhältnis ausgeben, kann man dies zum Beispiel mit einem kleinen Skript erledigen, das in der Programmiersprache Python geschrieben ist. Im Home-Ordner liegt schon eine Datei „pwm.py“.

Zuerst starten wir den Python-Interpreter:

## SD-Karten-Image

Beim Experimentieren – zu dem wir herzlich einladen! – kann es durchaus einmal sein, dass man in eine Sackgasse gerät. Da ist es mitunter hilfreich, wenn man den ursprünglichen Zustand der SD-Karte wieder herstellen kann. Deshalb bieten wir den Inhalt der SD-Karte als Extra-Download an. Zuerst lädt man das Image von der Elektor-Website herunter [8] (Download „SD Card Image“, 120180-12.zip). Anschließend entpackt man das Archiv:

```
unzip 120180-12.zip
```

Es dauert dann einen kurzen Moment bis die folgende Ausgabe erscheint:

```
Archive: ../120180-12.zip inflating: Elektor_Linux_Board - Build_New_SD_Card.txt inflating: gnublin.img
```

Jetzt kann man die zu beschreibende SD-Karte in den PC oder Kartenleser stecken. Das System hängt die SD-Karte nun automatisch ein, was wir hier aber nicht brauchen, denn wir möchten das Image 1:1 auf die SD-Karte schreiben. Daher muss man die Karte zuerst wieder manuell aushängen.

Am besten geht man auf die Konsole und gibt dort nach dem Anstecken des Kartenlesers den Befehl `dmesg` ein. Nun erhält man eine Ausgabe wie:

```
[ 1069.427374] sdf: sdf1 sdf2 [ 1069.430857] sd 5:0:0:0: [sdf] No Caching mode page present [ 1069.430863] sd 5:0:0:0: [sdf] Assuming drive cache: write through [ 1069.430868] sd 5:0:0:0: [sdf] Attached SCSI removable disk [ 1070.002620] EXT2-fs (sdf1): warning: mounting unchecked fs, running e2fsck is recommended
```

In den letzten Zeilen steht, was für Gerätenamen der Kernel für die SD-Karte vergeben hat.

Jetzt muss man die SD-Karte manuell aushängen ...

```
umount /dev/sdf1
```

... wobei `sdf1` durch den Gerätenamen der SD-Karte zu ersetzen ist (genau genommen den Gerätenamen der ersten Partition).

Dann kann man das heruntergeladene Image auf die SD-Karte schreiben:

```
sudo dd if=gnublin.img of=/dev/sdf
```

`sdf` ist hier die Bezeichnung der ganzen Karte als Blockdevice.

Hinweis: Der Schreibvorgang kann bis zu 10 Minuten dauern.

```
python
```

In den Interpreter (wir befinden uns nun bereits in einem interaktiven Modus) laden wir anschließend das PWM-Modul (eine Bibliothek mit Python-Funktionen):

```
import pwm
```

Jetzt kann man die Funktionen des Moduls aufrufen. Eine dieser Funktionen nimmt direkt den PWM-Zählervergleichswert entgegen:

```
pwm.pwm_raw(1000)
```

Das Signal im Oszilloskop müsste nun wie in **Bild 5** aussehen.

Eine Eingabe des Tastverhältnisses in Prozent, über

```
pwm.pwm(50)
```

führt zu einem Signal wie in **Bild 6**.

Mit den Befehlen

```
pwm.pwm(1)
pwm.pwm(99)
```

lassen sich entsprechend Puls-/Pausenverhältnisse von 1 und 99 % generieren (**Bild 7** / **Bild 8**).

Mit Strg-d kann man den Python-Interpreter wieder beenden. Leider dauert es immer ein wenig, bis Python gestartet ist. Wenn der Interpreter aber einmal läuft, reagiert er recht flüssig.

### Netzwerkschnittstelle

Im letzten Teil haben wir bereits gezeigt, wie man das System dazu bringt, mit einem USB/UART-Adapter zusammenzuarbeiten. Nun wollen wir einen weiteren USB-Adapter installieren, um das Linux-Board mit einem Ethernet-Netzwerk verbinden zu können. Wir verwenden einen handelsüblichen USB/LAN-Adapter (**Bild 9**). Hier gibt es viele verschiedene Modelle, aber die meisten Geräte greifen auf ähnliche, klassische Chips zurück. Wer ganz auf Nummer sicher gehen will: Für diese Ausgabe haben wir einen „D-Link DUB-E100“ verwendet [4].

In der letzten Folge hatten wir für den USB/UART-Adapter einen eigenen Treiber in den Kernel integriert. Wie schon im Laufe der Serie erwähnt, bietet der Kernel aber auch die Möglichkeit, Geräte-Treiber als Modul zur Laufzeit nachzuladen. Beim Netzwerkadapter wollen wir genau das tun. Im Dateisystem liegen bereits verschiedene Treiber bereit.

Im Falle des D-Link-Adapters müssen wir folgenden Befehl eingeben:

```
modprobe asix
```

Wir müssten nun eine Ausgabe wie in **Bild 10** erhalten. Im Dateisystem befinden sich insgesamt drei Treiber:



Bild 9. Ein USB/LAN-Adapter erweitert unser Board um eine Netzwerk-Schnittstelle.

- asix
- pegasus
- net1080

Nutzt man einen anderen Adapter, kann man einfach die Treiber testweise nacheinander laden. Ob man den richtigen gefunden hat, sieht man nach der Eingabe von

ifconfig -a

Erscheint das „eth0“-Interface in der Ausgabe, dann ist das Netzwerk bereit zum Übertragen von Daten.

Falls keiner der Treiber passt, kann man im Kernel unter „Device Drivers“ -> „Network device support“ -> „USB Network Adapters“ weitere Treiber manuell hinzufügen. Entweder man nimmt den Treiber

fest in den Kernel auf (Empfehlung für Einsteiger, siehe [2]). Oder man muss die Module nach dem Übersetzen in das Dateisystem kopieren und später vom Kernel nachladen lassen. Alle Module übersetzen kann man mit:

make modules

Auf die Karte installiert man die neuen Module mit

make modules\_install INSTALL\_MOD\_PATH=/mnt

Statt /mnt muss man hier den Pfad zur eingehängten SD-Karte angeben.

Wird der Netzwerkadapter jetzt erkannt, dann kann man temporär eine IP-Adresse vergeben. Sicherheitshalber sollte man zuvor mit einem PC im gleichen Netzwerk prüfen, ob die IP-Adresse frei ist. Das geht unter Linux (und Windows) von der Konsole aus mit dem Befehl ping:

ping 192.168.0.7

Wenn das Programm meldet, dass kein Gerät antwortet ...

2 packets transmitted, 0 received, 100% packet loss, time 1006ms

... dann kann man diese IP-Adresse vergeben. Hierzu geben wir auf dem Linux-Board ein:

ifconfig eth0 192.168.0.7

Ein erneutes Ping vom PC aus sollte jetzt zu einer positiven Meldung führen (Bild 11).

Optional kann man sich auch von einem DHCP-Server automatisch eine Netzwerkadresse zuweisen lassen (Bild 12).

Soll der Treiber bei jedem Start automatisch geladen werden, dann kann man dessen Namen in die Datei „/etc/modules“ eintragen. Hier stehen alle Module untereinander, die automatisch beim Boot von Linux geladen werden sollen.

Die IP-Adresse hinterlegt man in der Datei „/etc/network/inter-

```
root@gnublin:/lib/modules/2.6.33/kernel/drivers/net/usb# usb 1-1: new high speed USB device using lpc-ehci and address 2
usb 1-1: New USB device found, idVendor=2001, idProduct=3c05
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1: Product: DUB-E100
usb 1-1: Manufacturer: D-Link Corporation
usb 1-1: SerialNumber: 000001

root@gnublin:/lib/modules/2.6.33/kernel/drivers/net/usb# modprobe asix
eth0: register 'asix' at usb-lpc-ehci.0-1, ASIX AX88772 USB 2.0 Ethernet, fc:75:16:cf:6b:7f
usbcore: registered new interface driver asix
root@gnublin:/lib/modules/2.6.33/kernel/drivers/net/usb#
```

Bild 10. Automatische Erkennung des USB/LAN-Adapters.

```
sauter@pc:~$ ping 192.168.0.7
PING 192.168.0.7 (192.168.0.7) 56(84) bytes of data.
64 bytes from 192.168.0.7: icmp_req=1 ttl=64 time=1.24 ms
64 bytes from 192.168.0.7: icmp_req=2 ttl=64 time=1.50 ms
--- 192.168.0.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.248/1.375/1.502/0.127 ms
```

faces“. Diese Datei existiert bereits in unserem Dateisystem. Mit einem Editor öffnet man diese und trägt bei „eth0“ die eigene IP-Adresse ein.

Nach dem Start ist das Elektor-Linux-Board nun immer im hauseigenen Netzwerk verfügbar.

## Webserver

Nachdem jetzt steht die Netzwerkverbindung steht, können wir einen kleinen Webserver starten, um eine erste Demoseite per Browser anzeigen zu können. Im Homeverzeichnis des Benutzers „root“ liegt ein kleines Skript, das den bekannten Webserver „lighttpd“ anwirft:

```
root@gnublin:~# ./lighttpd-init.sh
```

Syntax OK

```
root@gnublin:~#
```

Besucht man mit einem Browser die vorher vergebene IP-Adresse, dann erscheint die Webseite in **Bild 13**.

Ein Webserver stellt typischerweise statische HTML-Seiten zur Verfügung. Wenn wir uns nun zum Beispiel den Status einer LED im Browser anzeigen lassen wollen, dann muss unser Webserver eine HTML-Seite ausliefern, die dynamisch (abhängig vom LED-Status) zusammengebaut wird. Wir benötigen hierfür eine Schnittstelle zwischen dem Webserver und einem externen Programm, das erkennen kann, ob die LED nun gerade an oder aus ist, und eine entsprechende Webseite generiert.

Die einfachste Verbindung heißt hier CGI. Das „Common Gateway Interface“ ist eine Schnittstelle, mit welcher der Webserver nahezu beliebige Programme aufrufen kann. Eine Bedingung ist nur, dass das Programm kommandozeilenorientiert arbeitet, also von der Konsole aus (mit eventuellen Parametern) gestartet werden könnte. Außerdem muss das Hilfsprogramm direkt eine HTML-Seite ausgeben. Als CGI-Programm kann man ein einfaches Linux-Shell-Skript, ein PHP- oder Python-Programm und sogar ein C-Programm verwenden.

## LED schalten vom Browser aus

Für eine kleine Demo-Applikation wollen wir gleich die erste Möglichkeit nutzen. Wir erstellen uns eine einfache Skript-Datei, welche die Shell (Konsole) direkt ausführen kann. Zuerst aber müssen wir uns das CGI-Interface im Webserver einrichten.

In der Datei „/etc/lighttpd/modules.conf“ muss man den Eintrag ...

```
#include "conf.d/cgi.conf"
```

... mit einem Editor (z.B. nano oder vi) ändern in:

```
include „conf.d/cgi.conf“
```

Bild 11. Meldung nach einem Ping.

```
root@gnublin:~# udhcpc
udhcpc (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.182...
Lease of 192.168.0.182 obtained, lease time 864000
Setting IP address 192.168.0.182 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
  Adding DNS server 192.168.0.1
root@gnublin:~#
```

Bild 12. Hier lassen wir uns vom DHCP-Server eine Adresse zuweisen.

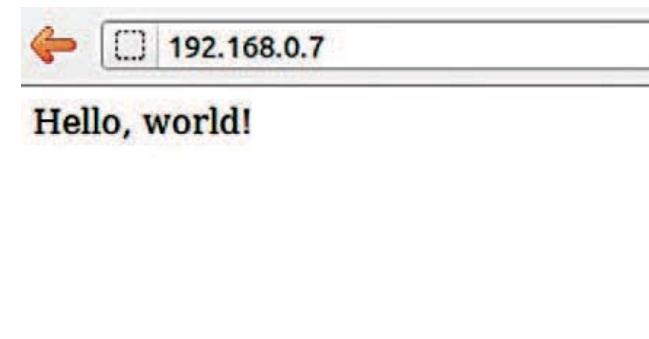


Bild 13. Testseite des Webservers.

Anschließend muss in der Datei „/etc/lighttpd/conf.d/cgi.conf“ der Eintrag

```
#alias.url += ( "/cgi-bin" => server_root + "/cgi-bin" )
```

in

```
alias.url += ( "/cgi-bin" => var.server_root + "/cgi-bin" )
```

geändert werden. Jetzt weiß der Webserver, dass er die Dateien im Ordner „/cgi-bin“ als Programm ausführen soll (und nicht als HTML-Seite an den Browser senden).

Damit wir schließlich ein einfaches Shell-Skript als CGI-Programm verwenden können muss in der Datei der Bereich ...



Bild 14. LED schalten im Browser.

```
cgi.assign = ( ".pl" => "/usr/bin/perl",
               ".cgi" => "/usr/bin/perl",
               ".rb" => "/usr/bin/ruby",
               ".erb" => "/usr/bin/eruby",
               ".py" => "/usr/bin/python" )
```

... mit der Zeile ".sh" => "/bin/sh" erweitert werden:

```
cgi.assign = ( ".pl" => "/usr/bin/perl",
               ".cgi" => "/usr/bin/perl",
               ".rb" => "/usr/bin/ruby",
               ".sh" => "/bin/sh",
               ".erb" => "/usr/bin/eruby",
               ".py" => "/usr/bin/python" )
```

Im nächsten Schritt muss man den Ordner für die CGI-Programme anlegen:

```
mkdir -p /srv/www/htdocs/cgi-bin
```

Und schließlich das Programm in **Listing 2** eingeben, nachdem man den Editor mit folgendem Befehl gestartet hat:

```
nano /srv/www/htdocs/cgi-bin/example.sh
```

Das Listing findet man auch im Download zu diesem Kursteil [5], so kann man sich einiges an Tipparbeit sparen.

Damit der Webserver die LED ansteuern kann, muss man zuvor noch den Pin konfigurieren und die Datenrichtung auf der Konsole einstellen:

```
echo 3 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio3/direction
```

Weil der Webserver aus Sicherheitsgründen nie als User „root“ läuft, müssen wir noch temporär Rechte vergeben, mit denen der Webserver auf die LED zugreifen darf:

```
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value
```

Jetzt darf jeder im System auf die LED zugreifen. Das ist eigentlich nicht optimal, doch ein detailliertes Vergeben der Rechte würde an dieser Stelle zu weit führen. Infos findet man hierzu unter [6]. Außerdem braucht der Webserver noch die passenden Rechte, um Logdateien in einem vorher generierten Verzeichnis ablegen zu dürfen:

```
mkdir /var/log/lighttpd
chown -R lighttpd:lighttpd /var/log/lighttpd
```

Startet man den Webserver jetzt ...

```
root@gnublin:~# /etc/init.d/lighttpd restart
```

... dann müsste die Meldung erscheinen:

Syntax OK

Wenn man per Browser auf die zuvor eingestellte IP-Adresse zugreift, dann öffnet sich die Seite in **Bild 14**, die den Status des LED-Pins anzeigt.

Auch eine kleine Steuerung haben wir eingebaut, mit Hilfe eines Mini-HTML-Formulars, das in diesem Falle nur einen Submit-Button enthält. Drückt man auf einen solchen Button, dann übermittelt der Browser normalerweise den Inhalt der Formular-Steuerelemente an unseren Webbrowser. In diesem Fall benutzen wir den Mechanismus

## Listing 2: CGI-Skript zur Generierung der Webseite.

```
#!/bin/sh

if [ "$REQUEST_METHOD" == "POST" ]
then
    if [ 'cat /sys/class/gpio/gpio3/value' == 1 ]
    then
        echo 0 > /sys/class/gpio/gpio3/value
    else
        echo 1 > /sys/class/gpio/gpio3/value
    fi

    echo "Content-Type: text/html; charset=utf-8"
    echo ""
    echo "<html>"
    echo "  <head>"
    echo "    <title>Webserver CGI Port 3 (LED)</title>"
    echo "  </head>"
    echo "  <body>"
    echo "    <h1>Control-Panel CGI Port 3</h1>"
    if [ 'cat /sys/class/gpio/gpio3/value' == 1 ]
    then
        echo "    Port: On"
    else
        echo "    Port: Off"
    fi
    echo "    <br><br>"
    echo "    <form action=\"/cgi-bin/example.sh\" method = \"POST\">"
    echo "      <input type=\"submit\" value=\"Click\">"
    echo "    </form>"
    echo "  </body>"
    echo "</html>"
```

mus nur, um dem Webserver mitzuteilen, dass er erneut unser CGI-Skript „/cgi-bin/example.sh“ aufrufen soll. Dieses togglet die LED1 auf dem Board und baut die neue Webseite mit der veränderten Statusmeldung zusammen.

## Ausblick

In der nächsten Ausgabe werden wir eine etwas komplexere HTML-Oberfläche erstellen, mit der wir weitere Funktionen auf dem Board steuern können. Doch natürlich ist es mit einem reinen User-Interface nicht getan, unsere Steuerung benötigt auch etwas Intelligenz. Dies können wir mit einem kleinen Programm, das im Hintergrund läuft, verwirklichen.

Für den übernächsten und letzten Teil dieser Serie (in der ersten Ausgabe des Jahres 2013) haben wir uns noch etwas Besonderes ausgedacht: Unsere Leser können selbst auswählen, welche Themen noch behandelt werden sollen. Auf einer speziellen Webseite [7] können Sie hierüber abstimmen!

(120182)

## Weblinks

- [1] [sauter@embedded-projects.net](mailto:sauter@embedded-projects.net)
- [2] [www.elektor.de/120181](http://www.elektor.de/120181)
- [3] <http://de.wikipedia.org/wiki/Pulsweitenmodulation>
- [4] <http://shop.embedded-projects.net/gnublin>
- [5] [www.elektor.de/120182](http://www.elektor.de/120182)
- [6] [http://en.gnublin.org/index.php/Permissions\\_GPIO](http://en.gnublin.org/index.php/Permissions_GPIO)
- [7] [www.elektor.de/linux-feedback](http://www.elektor.de/linux-feedback)
- [8] [www.elektor.de/120180](http://www.elektor.de/120180)

— Anzeige



# analog and sensor solutions.

ams verbindet nahtlos die reale analoge Welt mit der digitalen und schafft so intuitive Technologien, die unser Leben bereichern.

Treffen Sie die Experten!  
Halle A5 / Stand 107



ams bietet innovative  
analoge Lösungen für  
die anspruchvollsten  
Applikationen für  
Sensoren und Sensor-  
schnittstellen, Power  
Management und Wireless.

**ams**

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

# Nixie-VU-Meter

## Retro-Leistungsanzeige für Röhrenverstärker

Diese Schaltung ist die Krönung eines (Röhren-) Verstärkers. Mit ihr kommt man zu einer angenehm leuchtenden, kalibrierten Anzeige für Leistung oder Signalpegel. Die Intelligenz der Schaltung steckt in einem PSoC aus Kalifornien, das zwei Nixie-Röhren aus der Ukraine ansteuert.



Von Harald W. Cintra (Brasilien)

Entfernungen bedeuten kaum etwas und selbst Grenzen verschwinden. Das gilt besonders für Elektronik, denn diese Elektor-Schaltung aus dem Jahr 2012 kombiniert ein PSoC (Programmable System on Chip) und IN-9-Nixie-Röhren, die gut 40 Jahre Technikgeschichte trennen. Es ist interessant zu sehen, wie sich im Internet-Zeitalter „Sovjet-elektronisjk“-Bauteile aus den 1960ern und heutige Technik aus dem Silicon Valley vertragen.

### Warum IN-9?

Die Nixie-Röhre IN-9 ist für diese Schaltung ideal. Dabei handelt es sich um eine lineare Gasentladungsröhre mit orange-roter Leuchtfarbe und etwa 150 V Anodenspannung. Die Länge ihres Leuchtbalkens ist weitgehend proportional zum fließenden Strom. Die Linearität ist jedoch beschränkt (siehe Bild 1). Zwischen 0 und 8 mA gilt ein Zusammenhang von rund 10,3 mm/mA, doch darüber wird es zunehmend „borsj“ bzw. uneindeutig. Diese Röhren kann man per Internet immer noch als NOS (new old stock = neue Altlagerware) ordern. Witzigerweise geht das bestens über eine Auk-

### Technische Daten

- Stereo-VU-Meter mit Balkenanzeige
- 2x Nixie-Röhren vom Typ IN-9
- Leuchtfarbe: orange/rot
- Balkenlänge: ca. 85 mm
- Eingang: Line-Pegel (ca. 700 mV)
- PSoC CY8C27443 von Cypress
- Fertig programmiertes PSoC erhältlich
- Software-Logarithmierer
- 150-V-Anodenspannung mit 555
- Kompakte Platine (65 x 65 mm)

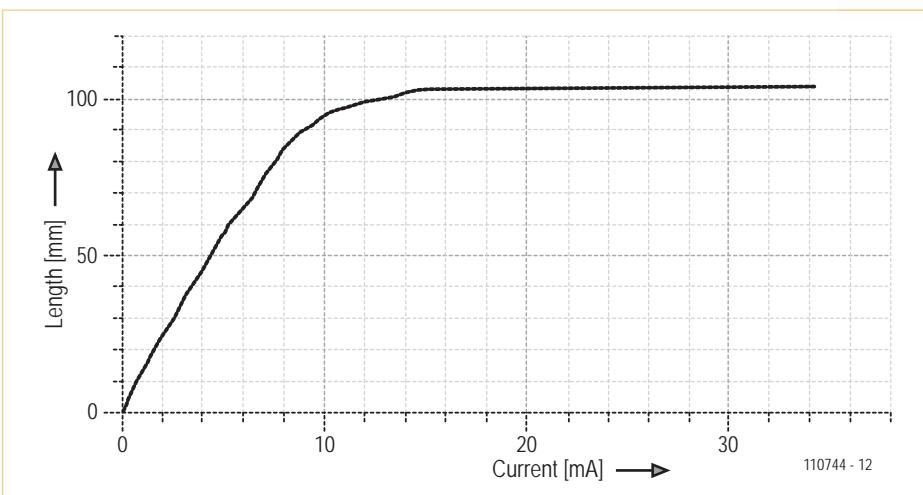


Bild 1. Gemessener Zusammenhang von Balkenlänge der Nixie-Röhre IN-9 und Anodenstrom (keine Herstellerangabe). Im Bereich von 0 bis 8 mA ist er linear. Neu gezeichnete Grafik des Originals (2004) mit freundlicher Genehmigung von Jeff Malins.

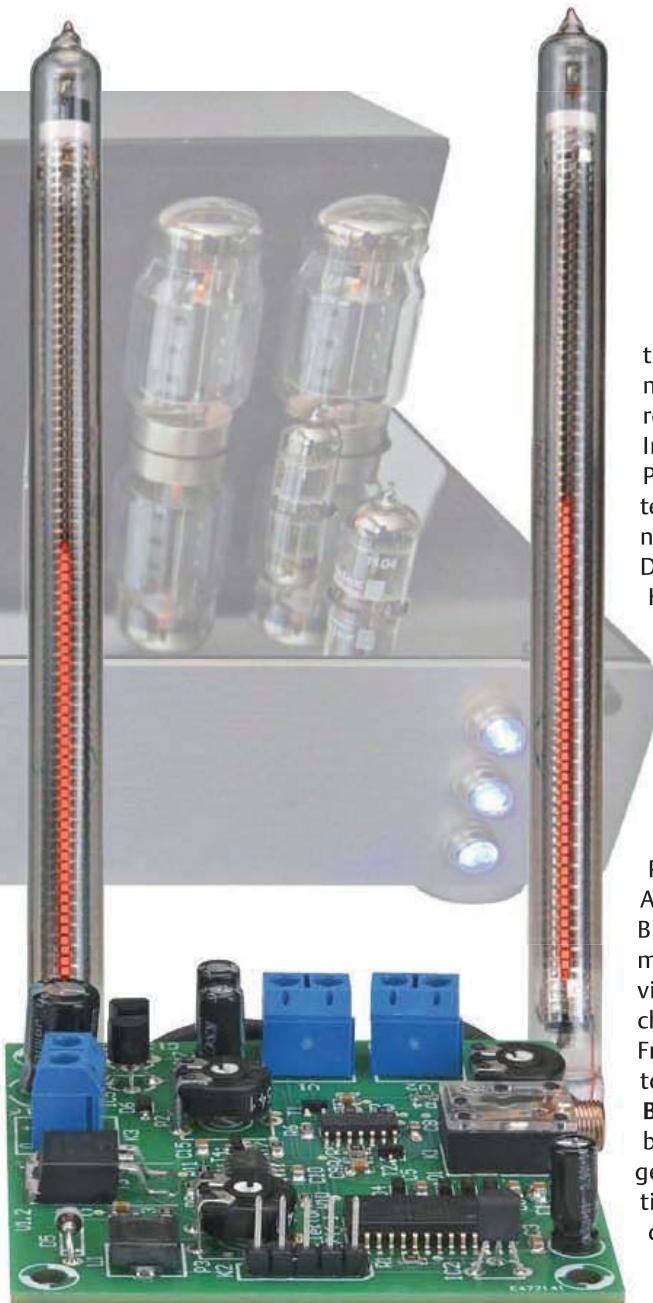


Bild 2. Nixie-Röhren und Kleinigkeiten in „Sowjet-Verpackung“. Keine Angst: Alles funktioniert prima!

tionswebseite, deren Betreiber nicht weit weg von der Firma Cypress in Kalifornien ansässig ist... In letzter Zeit wurden Elektor-Projekte wie ein Thermometer [1] und die Sputnik-Uhr [2] mit numerischen Nixies ausgestattet. Deshalb sei für den theoretischen Hintergrund auf diese Artikel verwiesen. Hier genügt eine kurze Anleitung für die, die es nicht mehr selbst erlebt haben, wie einmal Röhren die Elektronik dominiert haben. Es ist schließlich kaum anzunehmen, dass ein Elektroniker 2060 angesichts eines PSoC aus dem Häuschen gerät. Aber warten wir das einfach ab ;-) Bezuglich Datenblättern sollte man bei der IN-9 besser nicht viel Offizielles erwarten. Bei solchen Antiquitäten ist Google dein Freund. Die Lieferung für Elektor wurde (so wie sie ankam) für **Bild 2** abfotografiert (spaßeshalber wurde eine Babuschka dazu gestellt). Neben dem gegenwärtigen Stand ukrainischer Verpackungstechnik kann man einige numerische Nixie-Röhren, einen Sockel und ein paar Treiber-ICs

bewundern, die wir zusammen über eBay geordert hatten.

### Ein PSoC-Job

Eine VU-Anzeige muss logarithmisch sein. Mit einer linearen Anzeige würden optischer und akustischer Eindruck nicht zusammenpassen. In diesem Projekt wird per Software logarithmiert.

Das Cypress-PSoC CY8C27443 ist in Elektor selten zu finden. Dieses PSoC eignet sich aufgrund der Leistung (mit den vielen ADCs, DACs und Verstärkern) sehr gut für diesen Zweck. Die Anzahl externer Bauteile wird so auf ein Minimum beschränkt.

Mit Ausnahme der Elektronik für die Anodenspannung etc. stecken alle wichtigen Funktionen im PSoC - siehe die Blockschaltung von **Bild 3**. Von links nach rechts: Auf den Eingangsverstärker folgen ein Gleichrichter und ein zweiter Verstärker. Ein A/D-Konverter liefert die Daten für den software-basierten Logarithmierer. Die logarithmischen Daten werden von einem D/A-Konverter in analoge Werte umgesetzt und steuern eine Stromquelle, die eine IN-9-Röhre treibt. Das Ganze natürlich doppelt für den linken und rechten Kanal, denn anders als 1960 sind Audio-Verstärker heute stereo.

### Konkrete Kombination

Die Schaltung von **Bild 4** beweist, dass ein PSoC die Anzahl der notwendigen Bauteile stark reduzieren kann.

Die Elektronik wird von einem 12-V-Stecker-Netzteil gespeist. Die notwendige Anodenspannung von 150 V wird mit Hilfe eines einfachen Aufwärtsreglers generiert, der mit dem altbekannten 555 (IC6) als AMV (Astabilier MultiVibrator) operiert. Mit Hilfe des Eingangs „control voltage“ wird dabei die Ausgangsspannung stabil gehalten. R10, P3, R13 bilden einen Spannungsteiler, der über T4 das Rückkoppelungssignal für die Regelung liefert. Mit P3 kann die Anodenspannung auf 150 V eingestellt werden. Ein Chip wie IC1 benötigt 5 V. Die Opamps benötigen mit 10 V etwas mehr. Beide Spannungen werden mit je einem 7805-Spannungsregler erzeugt. Bei IC5 liegt eine 5,1-V-Z-Diode in der Masseleitung, was die Ausgangsspannung um 5,1 V erhöht.

Mit einer 5-V-Versorgung und einem sau-

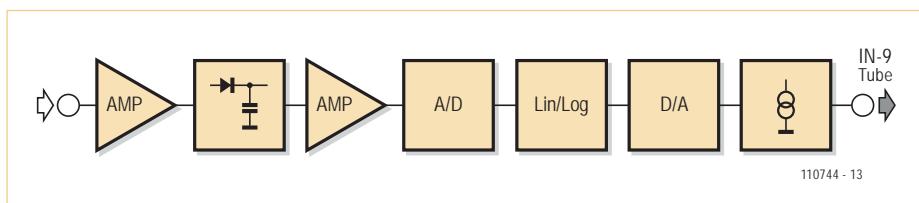


Bild 3. Audio-Signalkette – fast alles wird im PSoC erledigt.

## Alte Sowjet-Elektronik trifft Silicon-Valley-High-Tech

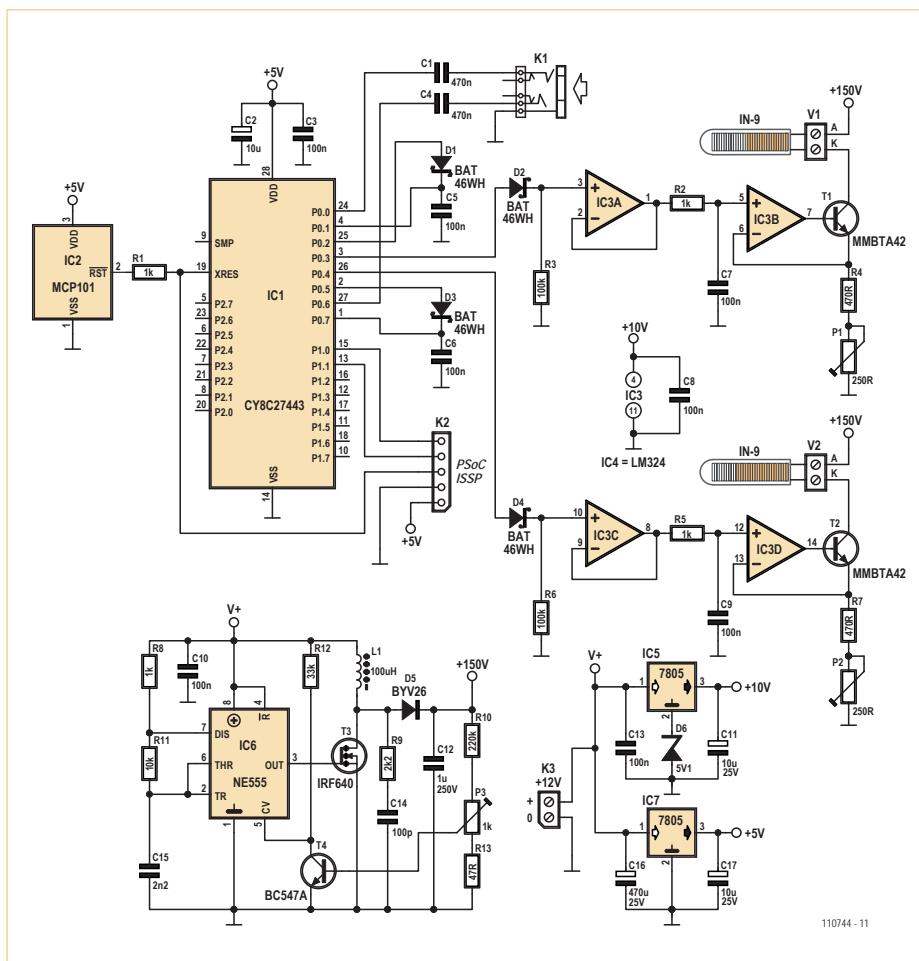


Bild 4. Die Schaltung ist ein gutes Beispiel dafür, wie ein PSoC die Anzahl an Bauteilen klein halten kann. Doch ohne die Hilfe der IN-9-Nixie-Röhren wird man nichts von seiner Arbeit sehen. Neben den 1960er Röhren ist mit dem NE555 noch ein IC aus den 1970er Jahren notwendig.

beren Reset-Signal funktioniert so ein PSoC wie er soll. Letzteres generiert IC2. Dieser Reset-Generator vom Typ MCP101 sorgt für einen sauberen Power-on-Reset.

Zum Audio-Pfad: Das Eingangssignal mit Line-Pegel an K1 passiert zunächst Kondensatoren (C1/C4) und gelangt dann direkt in das PSoC, wo es zweifach verstärkt wird. Das verstärkte Signal wird dann zwecks Gleichrichtung mit Schottky-Dioden (D1/D3, Low-Drop-Typen) an PSoC-Pins geführt. Über weitere Pins gelangt das gleichgerichtete Signal wieder in das PSoC.

Im PSoC wird dieses Signal zunächst leicht verstärkt und dann einem internen 8-bit-A/D-Konverter zugeführt. Dessen Ausgangswerte selektieren aus einer Tabelle die passenden logarithmischen Werte, die dann per D/A-Konverter zu analogen Signalen werden. Diese verlassen das PSoC und gelangen über Dioden (D2/D4) an als Puffer geschaltete Opamps (IC3A/C). Nun kommen Tiefpässe mit 1,6 kHz Grenzfrequenz (R2/C7, R5/C9) an. Die gefilterten Signale steuern dann die Stromquellen aus Opamp (IC3B/D) und Hochspannungstransistor (T1/T2). Über Trimmotis (P1/R4) kann man den Strom und damit den Ausschlag der Nixie-Röhre variieren. Der Zweck der Dioden D2/D4 im Signalweg ist der, eine kleine Schwelle einzufügen, damit nicht jedes kleinste Signal schon Aktivität bei den Röhren nach sich zieht. P1/P2 werden so eingestellt, dass der maximale Eingangspiegel auch dem maximalen Ausschlag der Röhren entspricht.

Zum PSoC: Bild 5 zeigt die interne Konfiguration. In Bild 6 erkennt man die Eingangsverstärker PGA\_2/4, deren Ausgangssignale den Chip verlassen, gleichgerichtet mit PGA\_1/3 gepuffert und danach per DUALADC8 digitalisiert werden. Mit



etwas Mühe kann man sehen, wie die D/A-Ausgänge an die Ausgangs-Pins geroutet werden.

Die Design-Files für das PSoC sind kostenlos über die Elektor-Website zu diesem Artikel erhältlich [3]. Cypress bietet eine Menge Tools und Software, um das Innenleben eines PSoCs passend zu konfigurieren. Wem das zu viel ist, oder wer es einfach bequemer haben möchte, der kann über Elektor [3] auch ein fertig programmiertes PSoC beziehen.

## Aufbau

Die Elektronik wird auf der Platine (Bild 7) aufgebaut. Dank Hybrid-Bestückung kommen

bedrahtete Bauteile und SMDs vor. Die SMDs bestückt man selbstverständlich zuerst. Sowohl das PSoC als auch das LM324 haben einen ausreichenden Pin-Abstand. Man kann sie noch gut von Hand löten, wenn man sich Mühe gibt. Hierzu gibt es viele Methoden, die auch schon in Elektor beschrieben wurden.

Die Elkos sind radiale bedrahtete Ausführungen, bei denen man lediglich auf korrekte Polung achten muss.

Für die IN-9-Röhren empfehlen sich Sockel, um Brüche der Anschlüsse am „Murphy Spot“ zu vermeiden, wo die Drähte den Glaskolben verlassen. Wenn die Röhren aber gut an ihrem Bestimmungsort fixiert sind, dann kann

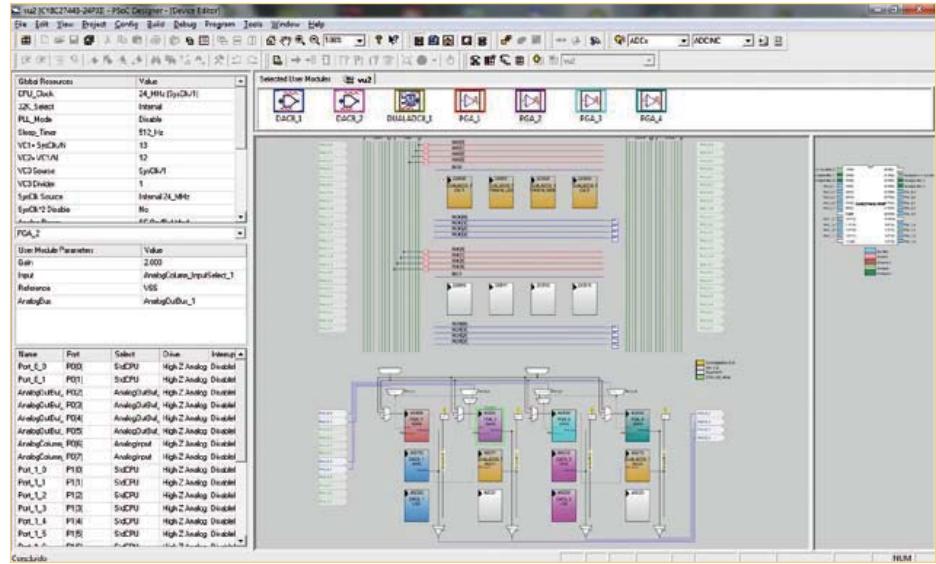


Bild 5. Ein PSoC entspricht im Prinzip einem High-Tech-Lego-Baukasten: Man hat die Elemente - was daraus wird, liegt am Konstrukteur. Man probiert aus und optimiert, bis die gewünschte Lösung stimmt. Das Bild zeigt, wie man ein PSoC als Kern eines Nixie-VU-Meters programmiert.

man die 40 mm langen Drähte auch einfach mit Litzen verlängern, um sie mit den Schraubklemmen V1/V2 der Platine zu verbinden. „K“ entspricht der Kathode und „A“ der Anode. Man sollte immer im Hinterkopf

behalten, dass hier eine recht hohe Spannung anliegt, was entsprechende Isolierung und Vorsicht bedingt. Übrigens gibt es auch ein kurzes Video auf dem Elektor YouTube-Kanal „ElektorIM“

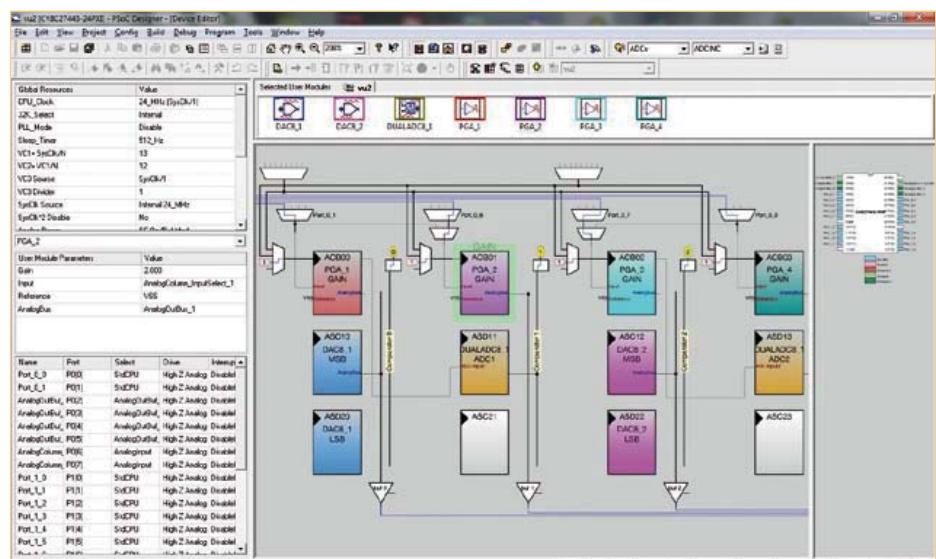
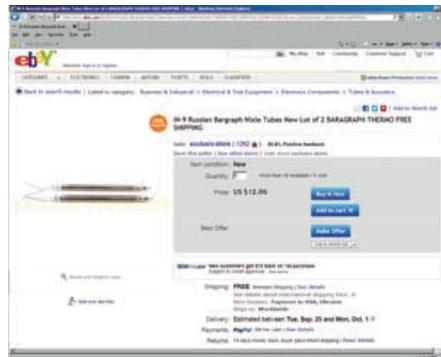


Bild 6. Ein näherer aber abstrakter Blick auf die Verstärker-Module des PSoCs.

[4], in dem das Nixie-VU-Meter in Aktion zu sehen ist.

Ein wichtiger Hinweis: **Bauen Sie diese Schaltung nur dann, wenn Sie sich mit hohen Spannungen auskennen. Dies gilt besonders auch für das Anzapfen von Audiosignalen in Röhrenverstärkern.**

(110744)



## Weblinks

- [1] Nixie-Thermo-/Hygrometer, Elektor Juni 2012, [www.elektor.de/110321](http://www.elektor.de/110321)
- [2] Sputnik-Zeitmaschine, Elektor Januar 2007, [www.elektor.de/050018](http://www.elektor.de/050018)
- [3] [www.elektor.de/110744](http://www.elektor.de/110744)
- [4] [www.youtube.com/user/elektorim?feature=results\\_main](http://www.youtube.com/user/elektorim?feature=results_main)

## Stückliste

### Widerstände:

(alle Widerstände 0805)

R1,R2,R5,R8 = 1 k

R3,R6 = 100 k

R4,R7 = 470  $\Omega$

R9 = 2k2

R10 = 220 k

R11 = 10 k

R12 = 33 k

R13 = 47  $\Omega$

P1,P2 = 220  $\Omega$  Trimmpot, liegend

P3 = 1 k Trimmpot, liegend

### Kondensatoren:

C1,C4 = 470 n (0805)

C2,C11,C17 = 10  $\mu$ /16 V, radial RM 2,5 mm

C3,C5,C6,C7,C8,C9,C10,C13,C14 = 100 n (0805)

C12 = 1  $\mu$ /250V (1825)

C15 = 2n2 (0805)

C16 = 470  $\mu$ /25 V, radial RM 5 mm

IC3 = LM324ADT (SOIC-14)

IC4 = NE555DT (SOIC-8)

IC5,IC6 = 78L05

### Spulen:

L1 = 100  $\mu$ , 860 mA, Multicomp  
MCSDRH73B-101MHF

### Halbleiter:

D1,D2,D3,D4 = BAT46WH

D5 = BV26

D6 = BZD27C5V1P, Z-Diode 5V1

T1,T2 = MMBTA42

T3 = IRF640NSPBF

T4 = BC847

IC1 = CY8C27443-24SXI, Elektor # 110744-41\*

IC2 = MCP101-475HI/ZU

### Außerdem:

K1 = 3,5 mm Klinkenbuchse, stereo, für Platinenmontage (Lumberg 1503 09)

K2 = 5-pol. SIL-Pfostenstecker, RM 0,1" (2,54 mm)

K3,V1,V2 = 2-pol. Schraubklemme für Platinenmontage, RM 5 mm

2x Nixie-Röhre IN-9 Nixie  
Platine # 110744-1\*

\* erhältlich via [www.elektor.de/110744](http://www.elektor.de/110744)

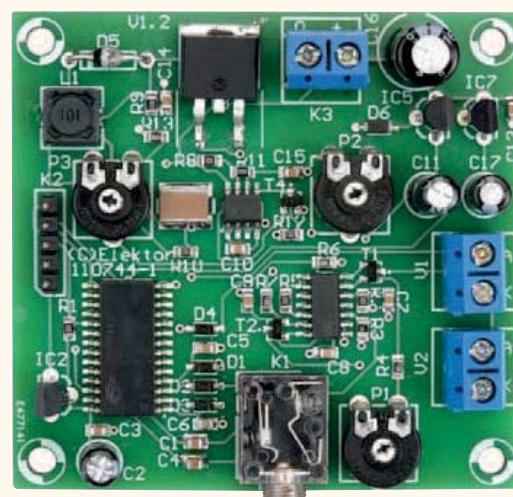
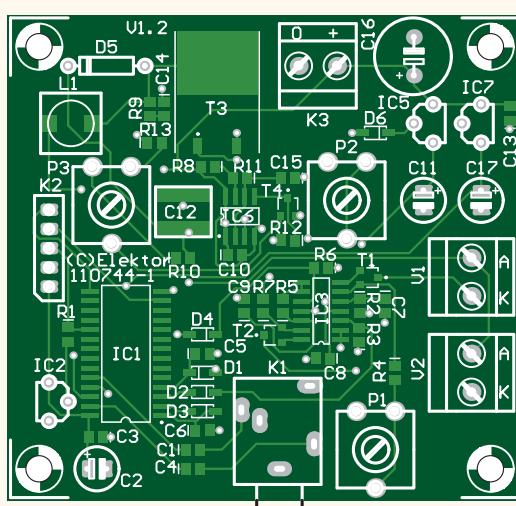


Bild 7. Die vom Elektor-Labor entwickelte Platine ist für eine gemischte Bestückung mit bedrahteten Bauteilen und SMDs gemacht.

# ERHÖHTE USB LEISTUNG

## Optimierte USB Brücken Lösungen

## X-CHIP

Umfangreiche  
Schnittstellen

UART, FIFO, SPI, I<sup>2</sup>C, FT1248

Erweiterte Funktionen

Ladegerät-Erkennung

Low Wirkleistung (8 mA, typisch)

Interne MTP-Speicher

Erweiterbare Taktung; Uhr

Erzeugung und Systemuhr aus

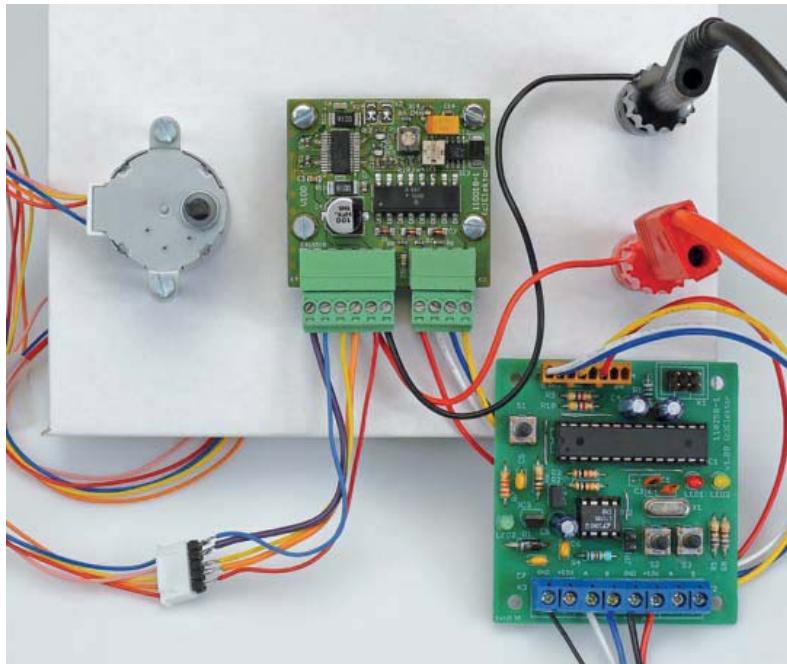
Außergewöhnliche Treiber

Windows, MacOS, Android, and Linux



# Eine Lib für den Bus

## Schnell zur eigenen ElektorBus-Firmware



Vor genau einem Jahr haben wir Software vorgestellt, mit der man eine PC-Steuerung für die eigene Bus-Anwendung schnell und bequem verwirklichen konnte. Das „Rapid-Development“-Konzept weiten wir nun auf die Firmware der Mikrocontroller-Knoten aus. Dank der ElektorBus-C-Bibliothek müssen sich Entwickler nicht mehr mit den einzelnen Bits der Bus-Protokolle herumschlagen. Die Bibliothek ist dank eines eigenen Hardware-Layers sogar unabhängig vom benutzten Controller. Das eröffnet interessante Perspektiven, die noch über den Bus hinausgehen.

Von Jens Nickel

Bei kniffligen Projekten kann es schon ein erster Erfolg sein, wenn man zu Software kommt, die schlicht und einfach tut was sie soll. So war es auch beim ElektorBus. Im Zuge der Bus-Artikel-Serie [1] mussten zuerst einmal die Protokolle entwickelt werden, dabei haben wir in die Controller meist nur Firmware geladen, mit denen wir die neuen Funktionen testen konnten.

Mit der Zeit wachsen aber die Ansprüche. Für verschiedene Typen von Bus-Teilnehmern (Experimental-Knoten mit Fotosensor oder Relais, Installationsplatine) war jeweils eine eigene Firmware zu entwickeln. Das Grundgerüst blieb zwar immer gleich; doch musste man erst mühselig herausfinden, an welcher Stelle nun der Code anzupassen war, um etwa einen Zahlenwert herauszuschicken statt zu empfangen. Und um ein intensives Studieren der Protokoll-Specs kommt man nicht herum, wenn man hartcodiert die Bytes einer Message festlegen muss, die etwa ein Relais anziehen oder abfallen lassen.

### Viele Vorteile

In genau solchen Fällen kann eine Bibliothek eine wertvolle Hilfe sein. Sie verbirgt das immer gleich bleibende Grundgerüst der Bus-

Firmware hinter ein paar Funktionsaufrufen, so dass der eigentliche Anwendungscode kurz und übersichtlich bleibt. Sie berechnet selbsttätig die nötigen Bytes, die man über den Bus schicken muss, so dass sich der Anwendungsentwickler nicht mehr mit den einzelnen Protokoll-Bits herumschlagen muss. Und schließlich macht sie den Anwendungscode auch unabhängig von der Hardware – also vom eingesetzten Busknoten-Board und dem verwendeten Mikrocontroller. Denn immer wieder haben wir Anfragen nach anderen möglichen Controllern (etwa PICs) bekommen. Die Sprache C besitzt glücklicherweise eine genormte Syntax, und die Compiler für die verschiedensten Chip-Typen halten sich daran. So kann man tatsächlich plattformunabhängigen Anwendungscode (Bild 1) schreiben, den man von einem Board auf ein anderes mitnehmen kann.

### Rufen wir die Library

Doch genug der Theorie! Mit folgenden Codezeilen

```
TransmitValue(OwnAddress, MasterAddress, 1, 0,
TestLEDStatus);
TransmitValue(OwnAddress, MasterAddress, 2, 0,
SensorValueRaw);
```

## Elektor-Produkte und Support

- Experimental-Knoten (Platine 110258-1 oder 3er-Set Platinen 110258-1C3)
- USB/RS485-Konverter (fertig aufgebaut und getestet 110258-91)
- Schrittmotor-Treiber (Platine 110018-1)

- Gratis Software-Download

Alle Produkte und Downloads sind über die Website zu diesem Artikel erhältlich: [www.elektor.de/120582](http://www.elektor.de/120582)



stellen wir eine ElektorBus-Nachricht zusammen, die zwei Zahlenwerte (solche und andere Informationseinheiten werden *Parts* genannt) auf zwei verschiedenen *Channels* transportiert, einmal den Status der TestLED und einmal einen Messwert (im Bereich von -1023 bis 1023).

Mit

```
SendParts();
```

versenden wir diese Parts anschließend innerhalb einer Message über den Bus. Welche Bytes mit welchen gesetzten Bits an welcher Stelle unserer 16-Byte-Nachricht stehen müssen, braucht uns nicht mehr zu interessieren.

Wenn Ihnen diese Funktionsaufrufe bekannt vorkommen, dann haben Sie unsere ElektorBus-Serie aufmerksam verfolgt. Im November 2011 hatten wir ja eine Javascript-Bibliothek vorgestellt, die man beim Entwickeln einer eigenen Bus-Steuerung nutzen kann [2]. Wir haben damals bewusst auf ein Bus-User-Interface gesetzt, das auf HTML und Javascript basiert, und daher ohne Änderung sowohl auf dem PC als auch einem Android-Gerät zu verwenden ist.

Soweit es ging, habe ich mich bei den Funktionen der neuen Library tatsächlich an der Javascript-Library JSBus orientiert. Das ist bei einigen Features recht gut gelungen, da Javascript eine C-ähnliche Syntax nutzt.

Allerdings musste ich auch auf den erheblich kleineren Speicher der Mikrocontroller Rücksicht nehmen. In der C-Library nicht implementiert ist zum Beispiel die Möglichkeit, Parts für verschiedene Empfänger zu sammeln, die dann automatisch zu mehreren Messages codiert werden. Vielmehr lassen sich hier vor dem Abschicken einer Message nur bis zu vier Parts zusammenstellen, die von der Bibliothek in einem einfachen Sendespeicher (Array) zwischengespeichert werden.

### Jetzt ruft sie uns zurück

Einen ähnlichen Speicher gibt es auch für empfangene Parts, er wird von der Bibliothek gefüllt, wenn eine Nachricht empfangen wurde, die für den Knoten bestimmt ist. Danach ruft die Bibliothek für jeden der Parts einmal die Funktion:

```
void ProcessPart(struct Part part)
{
...
}
```

auf. Diese muss zwingend im Main-File des Bus-Quellcodes definiert sein, das nun komplett für die Main-Funktion und die eigentliche Anwendung reserviert ist. Die Funktion ProcessPart(...) kann der Entwickler nun mit eigenem Code füllen, welcher auf die empfangenen Parts (etwa Sollwerte für einen Aktor oder eine andere Anweisung des Bus-Masters) reagiert. Auf die einzelnen Eigenschaften des Parts (zum Beispiel den übertragenen Zahlenwert) kann der Entwickler leicht zugreifen mit einem

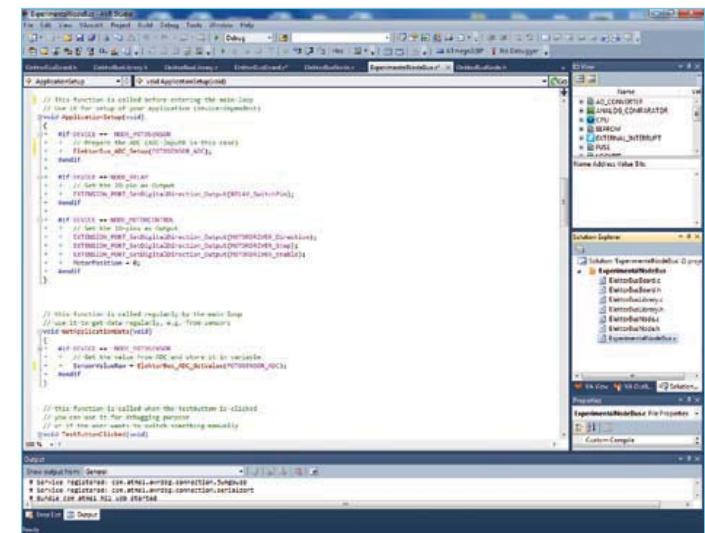


Bild 1. Eine ElektorBus-Anwendung kann ab jetzt unabhängig vom Knoten-Board und dem verwendeten Mikrocontroller entwickelt werden (hier in AVR Studio 5).

Ausdruck wie part.Numvalue (siehe den Kasten „Parts“). Ein Beispiel für solchen Anwendungs-Code wäre:

```
void ProcessPart(struct Part part)
{
    if ((part.Channel == 0) && (part.Parttype ==
        PARTTYPE_VALUE2))
    {
        SwitchTestLED(part.Numvalue);
        SwitchRelay(part.Numvalue);
    }
}
```

Abhängig vom empfangenen Zahlenwert (sinnvollerweise hier nur 0 oder 1) wird die TestLED auf dem Board aus- oder angeschaltet. Auch ein Relais wird zum Abfallen oder Anziehen gebracht. Der Unterschied zwischen den Funktionen SwitchTestLED(...) und SwitchRelay(...) ist nur, dass die erstere von der Bibliothek zur Verfügung gestellt wird (jedes ElektorBus-Board sollte über eine TestLED verfügen). Die zweite Funktion ist dagegen applikations-spezifisch, diese muss der Anwendungsentwickler komplett selbst schreiben (hierzu später mehr).

Es gibt noch weitere (ereignis-getriggerte) Funktionen, die von der Bibliothek aus ausgerufen werden. Für jede dieser Funktionen muss es eine Definition im Main-File geben (oder einem anderen Projektfile, das die ElektorBus-C-Library einbindet). Der Funktionsrumpf kann jeweils mit eigenem Code gefüllt werden, er muss aber nicht.

## Listing 1: Die Hauptschleife jeder ElektorBus-Firmware.

```
int main(void)
{
    ElektorBusLibrarySetup();
    ApplicationSetup();

    while(1)
    {
        GetApplicationData();
        ElektorBusEngine();
    }
}
```

die Adresse) initialisiert, dann folgt eine Endlosschleife. In dieser Schleife hat der Knoten erst Gelegenheit, applikationsspezifische Aufgaben auszuführen, also zum Beispiel Sensorwerte über den ADC einzulesen. Dann kommt die eigentliche *Bus-Engine*. Sie reagiert zuerst auf mittlerweile eingetroffene Nachrichten (auch vom Bus-Scheduler). Falls der Knoten *scheduled* ist und der Scheduler uns zum Senden auffordert, wird das *SendFlag* zum sofortigen Senden einer Nachricht gesetzt. Ansonsten sendet der Knoten als *FreeBusNode* lediglich in der *FreeBusPhase*, aber nur, falls es etwas zum Senden gibt (*SendEventFlag* == TRUE). Auch der Algorithmus, der auf die Bestätigung der Nachricht vom Empfänger wartet und gegebenenfalls das Versenden wiederholt (in der *FreeBusPhase* können ja Kollisionen auftreten) ist der gleiche geblieben (zu den Begriffen siehe [3]).

Das alles ist nun in der Bibliothek gekapselt, der Code ist für den Anwendungsentwickler nicht von Belang. Er sollte nur wissen, dass lediglich das *SendEventFlag* gesetzt wird, wenn er die Funktion *SendParts()* aufruft, die Nachricht wird dann nicht sofort versendet, sondern erst, wenn die nächste *FreeBusPhase* kommt. Für regelmäßig abgefragte Knoten (*scheduled*) wurde ein anderer Mechanismus eingebaut. Wenn die Nachricht des Schedulers eintrifft, dass der Knoten mit dem Senden an der Reihe ist, ruft die Bibliothek zuerst die Funktion

```
void PreparePartsForScheduledMessage()
{
    ...
}
```

auf, deren Definition ebenfalls im Anwendungscode stehen muss. Hier kann der Knoten die passenden Parts (etwa Sensorwerte) nun ganz aktuell zusammenstellen:

```
void PreparePartsForScheduledMessage()
{
    TransmitValue(OwnAddress, MasterAddress, 0, 0,
    SensorValueRaw);
}
```

Ein Aufruf von *SendParts()* ist hier nicht nötig.

Da die Bibliothek die immer wiederkehrenden Teile der Firmware enthält, kann die Hauptschleife schön kurz sein, wie man in **Listing 1** sieht. Die Namen der Funktionsaufrufe sprechen für sich. Die Funktionen *ApplicationSetup()* und *GetApplicationData()* muss der Anwendungsentwickler implementieren; hier kann man zum Beispiel einen ADC initialisieren und dann regelmäßig Messwerte einlesen.

## Hardware-Funktionen

Zum Initialisieren des ADCs können wir schreiben:

```
void ApplicationSetup()
```

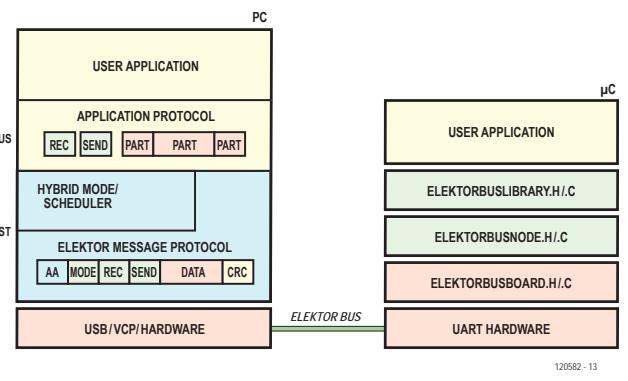


Bild 2. Software für das „Rapid Application Development“. Die ElektorBus-C-Library ist gleichzeitig für beide Protokoll-Schichten (Message- und ApplicationProtocol) zuständig. Dazu kommt ein Hardware-Layer.

Die folgende Funktion wird immer dann aufgerufen, wenn der Anwender den TestButton-Taster betätigt:

```
void TestButtonClicked(void)
{
    ToggleTestLED();
}
```

In diesem Beispiel lassen wir jedes Mal die TestLED toggeln (eine entsprechende Funktion wird ebenfalls von der Library zur Verfügung gestellt). Mit wenigen Zeilen Code haben wir so ein einfaches Debugging-Feature realisiert, mit dem festgestellt werden kann, ob die Software auf dem Knoten zumindest noch läuft. Eine Übersicht der bisher implementierten Funktionen der Bibliothek findet man im Kasten („Wichtige Funktionen“).

## Die Bus-Maschine

Nichts geändert hat sich am grundsätzlichen Ablauf der Firmware: Zuerst werden die Hardware und die Knoten-Parameter (wie

```
{
    ElektorBus_ADC_Setup();
}
```

Die Funktion ElektorBus\_ADC\_Setup() initialisiert den ADC (ADC-Channel 0). Die dahinter stehenden Befehle (bestimmte Werte müssen in bestimmte Register des Mikrocontrollers geschrieben werden) sind natürlich controllerabhängig. Solche Funktionen wurden in einer eigenen kleinen Bibliothek untergebracht, die in den Files „ElektorBusBoard.h“ und „ElektorBusBoard.c“ steckt. Der Sourcecode der eigentlichen ElektorBus-C-Library ist dagegen in den Files „ElektorBusLibrary.h“ und „ElektorBusLibrary.c“ zu finden (siehe hierzu **Bild 2**).

Warum diese Trennung? Nun, falls sich ein Leser entschließt, seine Anwendung von einem Board auf ein anderes Board (entweder mit einem anderen Controller oder einer anderen Pinbeschaltung) zu portieren, kann er nicht nur das Hauptfile mit dem Anwendungs-Code, sondern auch die Library-Files „ElektorBusLibrary.h/.c“ mitnehmen. Er muss nur die Funktionen in „ElektorBusBoard.c“ neu implementieren (vermutlich kann er sogar darauf hoffen, dass ein anderer Entwickler diese Aufgabe für das jeweilige Board schon erledigt hat). Dazu kommen sogenannte #define-Ausdrücke, die ein sehr schönes Feature von C sind. Im File „ElektorBusBoard.h“ sind eine Reihe solcher Ausdrücke zu finden wie zum Beispiel

```
#define LED_PORT_OUT PORTD
#define TESTLED_PIN 4
```

Vor dem eigentlichen Compilieren des Quellcodes sucht ein Präprozessor im Code zum Beispiel nach dem Ausdruck „TESTLED\_PIN“ und ersetzt jedes Vorkommen durch die Zahl „4“. Verwendet man als Entwickler immer schön die definierten und abstrakten Ausdrücke, dann kann der Quellcode der Bus-Anwendung (und der Bus-Bibliothek) immer gleich bleiben (ganz egal an welchen Portpin zum Beispiel die TestLED angeschlossen ist). Portiert man die Anwendung auf ein anderes Board, so sind lediglich die #define-Ausdrücke anzupassen.

### Board-Specs

Eine solche Bus-Hardware-Bibliothek macht vor allem dann Sinn, wenn man sich auf gewisse Grundfunktionen einigt, die jedes ElektorBus-Board mitbringen muss. Neben der schon erwähnten TestLED und dem TestButton schlage ich vor:

- bis zu 7 weitere LEDs an einer gemeinsamen, über ein Register zugänglichen Pingruppe namens LED\_PORT (LED1, LED2, ...; LED0 ist die TestLED),
- bis zu 7 weitere Taster an einem BUTTON\_PORT (Button1, Button2, ...; Button0 ist der TestButton),
- Eine Pingruppe, welche die RX/TX-Pins sowie zwei I/O-Pins für die DriverEnable- und ReceiverDisable-Funktion der RS485-Kommunikation enthält (ELEKTORBUS\_PORT),
- eine Pingruppe, deren Pins zum Beispiel auf einen Steckverbin-

### Listing 2. Relais schalten mit hardware-unabhängigem Code.

```
void SwitchRelay(unsigned char relayStatus)
{
    if (relayStatus == 0)
    {
        // Set pin to low
        EXTENSION_PORT_SetDigital_Low(EXTENSION_DIGITAL0_PIN);
    }
    else
    {
        // Set pin to high
        EXTENSION_PORT_SetDigital_High(EXTENSION_DIGITAL0_PIN);
    }
}
```

der zum Anschluss weiterer Hardware geführt sind (EXTENSION\_PORT). Dieser Port kann bis zu acht digitale I/O-Pins oder wahlweise bis zu acht analoge Eingänge zusammenfassen.

Für alle diese Features sind in ElektorBusBoard.h entsprechende hardwareunabhängige Ausdrücke definiert. Im Anwendungscode, etwa beim Schalten des Relais in der anwenderspezifischen Funktion SwitchRelay(), sollte man tunlichst diese verwenden, siehe **Listing 2**.

Beim dort zu findenden Ausdruck

```
EXTENSION_PORT_SetDigital_Low(EXTENSION_DIGITAL0_PIN)
```

handelt es sich um ein sogenanntes Makro, das ebenfalls in „ElektorBusLibrary.h“ definiert ist. Es bewirkt, dass der Digital-I/O-Pin Nr.0 des EXTENSION\_PORTS auf Low gesetzt wird (und das Relais abfällt). Da man vermutlich das Relais nicht immer am Pin 0 des EXTENSION\_PORTS (beim Experimental-Knoten ist das PC0) anschließen möchte, sollte man noch im Main-File einen eigenen #define-Ausdruck definieren:

```
#define RELAY_SwitchPin EXTENSION_DIGITAL0_PIN
```

Dann kann man schreiben:

```
EXTENSION_PORT_SetDigital_Low(RELAY_SwitchPin);
```

Falls das Relais während der weiteren Entwicklung an einen anderen Portpin wandert, ist nur der obige #define-Ausdruck zu ändern.

### Knoten-File

Ganz oben im Text haben wir in der Codezeile

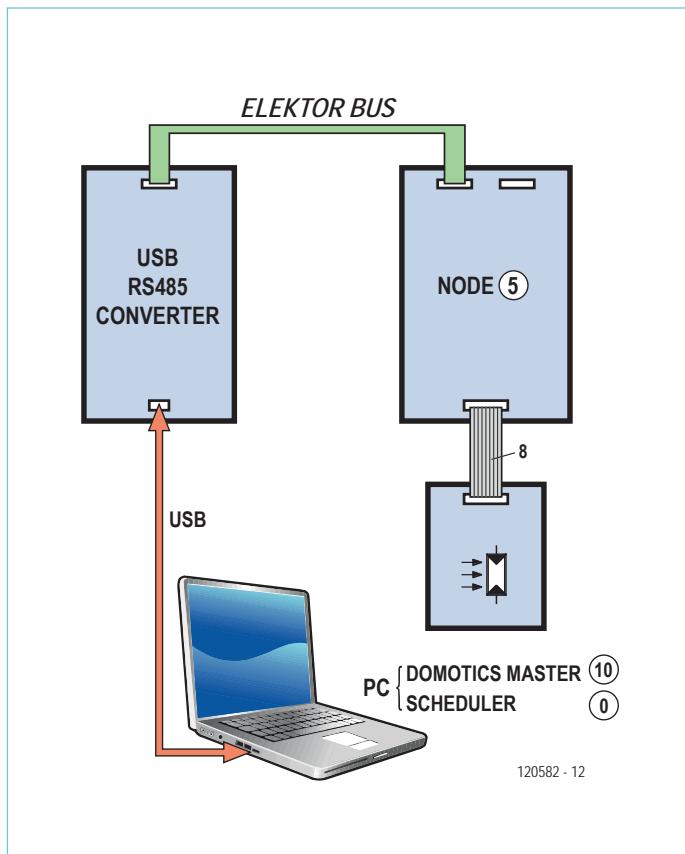


Bild 3. Für unser Anwendungsbeispiel genügen ein RS485/USB-Konverter und ein Experimental-Knoten, der mit einem ATmega328 ausgerüstet werden muss.

```
TransmitValue(OwnAddress, MasterAddress, 1, 0,
TestLEDStatus);
```

über die Variable `OwnAddress` auf die Adresse des Knotens zugegriffen. Auch die Bus-Library muss natürlich die Adresse des Knotens kennen, damit sie entscheiden kann, welche Nachrichten für den Knoten bestimmt sind. Die Knoten-Parameter kann die Bus-Bibliothek beim Start der Software über Funktionen ermitteln, die in einem weiteren File-Paar untergebracht sind: „ElektorBusNode.h/c“. Im Moment ist die Knotenadresse in der entsprechenden Funktion hartcodiert eingetragen:

```
unsigned char GetNodeAddress()
{
    return 5;
}
```

Während der weiteren Entwicklung dieses Projekts soll ja aber noch eine dynamische Adressierung der Knoten hinzukommen.

### Beispielsoftware

Zu diesem Bus-Artikel gibt es natürlich wieder Beispielsoftware [4]. Das Projekt für AVR Studio 5 enthält alle oben genannten Files, nebst einer Beispieldatenbank. Genau genommen sind es sogar drei, die gerade gewünschte kann man über einen `#define`-Ausdruck auswählen, den man ganz oben im File „ElektorBusNode.h“ findet:

```
#define DEVICE NODE_FOTOSENSOR
```

Dem Knoten wird dann beim Start des Programms eine entsprechende Adresse zugewiesen (5, 6 oder 7), außerdem wird

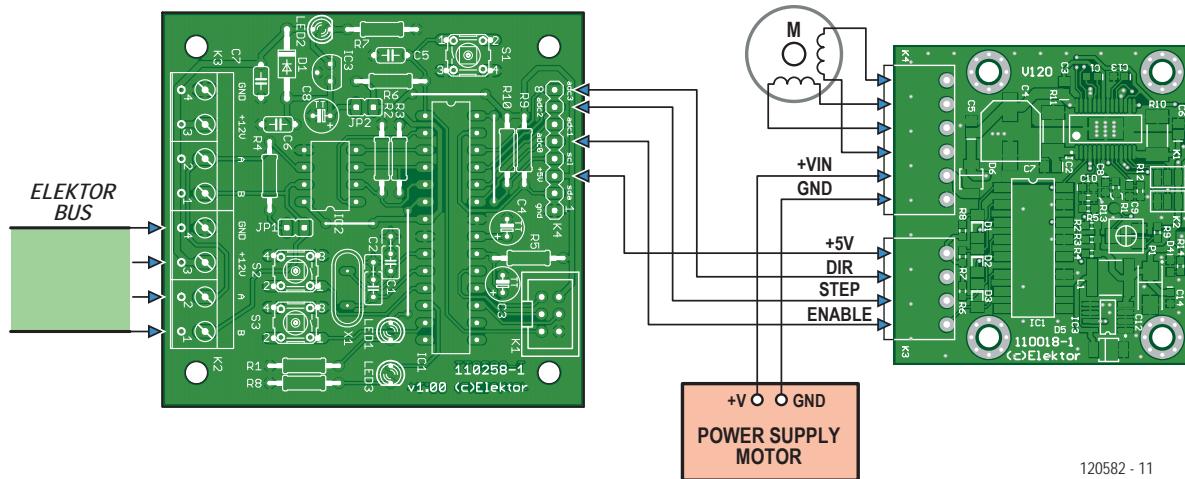
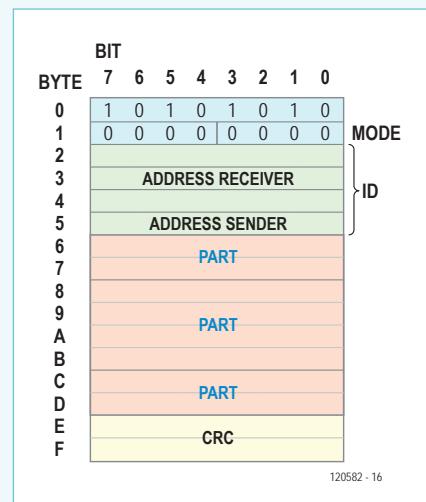


Bild 4. Für das dritte Applikationsbeispiel verbindet man den Experimental-Knoten mit der Schrittmotortreiber-Platine aus Elektor Juni 2012. Wir haben einen kleinen Schrittmotor von Ming Jong genutzt [9].

## Parts

Innerhalb der acht Datenbytes einer ElektorBus-Nachricht können gemäß dem *ApplicationProtocol* bis zu vier Parts (Informationseinheiten) übertragen werden, zum Beispiel Messwerte, Sollwerte, Grenzwertmeldungen oder Kommandos zum Wechsel der Einheit und Skalierung. In der ElektorBus-C-Library wird ein **Part** durch eine Datenstruktur (gekennzeichnet mit dem Schlüsselwort `struct`) abgebildet. Für jeden empfangenen Part ruft die Library die Funktion `void ProcessPart(struct Part part)` auf und übergibt ihr dabei eine solche Datenstruktur namens `part`. Der Anwendungsentwickler hat mit einem Ausdruck wie zu Beispiel `part.Numvalue` Zugriff auf alle Eigenschaften des empfangenen Parts:

Valid	Checksumme ok? (noch nicht implementiert)
Sender	Senderadresse
Receiver	Receiveradresse
Channel	Channel-Nummer
Setflag	Soll- oder Messwert?
Ackflag	AcknowledgeMessage oder Originalnachricht (Flag auf Application-Ebene)
Mode	Modebyte der Nachricht (mit Acknowledge-Flags auf Message-Ebene)
Parttype	Typ des Parts, hierzu sind Konstanten definiert: PARTTYPE_VALUE2, PARTTYPE_VALUE4, PARTTYPE_VALUEFLOAT, PARTTYPE_LIMIT, PARTTYPE_SCALE, PARTTYPE_INTERVAL.
Numvalue	übertragener numerischer Wert (z.B. -1023..1023 bei PARTTYPE_VALUE2)
Limit	0 = alles ok, 1 = Grenzwert unterschritten, 2 = Grenzwert überschritten
Quantity	Physikalische Größe (0..127, siehe [3])
Unit	Einheit (0..3, siehe [3])
Scale	Zehnerpotenz für Skalierung (-15..+15)
Interval	Intervall-Skaleneinheit für Meldungen des Sensors (siehe [3])
Preset	reserviert



festgelegt, ob der Knoten *scheduled* ist oder nicht. Natürlich kann man die Adressen im File „ElektorBusNode.c“ noch an eigene Bedürfnisse anpassen.

Die hardwareabhängigen Files „ElektorBusBoard.h“ und „ElektorBusBoard.c“ sind auf das kleine Experimental-Knoten-Board zugeschnitten. Allerdings passt das HEX-File nicht mehr in den Flash-Speicher eines ATmega88, dieser ist durch einen pinkompatiblen (und nur unwesentlich teureren) ATmega328 zu ersetzen. Da die Bus-Library die Basis für weitere Entwicklungen sein soll, stellt sich 32 KB Flash im Controller als weitere Anforderung an ein ElektorBus-Board heraus (noch besser wären 64 KB oder mehr). Für das Beispiel genügt ein einziger Experimental-Knoten, daneben benötigen wir noch einen RS485/USB-Konverter (siehe **Bild 3**).

### Ein bisschen Motorsteuerung

Damit man die Software auch ausprobieren kann, muss man entweder einen Fotowiderstand oder ein Relais an den EXTENSION\_PORT des Experimental-Knotens (Stiftleiste K4 mit 5 V, GND, PC0...

PC6) anschließen. Die Schaltung des Fotosensors (ein Poti tut es auch) und der Relais-Erweiterung findet man in der 8. Folge der Bus-Serie [5]. Als Bonbon haben wir noch eine dritte Anwendung integriert, nämlich eine einfache Ansteuerung eines Schrittmotors. Hierzu muss man die Schrittmotortreiber-Platine von Elektor [6][7] wie in **Bild 4** gezeigt an den Experimental-Knoten anschließen. Dies ist übrigens ein kleiner Vorgriff auf einen kommenden ElektorBus-Artikel, in dem wir eine bus-kompatible Schrittmotortreiber-Platine vorstellen werden.

Im Download [4] findet sich auch ein UIBus-Ordner mit einem ganz einfachen HTML-File zur PC-Steuerung des Ganzen (einfach auf den Desktop ziehen und den ElektorBusBrowser.exe [8] starten). Zuerst werfen wir den Scheduler an mit dem entsprechenden HTML-Button (**Bild 5**). Wenn die Originalfirmware in den ATmega328 geladen ist, dann müsste man Sensorwerte sehen. Wenn man den TestButton auf dem Board betätigt, sollte außerdem die rote TestLED auf dem Board toggeln. Der Status der LED wird auch im Radiobutton auf dem PC angezeigt.



Bild 5. Ein ganz einfaches HTML-User-Interface zum Testen unserer drei Anwendungsbeispiele.

Wenn NODE\_RELAY in der Firmware ausgewählt und ein Relais angeschlossen ist, kann man das Relais anziehen und abfallen lassen. Das geht dann auch manuell am Knoten mit dem TestButton. Lautet der Ausdruck in der Firmware dagegen #define DEVICE NODE\_MOTORCONTROL, dann kann man den Schritt-Motor vom PC aus 1000 Schritte vor- und zurückfahren lassen.

## Ausblick

Es gibt noch sehr viel zu tun, denn die Bus-Bibliothek liegt bisher nur in einer ersten Version vor und enthält nur die nötigsten Funktionen. Unter anderem sollten noch Intervall-Funktionen (regelmäßiges Aussenden von Sensorwerten vom Knoten aus) und im Zuge dessen

auch Funktionen wie eine Initialisierung eines Timers implementiert werden.

Dann haben wir aber schon eine nette, kleine Bibliothek, mit der wir diverse Controller-Funktionen recht unabhängig vom verwendeten Chip-Typ steuern können. Dass der Ansatz prinzipiell funktioniert, haben wir schon mit einem ATxmega256 getestet (der sich auf einem kommenden Elektor-Board befindet). Das eröffnet die Perspektive, einmal zu einer Art „Elektor Firmware Library“ zu kommen, die man auch für andere Entwicklungen nutzen kann (also nicht nur für den ElektorBus). Eine hochinteressante Vorstellung: Dass Elektor-Projekte anderer Elektroniker ganz einfach auf den Lieblings-Controller portiert werden könnten...

Über all das und mehr werden Sie aktuell auf dem Laufenden gehalten unter [www.elektor-projects.com](http://www.elektor-projects.com).

(120582)

## Weblinks

- [1] [www.elektor.de/elektorbus](http://www.elektor.de/elektorbus)
- [2] [www.elektor.de/110517](http://www.elektor.de/110517)
- [3] [www.elektor.com/Uploads/2012/2/ElektorBusReference.pdf](http://www.elektor.com/Uploads/2012/2/ElektorBusReference.pdf)
- [4] [www.elektor.de/120582](http://www.elektor.de/120582)
- [5] [www.elektor.de/110428](http://www.elektor.de/110428)
- [6] [www.elektor.de/110018](http://www.elektor.de/110018)
- [7] <http://rch.lt/schrittmotor>
- [8] [www.elektor.de/110708](http://www.elektor.de/110708)
- [9] [www.mingjong.net/index.php?action=prod\\_detail&p\\_id=88&lang=en](http://www.mingjong.net/index.php?action=prod_detail&p_id=88&lang=en)

## Wichtige Funktionen der ElektorBus-C-Library

### Funktionen und Variablen, die in ElektorBusLibrary.c implementiert sind:

`void SetValue(sender, receiver, channel, mode, setvalue)`

Fügt dem Part-Sendespeicher einen weiteren Part hinzu, der einen Soll-Wert auf einem Sensor/Aktor setzt.

`void TransmitValue(sender, receiver, channel, mode, currentvalue)`

Fügt dem Part-Sendespeicher einen weiteren Part hinzu, der einen Ist-Wert eines Sensors repräsentiert.

`void SendParts()`

Encodiert alle Parts, die im Part-Sendespeicher gespeichert sind, zu einer Message und versendet diese.

`void SwitchTestLED(unsigned char ledStatus)`

Schaltet die TestLED an (ledStatus = 1) oder aus (ledStatus = 0).

`void ToggleTestLED()`

Toggelt die TestLED.

## TestLEDStatus

In dieser Variablen steht der Status der TestLED (an = 1, 0 = aus).

## TestButtonToggleStatus

Bei jeder Betätigung des TestButtons wird diese Variable zwischen 0 und 1 umgeschaltet.

## OwnAddress

Die Adresse des Knotens.

## MasterAddress

Die Adresse des Master-Knotens für die Bus-Steuerung.

## FreeBusPriority

Gibt an, nach wie vielen *FreeBusPhases* der Knoten ein missglücktes Senden wiederholt.

## IsScheduled

Gibt an, ob der Knoten *scheduled* (regelmäßig abgefragt) ist oder nicht (= 1 oder = 0).

## NodeDevice

Eine Zahl, mit welcher verschiedene Bus-Applikationen (Geräte) im gleichen Quellcode unterschieden werden können, z.B. Fotosensor = 1, Relais = 2.

## Funktionen, die im Hauptfile implementiert sein müssen (mit oder ohne Code):

### void ApplicationSetup()

Wird vor Beginn der Buskommunikation aufgerufen, hier lassen sich zum Beispiel der ADC oder angeschlossene Hardware initialisieren.

### void GetApplicationData()

Wird periodisch aufgerufen, hier kann man zum Beispiel angeschlossene Sensoren abfragen.

### void TestButtonClicked()

Wird aufgerufen, wenn der TestButton (Taster) gedrückt wird.

### void ProcessPart(struct Part part)

Nach dem Empfang einer Message wird diese in bis zu vier Parts decodiert. Für jeden Part wird diese Routine einmal aufgerufen. Hier kann die Anwendung zum Beispiel auf Kommandos des Masters reagieren. Über die Variable *part* hat man Zugriff auf alle Eigenschaften des Parts (siehe Kasten „Parts“).

## void PreparePartsForScheduledMessage()

Ein Knoten, der *scheduled* ist, wird vom Scheduler regelmäßig zum Senden aufgefordert. Wenn eine solche Nachricht des Schedulers eintrifft, ruft die Library diese Funktion auf. Hier kann der Anwendungscode aktuell die Parts zusammenstellen, die nun versendet werden sollen, zum Beispiel Messwerte eines Sensors.

## Wichtige Funktionen, Makros und Defines des Hardware-Layers:

**void ElektorBus\_ADC\_Setup(unsigned char ADCchannel)**  
Initialisiert den ADC (Single-Shot, Referenzspannung = Betriebsspannung des Controllers).

**unsigned short ElektorBus\_ADC\_GetValue(unsigned char ADCchannel)**  
Veranlasst den ADC zum Einlesen eines Spannungswertes und gibt diesen zurück.

**EXTENSION\_PORT\_SetDigitalDirection\_Output(Pin)**  
Stellt einen Digitalpin des EXTENSION\_PORTS auf Ausgang. Pin ist die physische Pinnummer innerhalb der Pingruppe des EXTENSION\_PORTS (zum Beispiel Pin = 0 an PORT C => PC0).

**EXTENSION\_PORT\_SetDigitalDirection\_Input(Pin)**  
Stellt einen Digitalpin des EXTENSION\_PORTS auf Eingang.

**EXTENSION\_PORT\_SetDigital\_High(Pin)**  
Setzt einen Digitalpin des EXTENSION\_PORTS auf High.

**EXTENSION\_PORT\_SetDigital\_Low(Pin)**  
Setzt einen Digitalpin des EXTENSION\_PORTS auf Low.

**EXTENSION\_PORT\_Digital\_PinStatus(Pin)**  
Fragt den Status eines Digitalpins am EXTENSION\_PORT ab (==1 High, ==0 Low).

**EXTENSION\_DIGITAL0\_PIN**  
**EXTENSION\_DIGITAL1\_PIN**

...  
Statt Zahlen sollte man für Pin diese Ausdrücke verwenden. Dann kann der Code ganz einfach von einem Board auf ein anderes (mit anderer Pinbeschaltung) portiert werden.

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

## ELV-Hauptkatalog 2013



Jetzt kostenlos anfordern  
[www.elektor.elv.de](http://www.elektor.elv.de)

## Jetzt als Neukunde profitieren



Schnell sein  
und € 5,- sparen\*

Als Neukunde erhalten Sie exklusiv eine **Sofortgutschrift von € 5,-** auf Ihre Bestellung ab einem Warenwert von € 25,-. Einfach bis zum 05.11.2012 bestellen und die Persönliche Bonus-Nummer (PBN) W00021 angeben.\*

\* Die Aktion gilt nur für Neukunden, die bis zum 05.11.2012 für einen Warenwert von mindestens € 25,- aus dem ELV-Sortiment bestellen und die PBN W00021 angeben. Die Gutschrift wird mit dieser Bestellung verrechnet. Die Aktion ist nicht kombinierbar. Fachhändler sind hiervon ausgeschlossen.

968 Seiten Neuheiten, Experten-Tipps, Kundenbewertungen, Projektideen und Technik pur!

Alle Infos unter [www.elektor.elv.de](http://www.elektor.elv.de)

## Digital-Multimeter DM9199 IP 67



- Robustes TrueRMS-Multimeter (IP 67)
- Sicherheitsklasse CAT IV (600 V)/CAT III (1000 V)
- Frequenzmessung bis 100 MHz
- Kapazitätsmessung bis 40 mF
- AC/DC-Strom-/Spannungsmessung
- Funktionen: Temperaturmessung, Data-Hold, Min.-/Max.-Recording, Spitzenwertaufzeichnung



WF-08 70 76

€ 79,-

Das sagen unsere Kunden im Web-Shop

★★★★★ Best.-Nr. WF-08 70 76

**Galoro:** Das Gerät macht einen soliden Eindruck. Die Bedienung ist durchdacht und gut verständlich. Durch den erhöhten Schutzgrad sind auch Messungen unter rauen Bedingungen möglich. Die durchgeführten Messungen mit diesem Gerät sind ausreichend genau und reproduzierbar ... Alles in allem ein gelungenes Gerät.

## USB-PC-Oszilloskop und Funktionsgenerator PC SU 200

NEU

velleman®



WF-10 73 52

€ 189,-

- **5-in-1:** 2-Kanal-Digital-Oszilloskop bis 12 MHz, Funktionsgenerator bis 500 kHz (Sinus 1 MHz), Transientenrecorder, Bode-Plotter und Spektrumanalysator
- Spannungsversorgung über USB-Schnittstelle (USB-Kabel im Lieferumfang enthalten)

## Diamex-USB-ISP-Programmer

inkl. 6-auf-10-Adapter



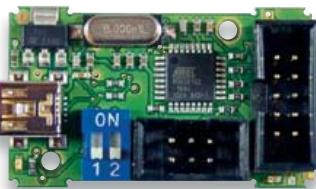
WF-10 69 83

€ 24,95

- Handlicher USB-Programmer
- Unterstützt alte AVRs, die über eine IPS verfügen, auch Arduino
- Spannungsversorgung über USB

Lieferung inkl. SPI-Anschlusskabel, 6-polig, ca. 15 cm Länge.

## Diamex-USB-ISP-Programmer



WF-10 26 65

€ 19,95

- Unterstützung von Classic, Tiny und Mega-Controllern
- Softwareunterstützung: AVR Studio (COM1...COM9; nicht für Version 5.x geeignet), AVRDUDE und Bascom
- Besonders hoher Target-Strom: bis ca. 450 mA bei 5 V und 3,3 V
- LEDs: USB- und Target-LEDs
- Abm. (L x B x H): 72 x 35 x 15 mm.

Lieferumfang: ISP-Programmer, Mini-USB 2.0-Anschlusskabel, SPI-Anschlusskabel, 6-polig, und SPI-Anschlusskabel, 10-polig.

## Electronic-Assembly-Starterkit mit Grafik-LCD EA EVALeDIP240B

ELECTRONIC ASSEMBLY



NEU



WF-10 72 54

€ 245,-

- USB-Schnittstelle
- Evaluationsboard sorgt für kürzeste Entwicklungzeiten
- Display-Farben: Blau-Weiß

### Lieferumfang:

- Display (240 x 128 Pixel) inkl. Touchpanel
- USB-Programmerboard bzw. USB-Evaluationsboard
- DVD mit Treibern für Windows, einem Editor, Compiler sowie Demos
- USB-Kabel
- 5x Interfaceboards für RS-232, RS-485, I²C, SPI und Mikrocontroller RS-232

Weitere Komponenten finden Sie unter [www.electronic-components.elv.de](http://www.electronic-components.elv.de)

## SMD-Sortierboxen, unbestückt, jeweils 10 Stück

NEU



### Größe 1 (B x H x T)

Außenmaß: 15,5 x 23,0 x 27,0 mm  
Innenmaß: 11,5 x 18,3 x 13,9 mm

ab

€ 2,95

### Größe 2 (B x H x T)

Außenmaß: 31,0 x 23,0 x 27,0 mm  
Innenmaß: 27,0 x 18,3 x 13,9 mm

Farbe

Best.-Nr.

Preis

Schwarz Antistatik	WF-04 03 29	€ 3,95
Altweiß	WF-04 03 28	€ 2,95
Gelb	WF-04 03 27	€ 2,95

Schwarz Antistatik	WF-04 03 35	€ 4,95
Altweiß	WF-04 03 34	€ 3,15
Gelb	WF-04 03 33	€ 3,15

### Größe 3 (B x H x T)

Außenmaß: 31,0 x 23,0 x 54,0 mm  
Innenmaß: 27,0 x 18,3 x 40,5 mm

Farbe

Best.-Nr.

Schwarz Antistatik	WF-04 03 41	€ 6,45
Altweiß	WF-04 03 40	€ 3,95
Gelb	WF-04 03 39	€ 3,95

### Größe 4 (B x H x T)

Außenmaß: 62,0 x 23,0 x 54,0 mm  
Innenmaß: 57,8 x 18,3 x 40,5 mm

Farbe

Best.-Nr.

Schwarz Antistatik	WF-04 03 47	€ 8,15
Altweiß	WF-04 03 46	€ 4,15
Gelb	WF-04 03 45	€ 4,15

## SMD-Sortierboxen, bestückt

ELV

Das erweiterbare SMD-Lagersystem von ELV: Unsere SMD-Boxen in unterschiedlichen Größen und Varianten lassen sich beliebig kombinieren und erweitern. Dank des transparenten und sicher schließenden Deckels fallen die winzigen Bauelemente nicht heraus, und Sie haben stets einen guten Überblick über Ihr SMD-Sortiment.

- Beliebig erweiterbar
- Sicher schließender Deckel



NEU

nur

€ 54,95

Widerstands-Sortiment in SMD-Sortierbox, gefüllt

Das Sortiment enthält 74 Werte der Reihe E12 von 1 Ω bis 1 MΩ, zusätzlich 0 Ω.

- Jeweils 50 Stück je Wert = 3700 Widerstände
- Spannung 150 V, Leistung 0,125 W, Toleranz ±5 %, Temperaturkoeffizient ±200 ppm/K
- Sortiert in SMD-Boxen (weiß) mit transparentem Klappdeckel (23 x 15,5 x 27 mm) mit Werteaufdruck
- 10 leere Boxen für eigene Ergänzungen

WF-10 55 67

€ 54,95

Kondensatoren-Sortiment in SMD-Sortierbox, gefüllt

Das Sortiment enthält 38 Werte der Reihe E6 von 1 pF bis 1 μF (zusätzlich 10 pF) mit 50 Stück je Wert.

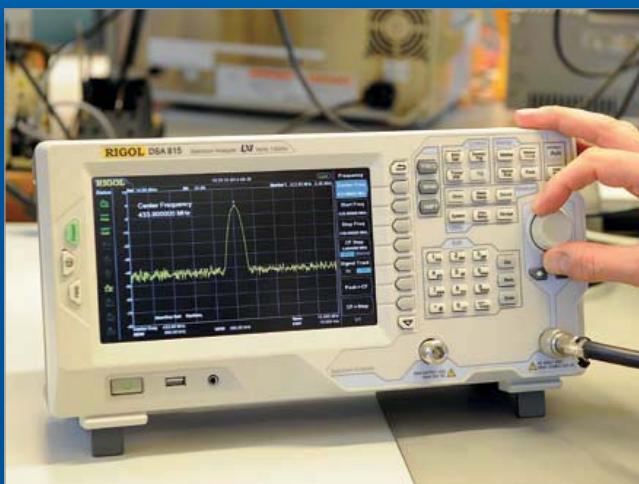
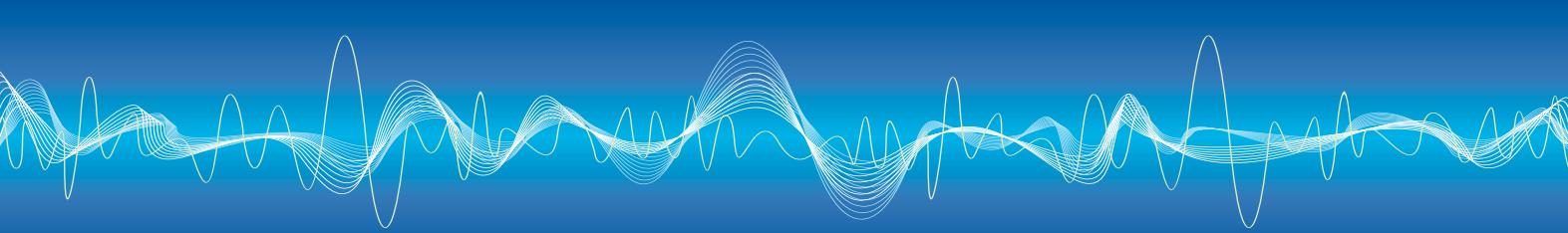
- Jeweils 50 Stück = 1900 Kondensatoren
- Nennspannung 50 V (16 V bei 470 nF, 680 nF, 1 μF; 6,3 V bei 10 μF)
- Sortiert in 38 SMD-Boxen (gelb) mit transparentem Klappdeckel (23 x 15,5 x 27 mm) mit Werteaufdruck
- 4 leere Boxen für eigene Ergänzungen

WF-10 56 46

ELV

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

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



# Spektrum-Analyser Rigol DSA 815-TG

Von Harry Baggen (Redaktion NL)

„Wir haben jetzt einen traumhaften Analyser im Vertriebsprogramm, den Sie sich unbedingt anschauen müssen. Der ist der Konkurrenz um Längen voraus, dabei kostet er weniger als 1500 €.“ Mit diesen Sätzen begann ein Anruf von AR Benelux, Distributor des noch jungen chinesischen Messtechnik-Herstellers Rigol. Alle Details klangen vielversprechend: Ein ausgewachsener Spektrum-Analyser für Frequenzen bis 1,5 GHz mit integriertem Tracking-Generator. Nur knapp 1500 € einschließlich Mehrwertsteuer soll er in Europa beim Importeur kosten. Wenn er hält, was er verspricht, so dachte ich, ist das mehr als preiswert. Die aktuelle Elektor-Ausgabe war gerade in trockenen Tüchern, und ich konnte wieder freier durchatmen. Ich hatte jetzt die Zeit, das angebliche (oder tatsächliche?) Mess- und Preiswunder in Augenschein zu nehmen. Doch bis es so weit war, liefen schon die Vorbereitungen für die nächste Elektor-Ausgabe an, es wurde eng mit dem Termin für die Reservierung des Testgeräts. Hinzu kam, dass Kollege Jan Buiting in Urlaub war, der einzige lizenzierte Funkamateure unter meinen Kollegen vor Ort. Er hätte an einer fachkompetenten Begutachtung dieses Spektrum-Analysers bestimmt gern mitgewirkt, so kenne ich ihn. Endlich stand das Gerät vor uns, dem Kollegen Ton Giesberts und mir. Wir waren etwas unschlüssig, wie wir vorgehen sollten, denn wir gehören der Spezies der „Gleichstrom-Füchse“ an. So nennen HF-Experten die Menschen, die mit Systemen umgehen, in denen die Signale die Frequenzmarke 100 kHz nicht übersteigen. Schließlich hatten wir einen Plan: Einige ausgewählte HF-Systeme, an denen das Elektor-Labor arbeitete, sollten uns als Testobjekte dienen.

## Grundsilder Eindruck

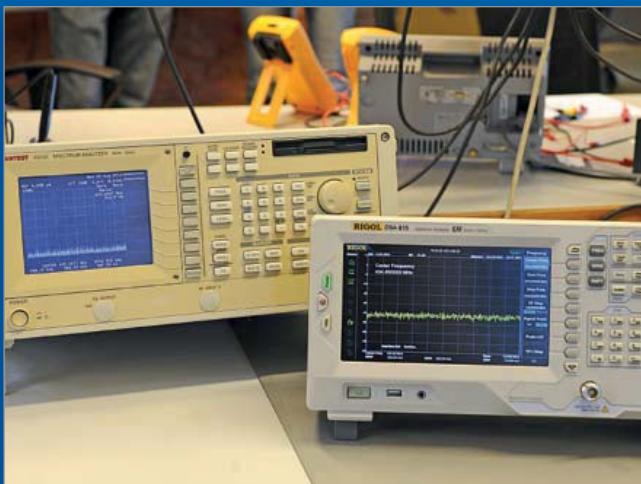
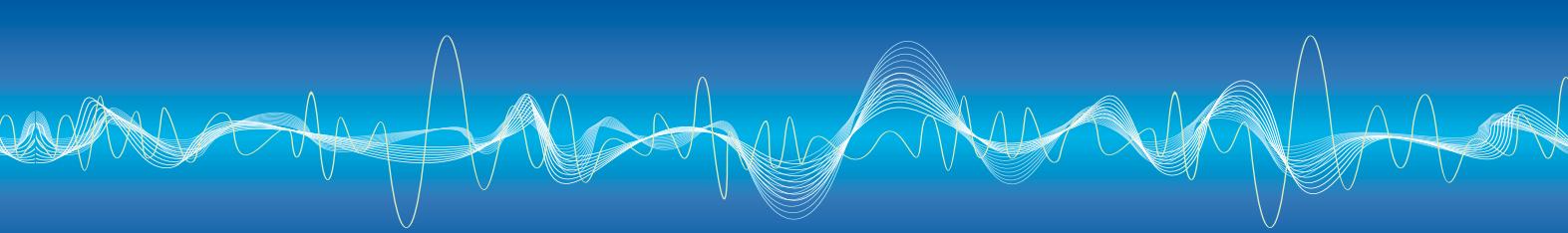
Der DSA 815-TG ist ein Spektrum-Analyser für den Frequenzbereich 9 kHz bis 1,5 GHz. Normalerweise kostet ein Gerät mit vergleichbaren Daten und Eigenschaften etwa das Doppelte, einen Tracking-Generator nicht eingeschlossen. Der Tracking-Generator gehört beim DSA 815-TG zur Standardausstattung. Die Vielseitigkeit dieses Spektrum-Analysers ist überwältigend, und beim genannten Preis dürften nicht nur engagierte Funk-

amateure ins Schwärmen geraten.

Beim Auspacken des Spektrum-Analysers fiel uns zuerst das Gewicht auf. Die 4,25 kg, die er auf die Waage bringt, sind im Vergleich mit digitalen Oszilloskopen gleicher Preisklasse recht stattlich. Die äußere Gestaltung und Verarbeitung ist makellos (hier haben manche anderen chinesischen Produkte Nachholbedarf!), die Farbgebung ist dezent und freundlich. Vorhanden sind zwei USB-Anschlüsse, ein Host und ein Client, so dass sowohl ein USB-Speicherstick als auch ein PC anschließbar sind. Ein Ethernet-Anschluss macht die Bedienung über ein Netzwerk möglich. Nachdem das Gerät eingeschaltet ist, besticht das 800 · 480 Pixel große Breitformat-Display durch seine Bildqualität. Auch bei stark seitlichem Betrachtungswinkel leidet die Erkennbarkeit von Zeichen nur unbedeutend. Die meisten Bedienelemente sind rechts neben dem Display angeordnet, dazu gehören mehrere Tasterfelder sowie ein Drehknopf. Auf dem Display werden rund um das Messfenster diverse Parameter eingeblendet. Auf der rechten Seite erscheinen Menüpunkte, die sich auf die daneben befindlichen Menütaster beziehen. Dieses Bedienkonzept ist auch bei Oszilloskopen und Analysern anderer Herkunft gebräuchlich. Gelegentlich verleitet die Menüanzeige dazu, auf das Display statt auf den zugehörigen Taster zu drücken. Vielleicht ist ein Nachfolgemodell des DSA 815-TG mit einem Touchscreen ausgestattet, so dass sich das Gerät wie ein Smartphone oder Tablet-Computer bedienen lässt.

## Erste Messung

Das Kennenlernen beginnt mit dem Durchhangeln durch die Menüs, was nicht bedeutet, dass es dort an Struktur mangelt. Im Gegenteil, es liegt an den fast zahllosen Möglichkeiten und Einstellungen, die das Gerät bietet. Das auf CD mitgelieferte Handbuch umfasst 220 Druckseiten, es kann allerdings nicht immer sofort Hilfestellung leisten. Zwar werden sämtliche Funktionen und Einstellungen minutiös beschrieben, doch manchmal lassen sich Irrwege nicht vermeiden, bevor die gesuchte Information gefunden ist. Nach einer mehr oder weniger langen Kennenlernphase lichtet sich solches Dickicht, eine Erfahrung, die keinem Kollegen des Elektor-Labors fremd ist. Zum ersten Ausprobieren an einem Messobjekt hatten wir einen



433-MHz-Minisender für Modellflugzeuge auserkoren, der noch in der Entwicklung ist. Der Träger bei 433 MHz ließ sich schnell finden, mit der Funktion „Auto“ wurde er vergrößert in die Mitte des Messfensters geschoben. Nach dem Rückschalten auf die volle Bandbreite 1,5 GHz zeigte sich, dass die zweite Harmonische nur etwa 12 dB kleiner als die Grundschwingung war. Erfüllt dieser Wert die geltenden Vorgaben? Das müssen wir noch überprüfen. Ein Spektrum-Analyser mit integriertem Signalgenerator ist das richtige Messgerät, wenn es darum geht, Filtercharakteristiken schnell und sicher zu bestimmen. Beim DSA 815-TG ist die Funktion „Normalisierung“ hilfreich, sie kompensiert kleine Abweichungen des Gespanns aus Signalgenerator und Signalauswertung vor dem Messvorgang.

Für das Einstellen der Parameter beim Messen und Darstellen der Signale bietet der DSA 815-TG eine Fülle an Möglichkeiten. Stellvertretend für die vielen einstellbaren Größen seien hier nur die Auflösungsbandbreite, die Abtastzeit und die Skalenteilungen genannt. Wichtige Parameter sind am Display ablesbar, diese Informationen stören die Signaldarstellung nicht. Selbstverständlich können in der Signaldarstellung Markierungen gesetzt werden. Die Messdaten wie auch die Einstellungen lassen sich nicht flüchtig speichern oder exportieren. Die Liste der Features ist außergewöhnlich lang, deshalb wollen wir es hierbei belassen. Da die Zeit drängte und unser Hochfrequenz-Experte Jan Buiting in Urlaub war, haben wir keine weiteren Gehversuche mit dem Spektrum-Analyser unternommen. Schon nach unserer ersten Kurzbegegnung mit dem Rigol DSA 815-TG stand fest, dass dieses Gerät fünf Sterne verdient. Nicht nur das Verhältnis des Gebotenen zum Preis hat uns überzeugt, auch sonst ist der DSA 815-TG ein Gerät, das die Herzen echter Hochfrequenz-Enthusiasten höher schlagen lässt.

Am Rand sei noch etwas erwähnt, das uns auffiel: Das Foto zeigt den Analyser von Rigol neben unserem Labor-Analyser, einem Modell von Advantest. Unser Gerät hat uns inzwischen über 20 Jahre hinweg treue Dienste geleistet. Trotzdem (oder gerade deshalb?) ist die Ähnlichkeit der Messgeräte einschließlich der Bedienelemente unverkennbar.

(120560)gd

## Weblinks

<http://eu.rigol.com/>  
[www.batronix.com](http://www.batronix.com)

# Da piept's wohl!

Von Thijs Beckers (Elektor-Redaktion & -Labor)

Kollege Ton Giesberts arbeitete im Elektor-Labor an einer Schaltung, die beim Finden verlorener Flugzeugmodelle helfen soll, die unglücklicherweise ein Maisfeld als Landeplatz auserkoren oder sich sonst irgendwie verirrt haben. Der Prototyp war fertig und funktionierte soweit gut. Doch würde ich diese Geschichte kaum erzählen, wenn nicht etwas Ungewöhnliches beim Test der Elektronik im Labor passiert wäre. Denn obwohl der Empfänger gut mit dem Sender kooperierte, fing er auch fremde Signale auf. Die hörten sich an wie drei kurze Pieptöne, die im Abstand von etwa einer Minute wiederkehrten. Mit einem Oszilloskop konnte man erkennen, dass die drei Piepser wie Manchester-codierte Kommunikationssignale aussahen. Zwar war der Empfänger auf das so genannte ISM-Band bei 433 MHz abgestimmt, doch wir hatten trotzdem keine Ahnung, wo diese Störungen herkamen. Wir vermuteten Sensoren für Feuer oder Einbruchsdetektoren als Übeltäter.

Unabhängig davon stellten wir fest, dass der Empfänger auch die Signale von Autoalarmanlagen registrierte. In Verbindung mit einem Oszilloskop könnte das beim Abhören von Autoalarmanlagen „hilfreich“ sein. Das klappt natürlich nur bei älteren Systemen mit Einweg-Kommunikation. Kürzlich hörte ich von Autodieben, die einen mit spezieller Elektronik ausgestatteten Modellhubschrauber über den Autos auf einem Parkplatz schweben ließen, während der eingebaute Sender die Autos mit vielen Codes für Schließanlagen attackierte. Eine unter dem Hubschrauber angebrachte kleine WLAN-Kamera ermöglichte einen guten Überblick über die Szene, und so konnten die Diebe sehen, wenn ein Auto auf einen zufälligen Code reagierte. Sie mussten also nur noch zum betreffenden Auto gehen und einsteigen, ohne eine Scheibe einzuschlagen.

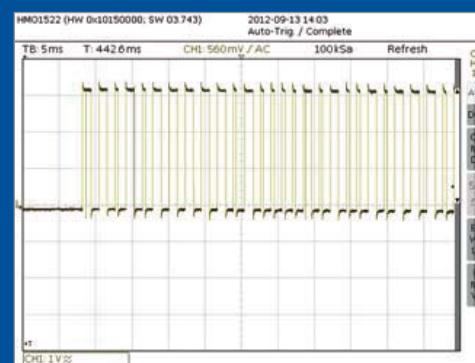
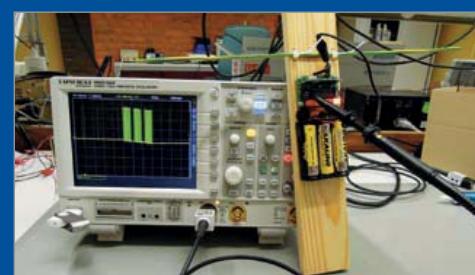
Das LED-VU-Meter des Empfängers reagierte übrigens auch

auf die Signale eines LC-Bildschirms, die wohl aus dessen Schaltnetzteil stammten. Neben PC-Bildschirmen reagierte es sogar auf das Display des Oszilloskops. Vielleicht wäre eine gute Metallabschirmung des Empfängers keine schlechte Idee gewesen.

Doch die Quelle des 3-Piep/Minute-Signals blieb weiter unentdeckt. Vielleicht unternehme ich demnächst kleine Spaziergänge mit dem TAPIR [1] in der Hand...

(120452)

[1] [www.elektor.de/120354](http://www.elektor.de/120354)



# Bauelemente-Tipps

Von Raymond Vermeulen (Elektor-Labor)

## Elektrische Energie - elektronisch gemessen

Angenommen, Sie wollen den Energie- oder Leistungsbedarf eines Verbrauchers bestimmen, der am Stromnetz angeschlossen ist. Das kann eine simple Glühlampe ebenso wie ein starker Elektromotor oder ein Wechselrichter sein. Oder Sie wollen die Energie messen, die Ihre Solaranlage in das öffentliche Stromnetz einspeist. Wie dem auch sei, Sie müssen die anliegende Spannung, den aufgenommenen (oder abgegebenen) Strom, die Phasenlage, die Wellenform sowie weitere wichtige Größen kennen. Die Halbleiter-Industrie hat so genannte Analog Front Ends (AFEs) auf den Markt gebracht, die genau diesen Bedarf abdecken. In den Bausteinen ist ein hohes Maß an Funktionalität integriert, so dass Sie dem AFE nur noch einen Mikrocontroller nachschalten müssen, der die Daten ausliest und aufbereitet. Hier stellen wir Ihnen zwei Beispiele vor.

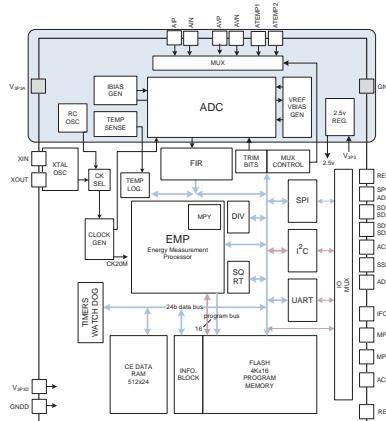
(120530)

### Maxim 78M6610+PSU

Mit diesem integrierten Baustein kann der Einschaltzeitpunkt auf den Nulldurchgang einer wechselwirkenden Größe gelegt werden. Einstellbar ist ein Offset, so dass sich beispielsweise das verzögern der Schaltverhalten eines Relais kompensieren lässt. Die Schaltvorgänge sind auch auslösbar, wenn die Spannung vorgegebene Grenzen übersteigt. Hier kann eine Hysterese den Schaltzeitpunkt verzögern. Eine besondere Fähigkeit ist das separate Erfassen der Energie, die auf die Harmonischen nicht sinusförmiger Größen entfällt. Spannungs- und Stromspitzen werden erkannt und registriert, und auch das Koppeln mit Temperatursensoren ist möglich.

- Temperatursensor, intern oder extern
- Einleiter-Messen von Strömen
- Relais-Steuerausgang für angeschlossene Systeme
- Messen von Wirk-, Blind- und Scheinleistungen
- Messen von RMS-Werten, sowohl Spannungen als auch Ströme
- Separates Erfassen von Grundschwingung und Harmonischen
- Speichern der letzten und höchsten RMS-Werte seit dem letzten Reset
- Speichern der höchsten Strom- und Spannungsspitze in vorgegebenem Zeitintervall
- Speichern des Power-Faktors
- Speichern der Netzfrequenz

Außerdem vorhanden: Konfigurierbare Ausgänge für ein anschließendes Alarmsystem.





Besuchen Sie uns auf der  
electronica 2012 in München  
(13.11.-16.11.)! Sie finden uns  
in Halle A5, Stand 314.

PCB Service

powered by Eurocircuits

## Platinen – Prototypen – Multilayer – Kleinserien

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



### PCB proto

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



### STANDARD pool

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



### TECH pool

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



### IMS pool

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



### On demand

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

**Wählen Sie den für Ihre Ansprüche passenden Service und bestellen Sie jetzt Ihre Platinen unter [www.elektorpcbservice.de](http://www.elektorpcbservice.de)!**

# GPIB-nach-USB-Konverter

## Low-cost-Interface für alte Messgeräte



Wenn Ihnen das Kürzel GPIB nichts sagt, sind Sie entweder jung und unschuldig oder un- beziehungsweise de-elektorisiert. Für alle anderen: Wir stellen hier einen für rund 40 € realisierbaren Wandler vor, mit dem man altehrwürdiges GPIB-Equipment an den modernen USB eines PCs anschließen kann.

Von Anders Gustafsson (Finnland)

Der GPIB (General Purpose Interface Bus, bei Hewlett Packard HPIB) ist ein Parallel-Interface nach IEEE488, das 1984 für den Anschluss von Sensoren und programmierbaren Messgeräten an einen Computer eingeführt wurde. An eine 24-polige Buchse können bis zu 15 Geräte angeschlossen und individuell adressiert werden. Das Equipment war zu GPIB-Hochzeiten irrsinnig teuer und das Kommunikationsprotokoll sehr komplex, so dass die Angelegenheit etwas für hartgesottene Elektronik-Profis und Raketenbauingenieure war, der Freizeit-Elektroniker aber damit gar nicht in Berührung kam.

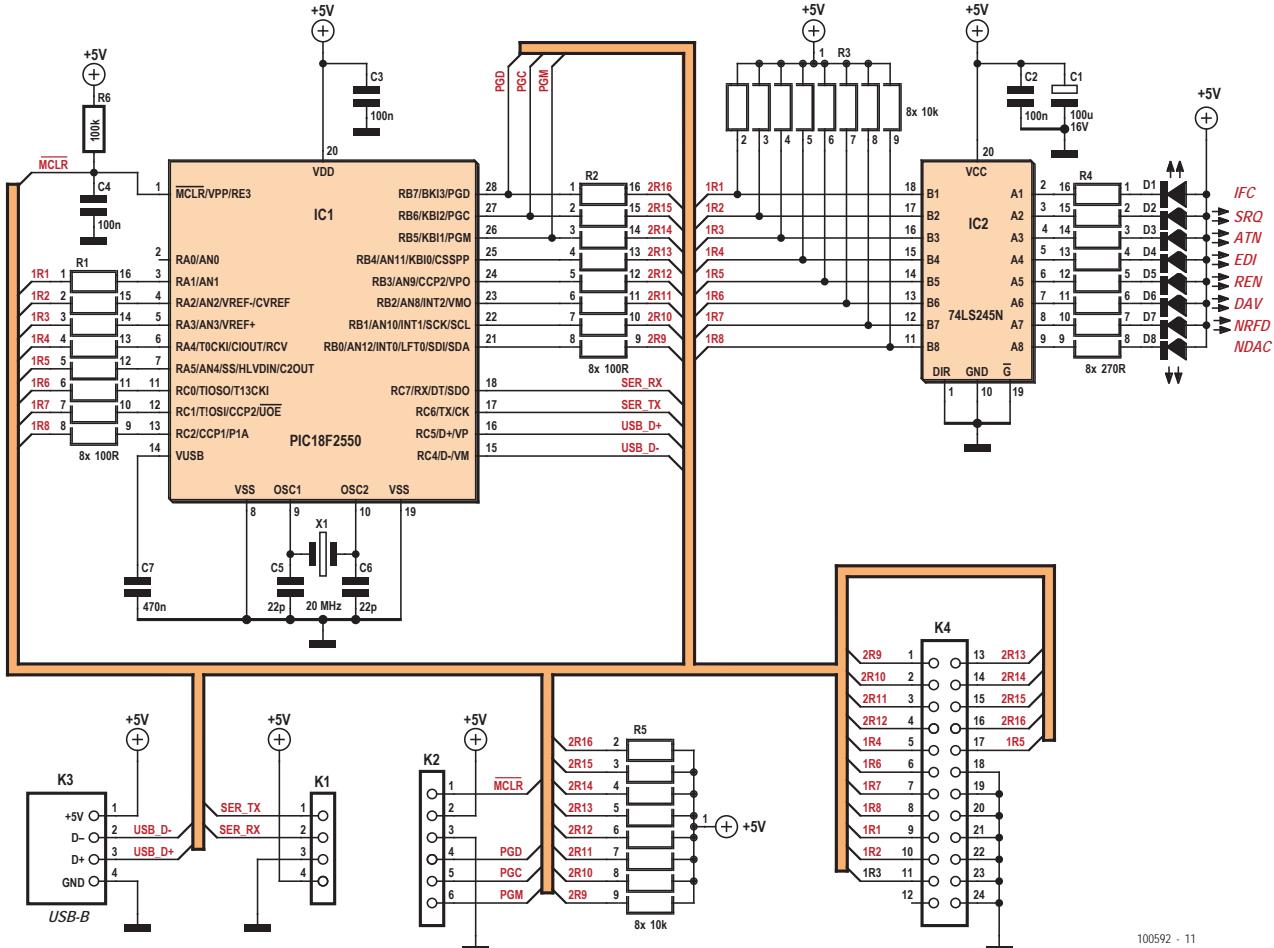
### Und warum?

Was soll nun der Sinn sein, Mühe und Geld zu verschwenden und einen Konverter zu bauen, um 30 Jahre alten Plunder an einen PC anschließen zu können? Die Antwort: Verschiedene Anbieter verkaufen (nicht zuletzt über eBay und ähnliche Märkte) wiederaufbereitete und kalibrierte, hochwertige und robuste GPIB-Messgeräte von Herstellern wie HP, Tektronix oder Marconi oft für weniger als 100 €. Risikofreudige Bastler können noch mehr sparen, wenn sie Geräte mit dem Label „nicht funktionsfähig“ oder „reparaturbedürftig“ erwerben und selbst wieder instandsetzen.

Während die meisten modernen chinesischen Messgeräte bei einem Defekt nahezu irreparabel sind, sind die altägyptischen GPIB-Geräte

### Herausforderungen für Teens und Twens

Analog Circuit Design, ASCII, Boat Anchor, Carriage Return, Centronics, Centronix, COM, Commodore, Floppy Disk, FUBR, Jim Williams, GPIB, IEEE, SN75160, SN75162.



in der Regel aus Standard-Bauteilen zusammengesetzt und nach guter alter US-Ingenieurtradition stabil wie Panzer aufgebaut. Oft kann man für das Schätzchen noch ein Service-Manual bei Quellen wie Boat Anchor Manual Archive [1] organisieren. Das Auffinden und die Reparatur kleiner und größerer Fehler ist zudem eine gute Übung. Nicht umsonst bezeichnen Gurus wie Jim Williams das Reparieren von Geräten als beste Möglichkeit, Elektronik zu lernen [2].

Viele der Zweit-, Dritt- oder Zehntheadgeräte verfügen über einen 24-poligen GPIB-Verbinder. Er sieht wie ein Centronics-Stecker aus (tatsächlich wurde die Schnittstelle auch von Commodore-Druckern und Floppylaufwerken genutzt) und ist heute kaum mehr zu sehen, war aber vor einigen Dekaden das Non-plus-ultra auf dem Gebiet der Datenerfassung.

Eine gründliche Erläuterung des GPIB würde den Rahmen dieses Artikels sprengen, es ist ausreichend zu wissen, dass man alle GPIB-kompatiblen Geräte an einen Computer mit einer USB-Schnittstelle anschließen und sie mit dem GPIB-Befehlssatz adressieren und konfigurieren kann. Zwar gibt es heutzutage auch GPIB-Interfaces für PCs, meist in Form von Einsteckkarten, aber es ist ja auch eine inter-

## Eigenschaften

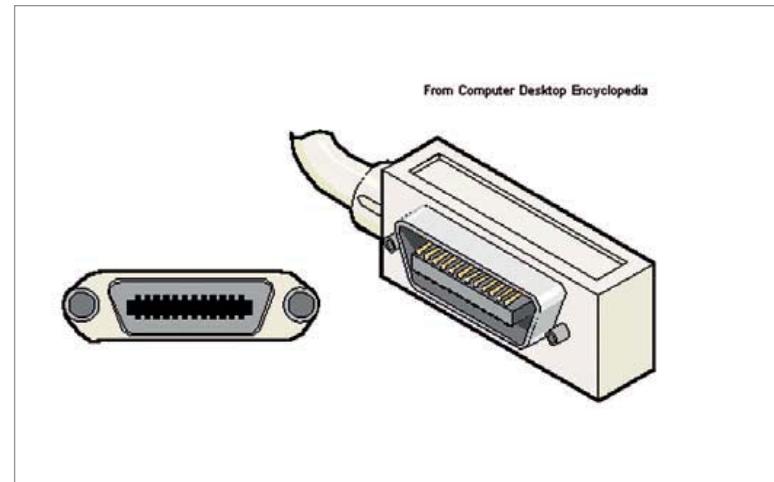
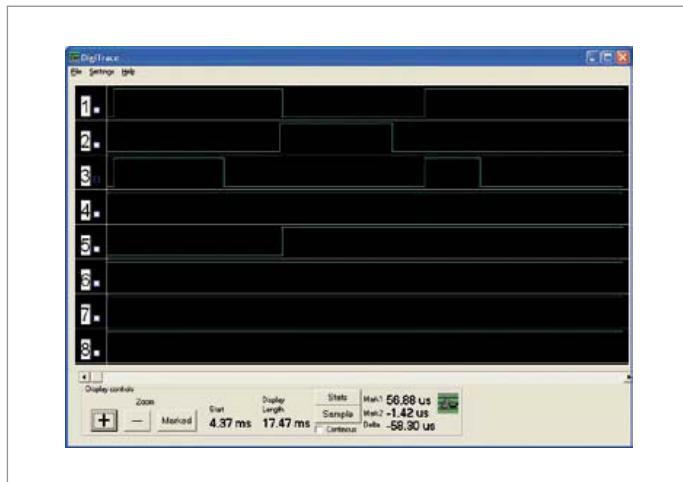
- Externer Low-cost-Adapter
  - Volle GPIB / IEEE488 Funktionalität über USB
  - PIC18F2550 @ 20MHz mit On-chip-USB-Interface
  - Emulation und Enumeration des PC-COM-Ports
  - Software in C entwickelt

essante Übung, einen Wandler für den omnipräsenten USB für weniger als 50 € zu bauen. Und das Rechnergehäuse kann geschlossen bleiben!

Elektor Produkte & Service

- Quellcode für PIC18F2550: freier Download
  - Platine #100592-1

Download/Bestellung unter [www.elektor.com/100592](http://www.elektor.com/100592)



## Talker und Listener

Wie Sie im Schaltplan sehen, benötigt man für diesen einfachen GPIB-USB-Wandler einen PIC-Mikrocontroller, getaktet mit 20MHz, ein bidirektionales Puffer-IC 74LS245 und ein paar passive Bauteile. LEDs zeigen den Status der GPIB-Befehlsleitungen an. Die Schaltung wird über den USB vom PC versorgt. Die wahre Magie dieser Schaltung steckt natürlich in der Software, die das CDC-Interface des USB verwendet, um den GPIB in einen altmodischen (sorry, es heißt eigentlich „legacy“) COM-Port zu verwandeln. Damit lässt sich der GPIB mit jeder Software ansteuern, die in der Lage ist, mit dem COM-Port zu kommunizieren. Beispiele sind hier für OpenOffice und Excel angegeben.

Ein voll ausgestatteter GPIB besitzt acht Datenleitungen (DIO1 bis DIO8) und acht Steuerleitungen, von denen allerdings nur drei für den aktuellen Transfer wichtig sind: DAV (Data Available), NRFD (Not Ready For Data) und NDAC (Not Data Accepted). Wenn als Ausgänge konfiguriert, besitzen die letzten beiden offenen Kollektoren mit Pull-

up-Widerständen, so dass alle „Listener“ am Bus sie auf High setzen müssen, damit der „Talker“ ein High lesen kann.

So stellt sich der Bus automatisch stets auf den langsamsten Listener ein und stellt damit sicher, dass keine Daten verloren gehen. Dieses Verfahren nennt man 3-Wege-Handshake. Im hier gezeigten Timing-Diagramm sind die oberen drei Signale DAV, NRFD und NDAC, der Rest sind Daten. NRFD geht auf High, wenn alle Listener bereit sind, Daten zu empfangen. In diesem Augenblick setzt der Talker die Daten auf den Bus und zieht DAV auf Low. Wenn der letzte Listener die Daten gelesen hat, geht NDAC auf High und wenige Millisekunden auch DAV. Dann startet der Zyklus erneut.

Es gibt zwar spezielle ICs für ein GPIB-Interface wie den SN75160/162, sie sind aber veraltet und ergo auch schwer zu besorgen. Deshalb benutzen wir dazu die gewöhnlichen I/O-Ports des PICs und täuschen einen offenen Kollektor vor, indem der Port für ein Low als Ausgang und für ein High als Eingang geschaltet wird.

## Hinweise zum PIC18F2550

(pic2550.txt im Software-Archiv 100592-11.zip)

Modifikation der PICDEM FSUSB Demos für den 2550er:

1. Kopiere das HardwareProfile → PICDEM FSUSB.h, nenne es HardwareProfile → PICDEM FSUSB 2550.h.
2. Editiere HardwareProfile.h; addiere eine Zeile:

```
#if defined(__18CXX)
  #if defined(__18F4550)
    #include „HardwareProfile - PICDEM FSUSB.h“
  #elif defined(__18F2550)
    #include „HardwareProfile - PICDEM FSUSB 2550.h“ // Added
```

3. Ändere den Prozessor-Typ in MPLAB (Configure, select device) auf 18F2550.

Pin	Label	Signal
1	DIO1	Data Transfer
2	DIO2	Data Transfer
3	DIO3	Data Transfer
4	DIO4	Data Transfer
5	EOI	End Or Identify
6	DAV	Data Valid
7	NRFD	Not Ready For Data
8	NDAC	Not Data Accepted
9	IFC	Interface Clear
10	SRQ	Service Request
11	ATN	Attention
12	Shield	Chassis Masse

Pin	Label	Signal
13	DIO5	Data Transfer
14	DIO6	Data Transfer
15	DIO7	Data Transfer
16	DIO8	Data Transfer
17	REN	Remote Enable
18	GND	DAV Masse
19	GND	NRFD Masse
20	GND	NDAC Masse
21	GND	IFC Masse
22	GND	SRQ Masse
23	GND	ATN Masse
24	GND	Signal Masse

Die Eingangsimpedanz ist sehr hoch, was den Bus nicht beeinflusst. Nur wenn eine Vielzahl von Geräten am gleichen Bus mit hoher Geschwindigkeit und/oder über lange Kabel gesteuert werden soll, ist der Einsatz dedizierter Bus-Interface-ICs eine gute Idee.

Der GPIB-Standard fordert, dass ein abgeschaltetes Gerät hochimpedant am Bus liegen soll, ansonsten kann es den Bus lahmlegen. Im Zweifelsfall sollten deshalb immer alle Geräte eingeschaltet sein.

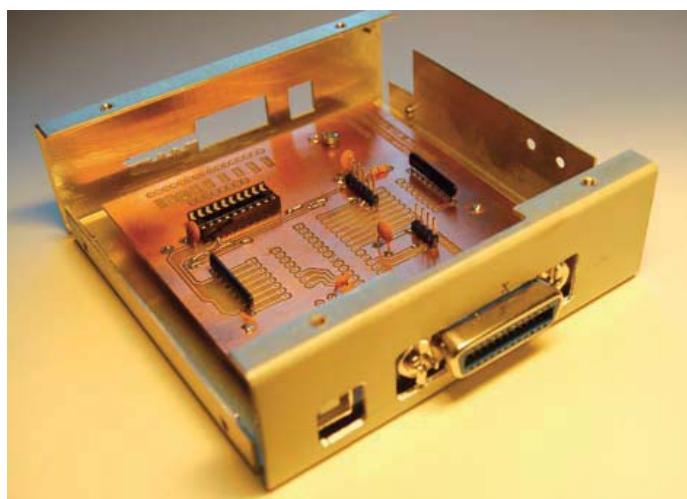
## Software

Die Software, die für dieses Projekt entwickelt wurde, basiert auf USB-Routinen von Microchip. Sie sind zwar kostenlos anwendbar, aber vom Urheberrecht geschützt, so dass sie nicht weiter als Quellcode verteilt werden dürfen. Die Archivdatei in [4] enthält deshalb nur alle anderen Elemente, die notwendig sind, um einen eigenen PIC für dieses Projekt zu brennen. An gleicher Stelle finden Sie auch Hinweise zum Erwerb eines fix und fertig programmierten PIC18F2550 von Elektor.

Um die HEX-Datei für den PIC zu erstellen, müssen zunächst die USB-Samples aus dem MCHPFSUSB-Framework [3] geladen werden. Gehen Sie dann zu C:\Microchip Solutions\USB Device → CDC → Basic Demo. Dort finden Sie ein Verzeichnis namens CDC → Basic Demo → Firmware. Legen Sie davon eine Kopie im gleichen Verzeichnis an und nennen Sie diese beispielsweise CDC → Basic Demo → Firmware GPIB. Dann werden die Dateien wie in pic2550.txt und in readme.txt beschrieben modifiziert. Öffnen Sie das Projekt USB Device → CDC → Basic Demo → C18 → PICDEM FSUSB.mcp und kompilieren Sie den Build komplett und ohne Fehler! Jetzt kann der PIC18F2550 von MPLAB aus programmiert werden.

## Aufbauen, anschließen, loslegen

Der Konverter lässt sich einfach auf der dargestellten Platine aufbauen. Ein 24-poliger IEEE/Centronics-Verbinder (male) für Platinenmontage ist allerdings schwer zu bekommen und kann ersetzt werden durch einen ausgeschlachteten Kabelverbinder und 24 Dräht-



## Stückliste

### Widerstände

R1,R2 = 100  $\Omega$ , DIL16-Array  
 R4 = 270  $\Omega$ , DIL16-Array  
 R3,R5 = 10 k, SIL9-Array  
 R6 = 10 k

### Kondensatoren

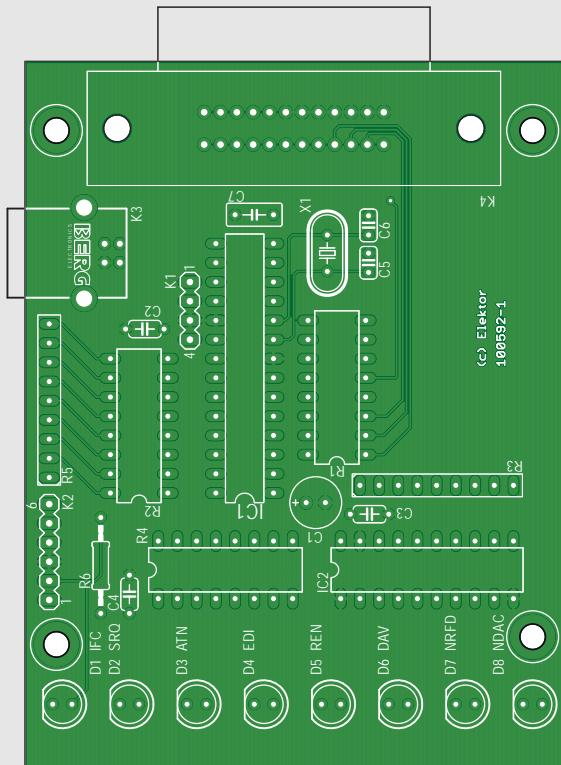
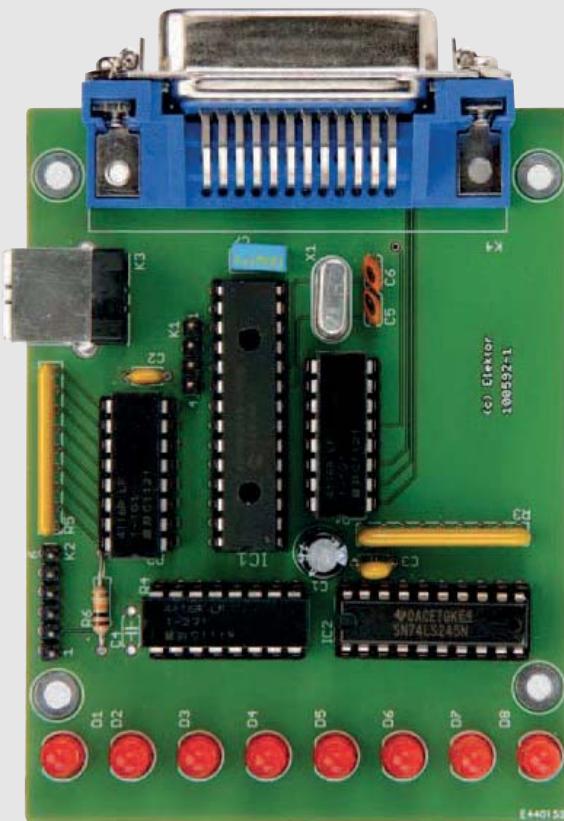
C1 = 100  $\mu$  16V, radial  
 C2,C3,C4 = 100 n  
 C5,C6 = 22 p, keramisch, RM 2,5 mm  
 C7 = 470 n

### Halbleiter

D1...D8 = LED, 3 mm, rot  
 IC1 = PIC18F2550-I/SP, programmiert, Elektor 100592-41  
 IC2 = 74LS245

### Außerdem

X1 = 20-MHz-Quarz  
 K1 = 4-poliger Pfostenverbinder, RM 2,54 mm  
 K2 = 6-poliger Pfostenverbinder, RM 2,54 mm  
 K3 = USB-Verbinder für Platinenmontage, Type B  
 K4 = 24-poliger IEEE488/Centronics-Verbinder, männlich, Platinenmontage, alternativ IDC-Version mit Platine verdrahten (Farnell # 1099278).  
 Platine 100592-1 [4]



chen zur Platine. Eine noch bessere Lösung ist der Einsatz eines 24-poligen IDC-Schneidklemmsteckers.

Ist alles erfolgreich aufgebaut, kann man den Wandler über ein USB-Kabel an den PC anschließen. Der PC meldet die neue Hardware und fragt nach weiteren Informationen. Man weist den PC auf die Datei mchpcdc.inf in C:\Microchip Solutions\USB Device → CDC → Basic Demo\inf hin. Ist alles korrekt abgelaufen, sollte jetzt ein neuer COM-Port zu sehen sein, den man mit einem Terminalprogramm (zum Beispiel HyperTerminal) ansprechen kann.

Als nächstes wird das GPIB-Equipment angeschlossen. Achten Sie auf die korrekte Adressierung. Um ein Gerät beispielsweise auf die Adresse 22°dezimal einzustellen, setzt man die Schalter (5,4, ... 1) auf 10110. Das Gerät kann nun mit „6“ als Listener und mit „V“ als Talker adressiert werden. Um das Gerät zum Sprechen zu bringen, sendet man 2<space>?UV Ctrl-Enter; und kann nun beobachten, wie es die (Mess-)Werte über HyperTerminal ausgibt.

Da der originale HPIB/GPIB-Standard nicht definiert, welche Befehle ein Gerät akzeptieren darf, sollte man das Handbuch des Geräts

## Auf dein GPIB-Kommando!

Die Software empfängt Daten auf dem GPIB und wandelt sie für den USB passend um. Beim Senden werden die Daten als Pakete verschickt, wobei das erste ein Befehl und das zweite leer (reserviert für zukünftigen Gebrauch) ist. Gültige Befehle sind:

- '1' – Sendet String (ATN nicht aktiviert)
- '2' – Sendet String als GPIB-Befehl mit ATN aktiviert
- '4' – Interface clear (setzt alle Geräte zurück)
- '6' – Assert GPIB REN (setzt Gerät in Remote-Modus)

'8' – Deaktiviert REN

'V' – Verbose-Modus (Echo eingegebener Befehle)

'Q' – Verbose-Modus aus

Befehlsstrings werden entweder durch einen Zeilensprung (line feed, ASCII 0A<sub>h</sub>) abgeschlossen, was ein EOI auf den Bus ausgibt; oder durch ein ETX (ASCII 0x03), was kein EOI verursacht. Der zweite Fall ist nützlich, wenn man zwischen dem Befehls- und Datenmodus während der Übertragung umschalten möchte.

### Auch interessant zu lesen:

- Kurzes GPIB Tutorial: [www.hit.bme.hu/~papay/edu/GPIB/tutor.htm](http://www.hit.bme.hu/~papay/edu/GPIB/tutor.htm)
- Tutorial von HP: <http://bitsavers.org/pdf/hp/hpib/TutorialDescrOfHPIB.pdf>
- Projektseite des Autors: [www.dalton.ax/gpib](http://www.dalton.ax/gpib)

### GPIB TUTORIAL

#### BACKGROUND

Instrumentation has always leveraged off widely used electronics technology to drive its innovation. The joweled movement of the clock was first used to build analog meters. The variable capacitor, the variable inductor, and the vacuum tube from radios were used to pioneer the first electronic instruments. Display technology was leveraged off the television for use in oscilloscopes and analyzers.

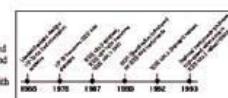
Today, cost-effective and powerful desktop and notebook computers are paving the way for new types of instruments – virtual instruments. Virtual instruments are designed and built by the user to match specific needs by leveraging off the power and low cost of PCs and workstations.

Software is the key to virtual instruments. Application software empowers the user with the tools necessary to build virtual instruments and expand their functionality by providing connectivity to the enormous capabilities of PCs, workstations, and their assortment of applications, boosting performance, flexibility, reusability and reconfigurability while eliminating at the same time development and maintenance costs.

FUNDAMENTALS OF VIRTUAL INSTRUMENTS	
Traditional Instruments	Virtual Instruments
Vendor-defined	User-defined
Function-specific, stand-alone with limited connectivity	Application-oriented system with connectivity to networks, peripherals, and applications
Hardware is the key	Software is the key
Expensive	Low-cost, reusable
Closed, fixed functionality	Open, flexible functionality leveraging off familiar computer technology
Slow turn on technology (5-10 year life cycle)	Fast turn on technology (1-2 year life cycle)
Minimal economics of scale	Maximum economics of scale
High development and maintenance costs	Software minimizes development and maintenance costs

#### INTRODUCTION

In 1965, Hewlett Packard designed the Hewlett Packard Interface Bus (HP-IB) to connect their line of programmable instruments to their computers. Because of its high transfer rates (eventually 1 Mbytes/s), this interface bus quickly gained popularity. It was later accepted as IEEE Standard 488-1975, and has evolved to ANSI/IEEE Standard 488.2-1987. Today, the name General Purpose Interface Bus (GPIB) is more widely used than HP-IB. ANSI/IEEE 488.2-1987 strengthened the original standard by defining precisely how controllers and instruments communicate. 8 standard Commands for Programmable Instruments (SCPI) took the command structures defined in IEEE 488.2 and created a single, comprehensive programming command set that is used with any SCPI instrument. Figure 1 summarizes GPIB history.



#### TYPES OF GPIB MESSAGES

## GPIB-Altertümchen für Sie!

- Stift/X-Y-Plotter
- Oszilloskop
- Stromversorgung
- Data Logger
- Sample & Hold-Einheit

- Telemetrie-Einheit
- Spektrum-Analyzator
- Multiplexer
- Funktionsgenerator
- Netzwerk-Analysator

Stichworte: Agilent, Advantest, Fluke, Hewlett Packard, Marconi, National Instruments, Tektronix, IEE 488.

# SDN – Software Defined NIC

## Ethernet für Mikrocontroller selbst stricken



Fertige Netzwerkanbindungen für Mikrocontroller gibt es wie Sand am Meer. Sie haben jedoch Nachteile: Die Hardware ist nicht ganz billig, die Einarbeitung zeitraubend und die Programmierung aufwändig.

Jedoch: Um Netzwerkpakete vom Controller aus auf die Reise zu schicken, benötigt man eigentlich nur wenige zusätzliche, einfache Bauteile. Die Entwicklung eines solchen „Software Defined NICs“ erfordert schon etwas Beschäftigung mit Netzwerkprotokollen. Die reine Nutzung dieser Lösung ist dafür kinderleicht.

Von Dr. Merten Joost (D)

Unsere SDR-AVR-Serie in den letzten Ausgaben hat gezeigt, was mit einem 8-bit-AVR-Controller alles möglich ist. Kaum zu glauben: Ein solcher, mit 20 MHz getakteter Mikrocontroller benötigt nur wenige externe Bauteile, um auch Ethernet-Netzwerkpakete aussenden zu können! Um zu zeigen, wie so eine „Software Defined Network Interface Card (SDN)“ funktioniert, benötigen wir aber zuerst etwas Theorie.

### TCP-Modell

Beschäftigt man sich mit dem Thema Netzwerk, kommt man üblicherweise an dem OSI-Referenzmodell für Netzwerkkommunikation nicht vorbei. In diesem Artikel wird jedoch das TCP/IP-Modell vorgezogen, da es der Realität näher kommt.

Wie in **Bild 1** zu sehen ist, besteht das Modell aus vier Ebenen, die aufeinander aufbauen. Die unterste ist als Netzzugangsschicht definiert, bei der die Netzwerkpakete auf das Ethernet-Kabel gelegt werden.

Die darüber liegende Vermittlungsschicht sorgt dafür, dass die Netzwerkpakete ihren Weg durch das Internet finden. Die Vermittlungsschicht stellt einen Kommunikationskanal zwischen zwei Rechnern her, die im Internet miteinander verbunden sind. Aufbauend auf diesen Dienst werden in der aufsitzenden Transportschicht Kommunikationswege zwischen Prozessen dieser Rechner aufgebaut. Das bekannteste Protokoll ist dabei das TCP (Transmission Control Protocol), das eine Punkt-zu-Punkt-Verbindung zwischen zwei Prozessen herstellt. Im Gegensatz zum IP (Internet Protocol) garantiert das TCP zudem, dass Pakete unversehrt und in der richtigen Reihenfolge beim Empfänger ankommen. Dies wird realisiert, indem der Empfang der Pakete quittiert wird. Aber es sind noch weitere, weniger bekannte Protokolle in dieser Ebene angesiedelt. Das UDP (User Datagram Protocol) stellt anders als das TCP kein sicheres Protokoll dar, denn es garantiert weder die Ankunft noch die richtige Reihenfolge der Pakete. Dafür bietet es mit Broadcasting einen Dienst, den

das TCP nicht beherrscht. Unter Broadcasting versteht man das gleichzeitige Versenden eines Paketes an mehrere Empfänger eines Netzes. Die Unsicherheit in der Übertragung muss außerdem relativiert werden. Alle unsicheren Schichten wie IP und UDP versuchen ihr Bestes, dass Pakete ausgeliefert werden. Gegebenenfalls werden Pakete auch wiederholt gesendet, falls festgestellt wird, dass Pakete verlorengegangen sind.

Die oberste Ebene im TCP/IP-Modell ist die Anwendungsschicht, die wiederum Dienste der unteren Schichten nutzt. Das bekannteste auf TCP aufbauende Protokoll ist wohl das HTTP, das im WWW für die Kommunikation zwischen Webservern und Webbrowsern sorgt. Es gibt aber auch Anwendungen, die bewusst auf das UDP setzen und die Unsicherheit der Datenübertragung in Kauf nehmen. So basiert interessanterweise das von Sun Microsystems stammende NFS (Net File System), das den Zugriff auf Dateien über ein Netzwerk ermöglicht, auf dem UDP.

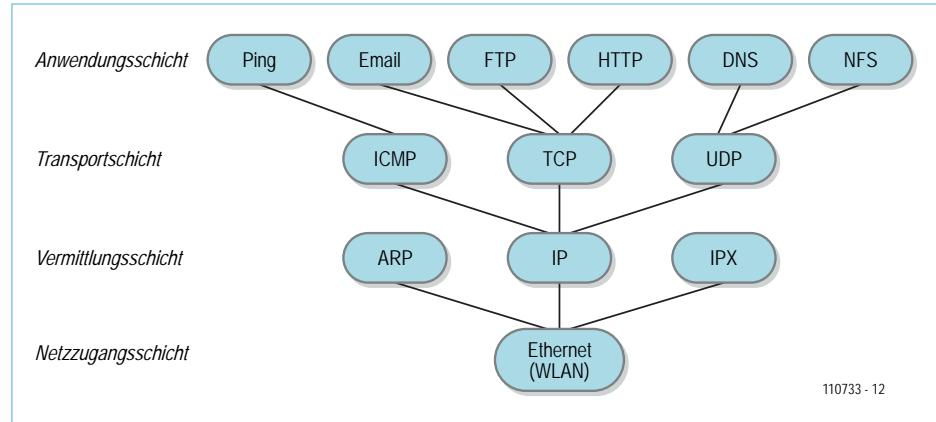


Bild 1. Das TCP/IP-Schichtenmodell.

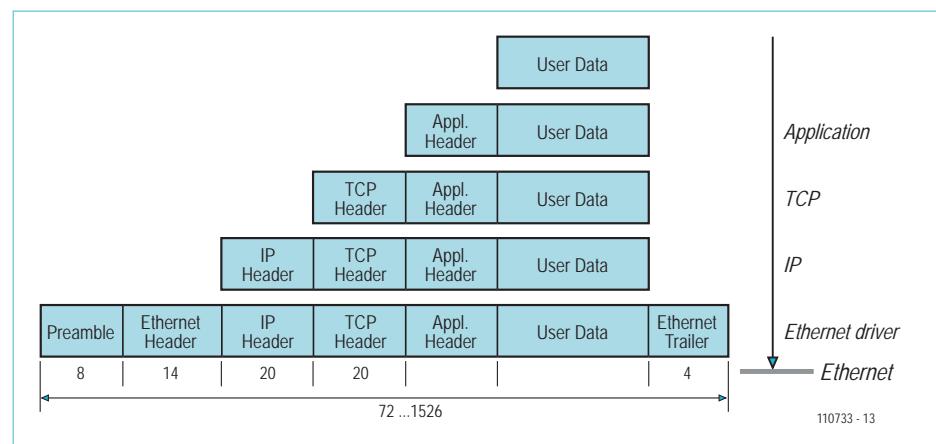


Bild 2. TCP-Datenkapselung.

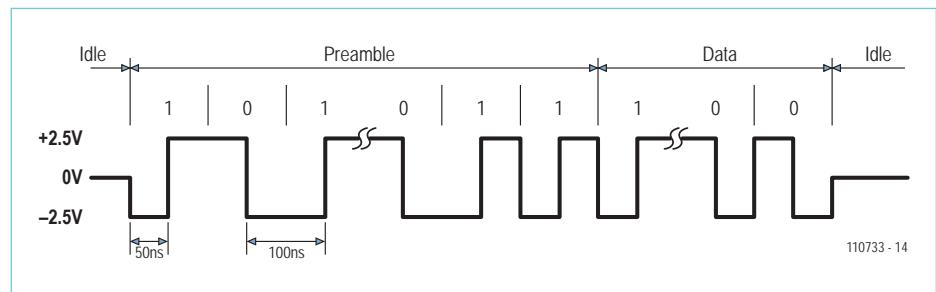


Bild 3. Präambel eines Ethernet-Pakets im Manchester-Code.

## Pakete packen

Alle Schichten packen die Daten, die sie von darüber befindlichen Schichten erhalten, in einen eigenen Frame ein. In diesem Frame sind die Informationen enthalten, die dafür sorgen, dass die Pakete beim empfangenden Prozess ankommen. Einige Protokolle hängen einen Header vor die Daten, andere hängen noch einen Trailer an. **Bild 2** gibt einen Überblick über den Aufbau eines Datenpakets, das über ein Ethernet-Kabel gesendet wird. Die User-Daten werden zunächst von der Applikation (z.B. einem Webserver) mit einem Header versehen und an die TCP-Schicht weitergeleitet. Diese hängt einen weiteren Header vor das Gesamtpaket, in dem ihm u.a. eine eindeutige Nummer zugewiesen wird. Weiterhin enthält der TCP-Header eine sogenannte Port-Nummer zur Identifikation des empfangenden Prozesses auf dem Zielrechner. Gleichzeitig kann im Header eine möglicherweise zuvor empfangenes Paket bestätigt werden. Anschließend wird das Paket an die IP-Schicht weitergereicht. Auch diese ergänzt das Paket um einen eigenen Header, der zum Beispiel die IP-Adressen von sendendem und empfangendem Rechner enthält. Danach wird das Paket an die Netzwerkhardware weitergereicht, die dem Paket weitere Daten hinzufügt. Neben dem Ethernet-Header, der u.a. die MAC-Adressen der beteiligten Rechner enthält, berechnet die Netzwerkkarte eine Prüfsumme nach dem CRC32-Verfahren und hängt das Ergebnis als Trailer an das Paket an. Zudem wird eine Präambel vor das Paket gesetzt. All diese Header, Trailer und die Präambel müssen in einer Softwarelösung nachgebildet werden.

## Elektrische Schicht

Der Einfachheit halber beschränkt sich die Beschreibung auf 10-Mbit-Ethernet. In seiner ursprünglichen Form [1] wurde als Medium ein Koaxialkabel genutzt. Auf diesem Kabel ist ein differentielles Signal mit einem Spannungshub von  $\pm 0,7$  V vorgesehen. Bei den heute verwendeten Twisted-Pair-Kabeln wird mit Pegeln von etwa  $\pm 2,5$  V gearbeitet. Die Codierung der Daten erfolgt im Manchesterverfahren nach IEEE 802.3. Dabei werden eine „1“ mit einer positiven Flanke und eine „0“ mit einer negativen

Flanke codiert (siehe **Bild 3**). Durch diese Codierung enthält das Signal indirekt einen Takt, da bei jedem Bit ein Flankenwechsel garantiert wird. Bei einem Datendurchsatz von 10 Mbit verdoppelt sich dadurch logischerweise auch die Übertragungsfrequenz auf 20 MHz. Bild 3 zeigt gleichzeitig die Präambel, die jedem Ethernet-Paket vorangestellt wird. Die Präambel ist acht Byte lang und besteht bis auf die beiden letzten Bits aus einer Folge von „1“ und „0“. Die beiden letzten Bits bestehen aus zwei Einsen und

deuten den Beginn der eigentlichen Daten an. Die Übertragung der Daten erfolgt im Little-Endian-Format - die niedrigen Bits werden also zuerst gesendet. Im Anschluss an die Daten wird die CRC32-Prüfsumme übertragen. Die Prüfsumme hat eine Länge von vier Bytes. Die Spezifikation schreibt vor, dass Ethernet-Pakete 72...1.526 Bytes lang sein müssen. Sind weniger Daten zu übertragen, müssen die Pakete aufgefüllt werden, um die Mindestlänge zu erreichen. Dies ist bei einer

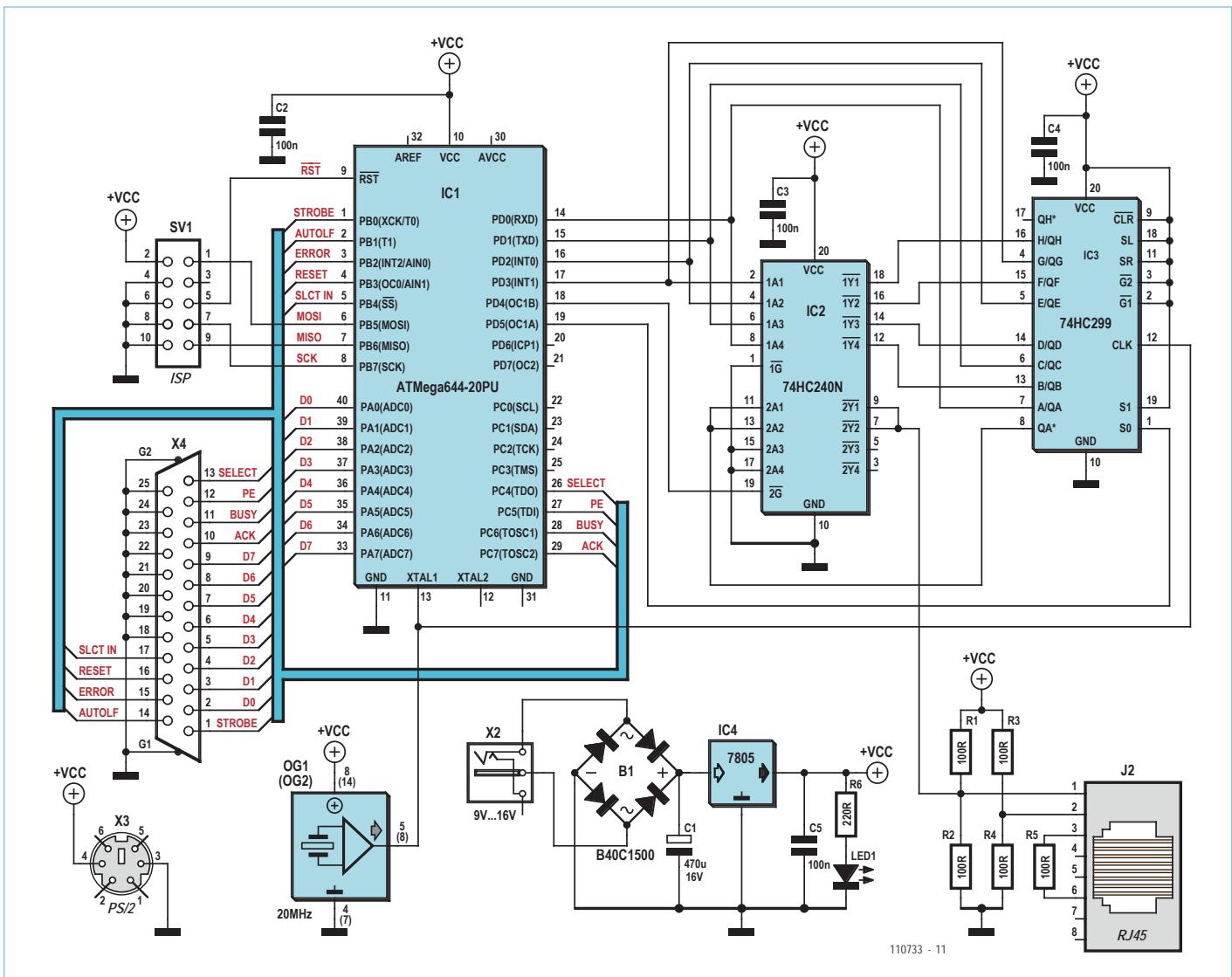


Bild 4. Die Schaltung des „Software Defined NICs“.

Implementierung ebenfalls zu beachten. Koppelt man eine 10-Mbit-Karte mit einem 100-Mbit-fähigen Gerät, so wird als mögliche Datenrate sofort 10 Mbit/s erkannt. Netzwerkgeräte sind abwärtskompatibel und passen ihre Übertragungsgeschwindigkeit der langsameren Komponente an. Zur Identifizierung senden Netzwerkgeräte kurze Impulse in regelmäßigen Abständen, an denen der Partner die Übertragungsfähigkeit erkennen kann. Bei 10 Mbit wird alle 16 ms ( $\pm 8$  ms) ein sogenannter NLP (Normal Link Pulse) gesendet. Dabei handelt es sich

um einen 100 ns langen positiven Impuls. Auch dieser Impuls muss in Software nachgebildet werden.

### AVR-NIC

Nun stellt sich die Frage, wie eine 20-MHz-CPU ein 20-MHz-Signal senden und empfangen kann. Eine Teilantwort vorweg: Der Empfang ist schlichtweg unmöglich. Allein das Empfangen und Wegschreiben in den Hauptspeicher benötigt mehrere CPU-Takte. An die gleichzeitige Interpretation der Daten ist gar nicht erst zu denken.

Ohne Empfang ist aber auch der Einsatz von TCP unmöglich, da ja empfangene Pakete quittiert werden müssen, was eine bidirektionale Kommunikation voraussetzt. Also bleibt als Übertragungsprotokoll nur UDP. Dem Nachteil der unsicheren Übertragung steht dabei der Vorteil des Broadcastings gegenüber, bei dem kein spezieller Empfänger in den Paketen eingetragen werden muss. Dies ermöglicht das Streaming von Video- bzw. Audiodaten ins lokale Netzwerk, die jeder angeschlossene Rechner empfangen kann.

NEU

Günter Spanner

# Arduino

Schaltungsprojekte  
für Profis



- Programmiergrundlagen
- Einfache Projekte für den Einstieg
- Schnittstellen- und Hardware-Erweiterungen
- Komplexe Anwendungen für Fortgeschritten

elektor

elektor

# Arduino

## Schaltungsprojekte für Profis

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

Einfache Einstiegsbeispiele sorgen für den schnellen Erfolg. Eine komplizierte Auswahl von Parametern, wie etwa Prozessorversion oder Schnittstelleneinstellungen sind nicht erforderlich. Erste Beispielprogramme können innerhalb weniger Minuten auf das Arduino-Board geladen und getestet werden. Unterstützt wird der Arduino-Anwender durch eine Fülle von Software-Bibliotheken. Die täglich wachsende Flut von Libraries stellt den Einsteiger vor erste Probleme. Nach einfachen Einführungsbeispielen ist der weitere Weg nicht mehr klar erkennbar. Hier fehlen oft detaillierte Beschreibungen und die mehr oder weniger gut beschriebenen Projekte führen eher zu Verwirrung. Ein klar erkennbarer roter Faden fehlt, da die Anwendungen natürlich von einer Vielzahl verschiedener Personen erstellt wurden, die alle jeweils ein spezielles Ziel vor Augen hatten.

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

270 Seiten (kart.) · Format 17 x 23,5 cm · ISBN 978-3-89576-257-4  
€ 39,80 · CHF 49,40

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

## EINKAUFSSPASS AUCH FÜR MÄNNER.

GROSSES PRODUKTSPEKTRUM ONLINE KALKULIERBAR.



**LEITON**

Einkaufen ist anstrengend? Dann kalkulieren Sie Ihre Leiterplatten gleich online bei LeitOn. Vor allem auch für Schablonen und flexible Leiterplatten! Das spart Ihnen Nerven, Geld und wertvolle Zeit! Und das bei dieser Online-Auswahl: 1- bis 12-lagige Multilayer mit vielen Sonderoptionen, flexible DK-Leiterplatten, SMD-Schablonen u.v.m. Der LeitOn Leiterplatten-Expressdienst garantiert: Platten sind gratis bei überschrittenem Liefertermin! Neugierig? Unsere persönliche Telefonberatung und unser kompetenter Außendienst helfen Ihnen gerne weiter.

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

Info-Hotline +49 (0)30 701 73 49 0



Ideen, die  
**FUNK**tionieren!

Besuchen  
Sie uns auf der  
13.-16. Nov.  
Stand A4.367

Sie haben die Ideen, wir die Lösung. Auch im stark frequentierten 2,4 GHz-Band ist eine zuverlässige Kommunikation möglich: dank DSSS-Modulation und zwei True-Diversity-Empfängereinheiten im Low-Power-Funktransceiver **STD-502-R** von Circuit Design. Niedriger Stromverbrauch und Batteriebetrieb machen das Modul ideal für Anwendungen, bei denen es auf Zuverlässigkeit bei großer Reichweite ankommt. STD-502-R erfüllt die EU-Norm EN 300440, die japanische Norm ARIB STD-T66 und die US-Norm FCC Part 15.247.



Nutzen Sie unsere Technologie  
und Kompetenz für Ihre Ideen.



Vertrieb durch:

**Reimesch**  
Kommunikationssysteme GmbH

Friedrich-Ebert-Str. 51429 Bergisch Gladbach  
Tel.: 0 22 04 / 58 47 51 · Fax: 0 22 04 / 58 47 67  
[www.reimesch.de](http://www.reimesch.de) · [kontakt@reimesch.de](mailto:kontakt@reimesch.de)

Vertrieb durch:

**CIRCUIT DESIGN** GmbH

Schleißheimer Str. 263 · 80809 München  
Tel.: +49 / 89 / 35 82 83-60 · Fax: +49 / 89 / 35 82 83-66  
[www.circuitdesign.de](http://www.circuitdesign.de) · [info@circuitdesign.de](mailto:info@circuitdesign.de)

Ethernet Header 14 Byte		
Destination MAC-Address 48 Bit		
Source MAC-Address 48 Bit		
Type 16 Bit		
IP-Header 20 Byte		
Version 4 Bit	Head. len 4 Bit	Type of Service 8 Bit
	Total Length 16 Bit	
	Identification 16 Bit	
Flags 3 Bit	Fragment Offset 13 Bit	
TimeToLive 8 Bit		Protocol 8 Bit
	IP-Header Checksum 16 Bit	
	Source IP-Address 32 Bit	
	Destination IP-Address 32 Bit	
UDP-Header 8 Byte		
Source Port 16 Bit		
Destination Port 16 Bit		
UDP Length 16 Bit		
UDP Checksum 16 Bit		

Bild 5. Vom SDN verwendete Ethernet-Header.

Auch das Senden im 20-MHz-Takt stellt eine anspruchsvolle Aufgabe dar, da hierzu die Daten in Echtzeit aus dem Speicher gelesen, ins Manchester-Format codiert und zum richtigen Zeitpunkt ausgegeben werden müssen. Gleichzeitig muss das Datenende überprüft und abschließend noch die CRC32-Prüfsumme ausgegeben werden. Dennoch genügt als zusätzliche Hardware

ein invertierender 8-bit-Treiber (74HC240) mit TriState-Ausgängen und ein 8-bit-Schieberegister (74HC299) mit paralleler Eingabe und serieller Ausgabe – siehe die Schaltung in **Bild 4**. Der Mikrocontroller gibt die Daten in Halbbytes aus. Mit vier Invertern von IC2 werden daraus acht Bits, welche die vier Daten-Bits im so genannten „invertierten Manchester-Verfahren“ codie-

ren. Diese acht Bits werden vom Schieberegister IC3 parallel eingelesen und seriell mit 20 MHz ausgegeben. In das Kabel werden die Daten über zwei parallelgeschaltete Inverter eingespeist, um so eine höhere Treiberleistung zu erreichen. Gleichzeitig wird dabei das noch invertierte Signal korrigiert. Das Kabel selbst wird an beiden Enden mit einem Spannungsteiler aus je zwei 100- $\Omega$ -Widerständen auf VCC/2 (= Ruhepegel) gelegt. Über eine Steuerleitung wird der Ausgang der beiden Treiber hochohmig gehalten, solange kein Paket gesendet wird. Mit dem Beginn einer Übertragung werden die Treiber freigeschaltet und erzeugen das differentielle Signal mit einem Spannungshub von etwa  $\pm 2$  V. Ein spezieller Übertrager ist nicht notwendig, eine einfache achtpolige Modularbuchse (RJ-45) genügt.

## Ethernet-Berechnungen

Damit die Ausgabe in Echtzeit erfolgen kann, muss das gesamte Daten-Paket einschließlich aller Header und Trailer fertig berechnet im RAM vorliegen, denn die Ausgabe der Daten lastet den Controller komplett aus. Daher müssen vor dem eigentlichen Senden alle noch variablen Inhalte der Header mit Daten gefüllt werden. Die Prüfsumme muss ebenfalls schon berechnet vorliegen.

**Bild 5** zeigt Aufbau und Inhalte der verwendeten Header. Im Ethernet-Header sind drei Felder auszufüllen. Die MAC-Adresse des Senders kann frei gewählt werden, muss sich aber von den anderen MAC-Adressen im lokalen Netz unterscheiden. Wenn nur ein bestimmter Zielrechner in Frage kommt, kann als MAC-Adresse des Empfängers eine konkrete MAC-Adresse eingetragen werden. Alternativ dazu kann die MAC-Broadcast-Adresse genutzt werden, die aus einer Folge von sechs 0xFF-Bytes besteht. Damit wird theoretisch jeder Rechner weltweit ein möglicher Empfänger. Das letzte Feld des Ethernet-Headers „Type“ ist mit dem Datum 0x0800 zu füllen. Diese Konstante zeigt an, dass es sich um ein IP-Paket handelt.

Etliche Felder des IP-Headers können mit festen Konstanten belegt werden. Sie beinhalten u.a. die IP-Versionsnummer, die Lebensdauer des Pakets und UDP als

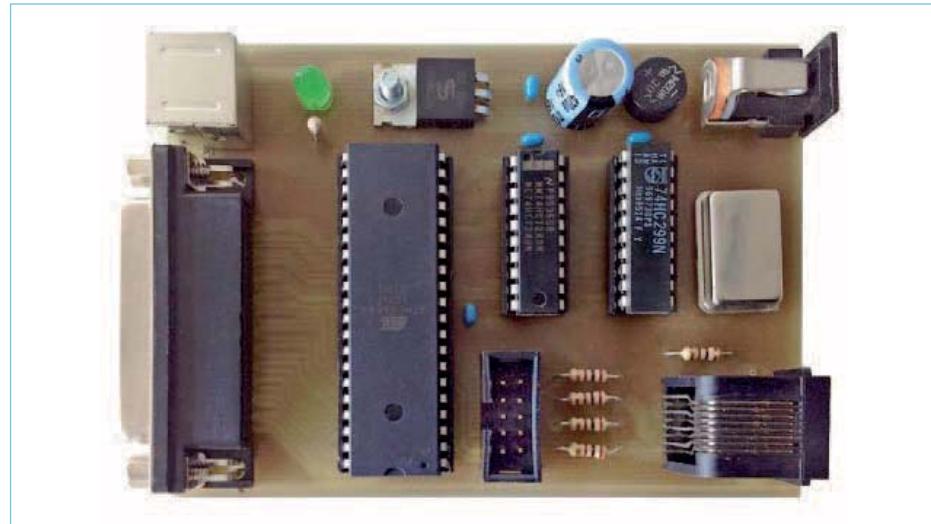


Bild 6. Der fertig aufgebaute Prototyp des Autors.

eingebettete Schicht. Auch die IP-Adressen von Sender und Empfänger können statisch sein, da sie sich nach dem lokalen Netz richten. Bei Broadcasting sind die letzten Bits entsprechend der Netmask mit Einsen zu füllen (die Netmask teilt die Adresse ein in die eigentliche Netzwerkadresse und einen Teil, der den Teilnehmer im lokalen Subnetzwerk bestimmt). Bei einer Netzwerkadresse von 192.168.1.0 und einer Netmask von 255.255.255.0 lautet die Broadcastadresse 192.168.1.255. Es existieren aber auch Felder, die bei jedem Daten-Paket neu zu bestimmen sind. Das Feld „Total Length“ beinhaltet beispielsweise die Länge des gesamten IP-Paketes (Header plus Daten). Damit Daten unterschiedlicher Länge gesendet werden können, wird die Paketlänge jedes Mal neu bestimmt. Außerdem muss eine Prüfsumme über den IP-Header berechnet und im Feld „IP Header Checksum“ abgelegt werden. Im Feld „Identification“ kann optional eine fortlaufende Nummer einge tragen werden, um das Paket identifizieren zu können.

Im UDP-Header sind die Ports von sendendem und empfangendem Prozess einzutragen. Hier reicht in der Regel ein konstanter Wert, der mit dem Port des empfangenden Prozesses übereinstimmen muss. Port-Nummern unter 1.000 sollten vermieden werden, da viele dieser Nummern bereits fest vergeben sind. Das Feld „UDP-Length“ ist mit dem Feld „IP-Length“ vergleichbar und wird in Abhängigkeit von der Paketgröße jedes Mal neu berechnet. Die UDP-Prüfsumme ist glücklicherweise optional und wird in dieser Version nicht berechnet.

Abschließend wird überprüft, ob das Gesamtpaket die nötige Mindestlänge besitzt. Ist dies nicht der Fall, wird das Paket um eine entsprechende Anzahl an Bytes verlängert. Der Inhalt dieser „padding bytes“ ist dabei beliebig. Erst jetzt kann die Prüfsumme über das fertige Datenpaket abschließend berechnet werden. Die Ethernet-Präambel fließt in die Berechnung nicht mit ein, die „padding bytes“ hingegen schon. Das Ergebnis der Berechnung wird an das Paket angehängt. Der Source-Code für die CRC32-Berechnung wurde einem Linux-Kernel entlehnt, kompiliert, disassembliert und anschließend aufwändig von Hand optimiert.

Es wurde eine Bibliothek in GNU-Assembler erstellt, die alle oben genannten Aufgaben erfüllt und zudem C-kompatibel ist. Damit können alle Anwendungen mit dem AVR-GCC (zum Beispiel in AVR Studio 5) programmiert werden. Die Bibliothek stellt zwei Funktionen zur Verfügung. UDPInit() erwartet als Parameter einen Pointer auf

einen Speicherbereich, in dem später das gesamte Ethernet-Paket liegen wird. Die Funktion trägt dann alle statischen Header-Informationen in diesen Speicherbereich ein. Weiterhin initialisiert die Funktion den 8-bit-Timer0 so, dass alle 13 ms eine Interrupt-Routine aufgerufen wird, die den NLP sendet. Die Daten werden 50 Bytes hinter dem Beginn des Paket-Speicherbereichs einge tragen. Die zweite Funktion UDPSend() berechnet anhand des Datenpaketes die noch fehlenden Header-Informationen, korrigiert nötigenfalls die Paketlänge, berechnet die CRC32-Prüfsumme, und schickt das Paket ins Netz. Diese Funktion erwartet als Parameter zwei Pointer, die auf den Beginn des Pakets und das Ende der eigentlichen Daten zeigen.

### Aufbau und Inbetriebnahme

Der SDN belegt die unteren sechs Pins eines beliebigen Ports eines AVR-Controllers. Über vier Pins werden die Daten ausgegeben, die beiden anderen Pins steuern das Schieberegister und die TriState-Inverter. Der Controller wird über einen externen

## IP-Cam

Als Testanwendung wurde eine alte Parallel-Port-Kamera an die noch freien Pins des Mikrocontrollers angeschlossen. Der Treiber-Code „cam.c“ plus Header „cam.h“ für die Kamera ist von einem alten, unter GPL stehenden DOS-Treiber inspiriert und auf den Controller portiert worden. Der Controller liest die Kamera zeilenweise aus und formt aus jeder Zeile ein UDP-Paket, dem die ent-

sprechende Zeilennummer vorangestellt wird. Die PC-Software „canvas.exe“ nimmt die UDP-Pakete entgegen und stellt das Bild zeilenweise dar. Es werden ca. zwei Bilder/Sekunde übertragen, was für einfache Überwachungsaufgaben genügt. Die so realisierte einfache IP-Kamera mit einem Materialwert von etwa 10 Euro ist eine schöne Demo-Anwendung des hier beschriebenen

Software Defined NICs.

Wer dies nachbauen will: Bei der Kamera handelt es sich um die „Creative Labs Video Blaster WebCam II, Modell 1100001424“ für den Parallel-Port, die gelegentlich für kleines Geld auf ebay.de oder ebay.com zu finden ist.

Oszillator mit 20 MHz getaktet. Das Takt-Signal ist gleichzeitig für das Schieberegister zuständig. Der geringe Aufwand lediglich mit IC2 und IC3 ermöglicht eine sehr kompakte einseitige Platine, deren Layout-Dateien nebst Sourcecode auf der Elektor-Webseite zu diesem Artikel [2] kostenlos als Download zur Verfügung stehen. **Bild 6** zeigt den Prototypen des Autors.

Im AVR Studio muss die Datei „udp.S“ als Source-Datei in das Projekt hinzugefügt werden. Das Listing „udp\_avr.c“ zeigt ein einfaches Testprogramm. Es reserviert einen gegenüber den Nutzdaten um 54 Byte größeren Speicherbereich. Die ersten 50 Byte sind für die Header und die letzten 4 Byte sind für die Prüfsumme reserviert. Um auch kleinere Pakete senden zu können, ohne den Speicherbereich jedes Mal neu zu allozieren, wird der Sendefunktion ein zweiter Parameter übergeben, der auf das Ende des zu sendenden Datenbereichs zeigt. Wird die Schaltung an ein Netzwerk angeschlossen, kann an einem PC mit dem freien Programm Wireshark [3] das Eintreffen der Datenpakete beobachtet werden.

## PC-Source

Um die vom SDN gesendeten Daten in einem eigenen PC-Programm zu nutzen, muss man UDP-Pakete empfangen können. Listing „udp\_pc.c“ zeigt, dass nur wenige Zeilen in C (GCC unter cygwin) für den Empfang von Daten über einen UDP-Socket genügen. Nach dem Aufbau des Sockets genügt der Aufruf der Funktion „recv()“. Auf dem PC ist man jedoch nicht an C gebunden. Für den Datenempfang nötige UDP-Sockets existieren auch unter anderen Hochsprachen wie Java oder Delphi.

## Durchsatz

Bei 10-Mbit-Ethernet geht man von 1 MB/s als Maximum aus. **Bild 7** zeigt den erreichbaren Datendurchsatz in Abhängigkeit von der Paketlänge. Man sieht, dass „nur“ eine Datenrate von 400 KB/s erreicht wird. Dies ist die Folge der aufwändigen Berechnung der CRC32-Prüfsumme. Messungen haben ergeben, dass die Berechnung der Prüfsumme etwas mehr Zeit in Anspruch nimmt als das eigentliche Senden der Daten.

(110733)

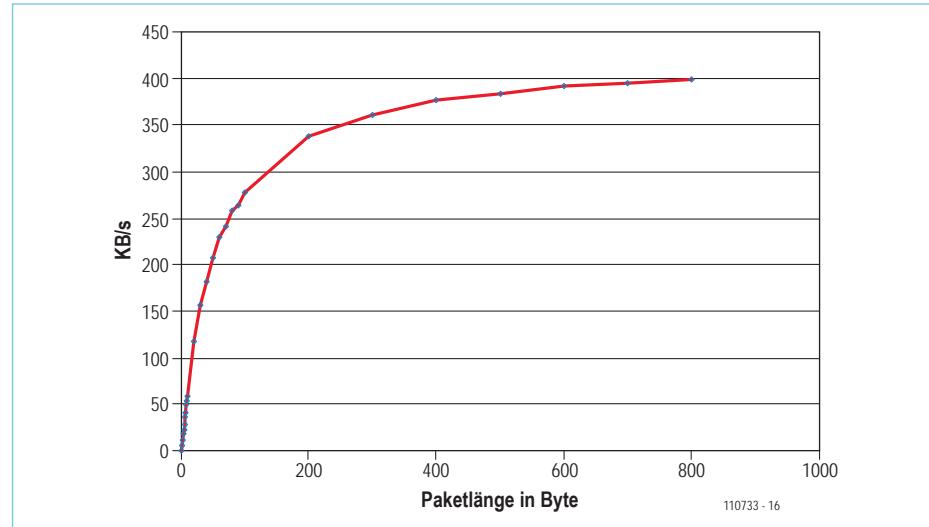


Bild 7. Der Durchsatz des SDN in Abhängigkeit der Paketlänge.

## Listing „udp\_pc.c“

```
#define BUflen 800
#define PORT 6668

main()
{
    SOCKET s;
    struct sockaddr_in adresse;
    unsigned char buf[BUflen];
    int num;

    if((s = socket(AF_INET,SOCK_DGRAM, 0))<0)
        printf("no socket %d\n",s),exit(0);
    adresse.sin_family      = AF_INET;
    adresse.sin_port        = htons(PORT);
    adresse.sin_addr.s_addr = htonl(INADDR_ANY);

    if(bind(s, (struct sockaddr*)&adresse, sizeof(adresse)) != 0)
        printf("bind failed\n"),exit(0);

    num=recv(s,buf,BUflen,0);
    // do something
}
```

## Weblinks

- [1] [http://research.microsoft.com/en-us/um/people/gbell/Ethernet\\_Blue\\_Book\\_1980.pdf](http://research.microsoft.com/en-us/um/people/gbell/Ethernet_Blue_Book_1980.pdf)
- [2] [www.elektor.de/110730](http://www.elektor.de/110730)
- [3] [www.wireshark.org](http://www.wireshark.org)
- [4] <http://userpages.uni-koblenz.de/~physik/informatik/techC.pdf>

## Über den Autor

Dr. Merten Joost ist Dozent an der Universität in Koblenz und lehrt dort die Bereiche Digitalelektronik, Mikrocontroller und Robotik. Die Inhalte dieses Artikels sind auch Bestandteile seiner Vorlesung Mikrocontroller [4], die neben Grundlagen der Rechnernetze auch die effiziente Programmierung von Mikrocontrollern vermittelt.

# Entwickeln Sie Prototypen noch von Hand?



## NI Multisim 12.0

Die leistungsstarke Umgebung für Elektronikentwicklung und Elektronikausbildung

- Intuitive Bedienung auch ohne SPICE-Kenntnisse
- Zeitersparnis durch schnelle Designvalidierung
- Cosimulation von analogen und digitalen Systemen mit NI LabVIEW
- PCB-Layouts mit NI Ultiboard

>> Mehr Informationen und kostenfreie Testversion unter [ni.com/multisim/d](http://ni.com/multisim/d)

089 7413130



[facebook.com/NationalInstrumentsDACH](https://facebook.com/NationalInstrumentsDACH)

Besuchen Sie uns auf  
der electronica 2012  
in Halle 1 an Stand 317



© 2012 | National Instruments, NI, LabVIEW, Multisim und Ultiboard sind Marken der National Instruments Corporation. Andere Produkt- und Firmennamen sind Warenzeichen der jeweiligen Unternehmen.

**elektor**  
PCB Service

powered by Eurocircuits

**25% Rabatt auf alle neuen Elektor-Platinen!**

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

Elektor-Platinen jetzt unter [www.elektor.de/pcb](http://www.elektor.de/pcb)  
zum Einführungspreis bestellen und 25% sparen!

# Gleichrichter mit niedrigem Leckstrom mit Nanopower-Opamp

Von Martin Tomasz, Senior Scientist, Touchstone Semiconductor, Inc.

Ein Nanopower-Opamp, ein p-Kanal-MOSFET mit niedriger Schwellspannung und zwei Widerstände sind die Ingredienzien eines Gleichrichters mit einer niedrigeren Schleusenspannung als bei einer Diode (Bild°1). Da die gleichgerichtete Ausgangsspannung die angeschlossene aktive Schaltung versorgt, ist keine zusätzliche Spannungsversorgung erforderlich. Gleichzeitig ist der Ruhestrom der Schaltung geringer als der Leckstrom der meisten Schottky-Dioden. Die Schaltung erzielt eine aktive Gleichrichtung bei Spannungen von nur 0,8°V, bei noch niedrigeren Spannungen arbeitet die integrierte Body-Diode des MOSFETs als gewöhnliche Diode.

Bei einer bestimmten Differenz zwischen der Aus- und Eingangsspannung schaltet der Opamp den MOSFET ein. Es gilt die Gleichung:

$$U_{GATE} = U_{OUT} - (R_2 / R_1) \times (U_{IN} - U_{OUT})$$

wobei  $U_{GATE}$  die Spannung am Gate des MOSFETs,  $U_{IN}$  die Eingangsspannung und  $U_{OUT}$  die Ausgangsspannung ist. Die Ein- und Ausgangsspannungen verhalten sich zu den Spannungen am MOSFET wie folgt:

$$U_{DS} = U_{IN} - U_{OUT} \text{ und } U_{GS} = U_{GATE} - U_{OUT}$$

wobei  $U_{DS}$  die Drain-Source- und  $U_{GS}$  die Gate-Source-Spannung ist. Zusammengefasst ergeben diese Gleichungen die Gate-Source- als Funktion der Drain-Source-Spannung:

$$U_{GS} = -(R_2 / R_1) \times U_{DS}$$

Es ist eine gute Idee,  $R_2$  12 Mal größer zu wählen als  $R_1$ . Dann fallen zwischen Drain und Source nur etwa 40°mV ab, und der MOSFET wird auch bei niedrigen Drainströmen eingeschaltet (Bild°2). Ein höheres Verhältnis reduziert diese Spannung weiter bis in die Grenzen der Eingangs-Offset-Spannung des Opamps von 6°mV.

Die Versorgung des Opamps (Touchstone Semiconductor TS1001) übernimmt der am Ausgang angeschlossene Kondensator  $C_1$ . Da der Opamp Rail-to-rail-Ein- und Ausgänge aufweist und keine Phasenumkehr beim Betrieb nahe dieser Rail-Spannungen stattfindet, arbeitet diese Versorgung ganz gut. Zusätzlich arbeitet der Opamp mit sehr niedrigen Betriebsspannungen (bis hinunter zu 0,8°V), so dass auch die Gleichrichtung bei solch niedrigen Spannungen funktioniert. Der nichtinvertierende Anschluss des Opamps ist mit  $V_{DD}$ , der Ausgang mit dem Gate des MOSFETs verbunden. Die Schaltung konsumiert nur wenig mehr als 1°µA beim Gleichrichten einer 100-Hz-Sinuswelle, das ist weniger als der Leckstrom der meisten Schottky-Dioden. Der Schwellwert des BSH205 liegt so niedrig, dass

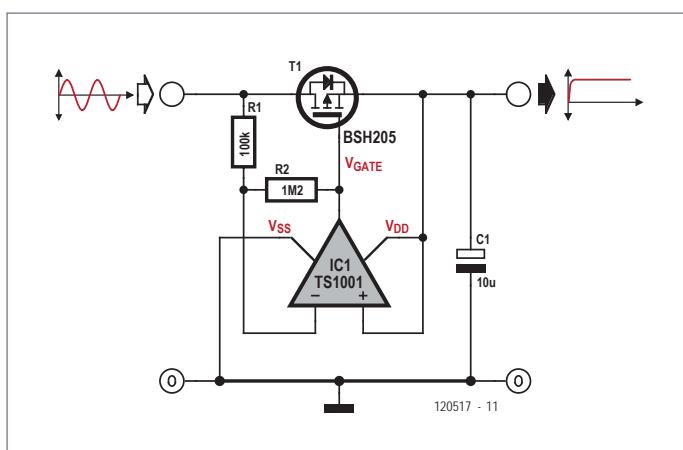


Bild 1. Die Schaltung arbeitet als Gleichrichter, weist aber eine Schleusenspannung von nur 40°mV oder weniger auf. Der Leckstrom ist dabei geringer als bei einer Schottky-Diode.

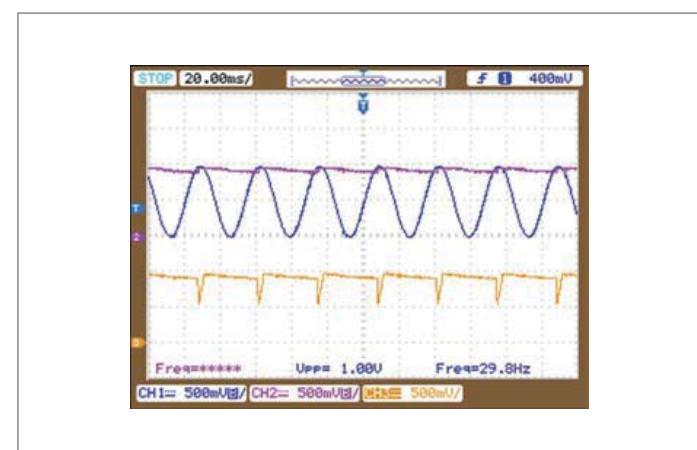


Bild 2. In diesem Diagramm sieht man das Ausgangssignal (violet) bei einer sinusförmigen Eingangsspannung (blau). Die Gatespannung des MOSFETs (gelb) fällt nur ab, wenn die Differenz zwischen Aus- und Eingang unter 40°mV sinkt.

1. Expand the following:  
 $(2x + 3y)^2$  (5 points)

$$\begin{aligned}
 & (2x + 3y)^2 \\
 & (2x + 3y)^2 \\
 & (2x + 3y)^2 \\
 & (2x + 3y)^2
 \end{aligned}$$

+2

Very funny

If only RF could  
be so easy.

**Linx**  
TECHNOLOGIES

Wireless made simple®

RF Modules  
Remote Controls  
Antennas  
RF Connectors  
Custom Designs

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

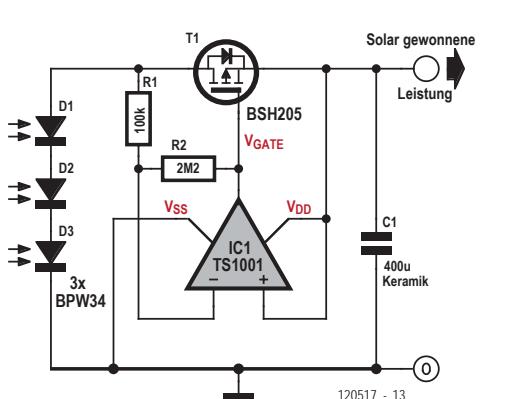


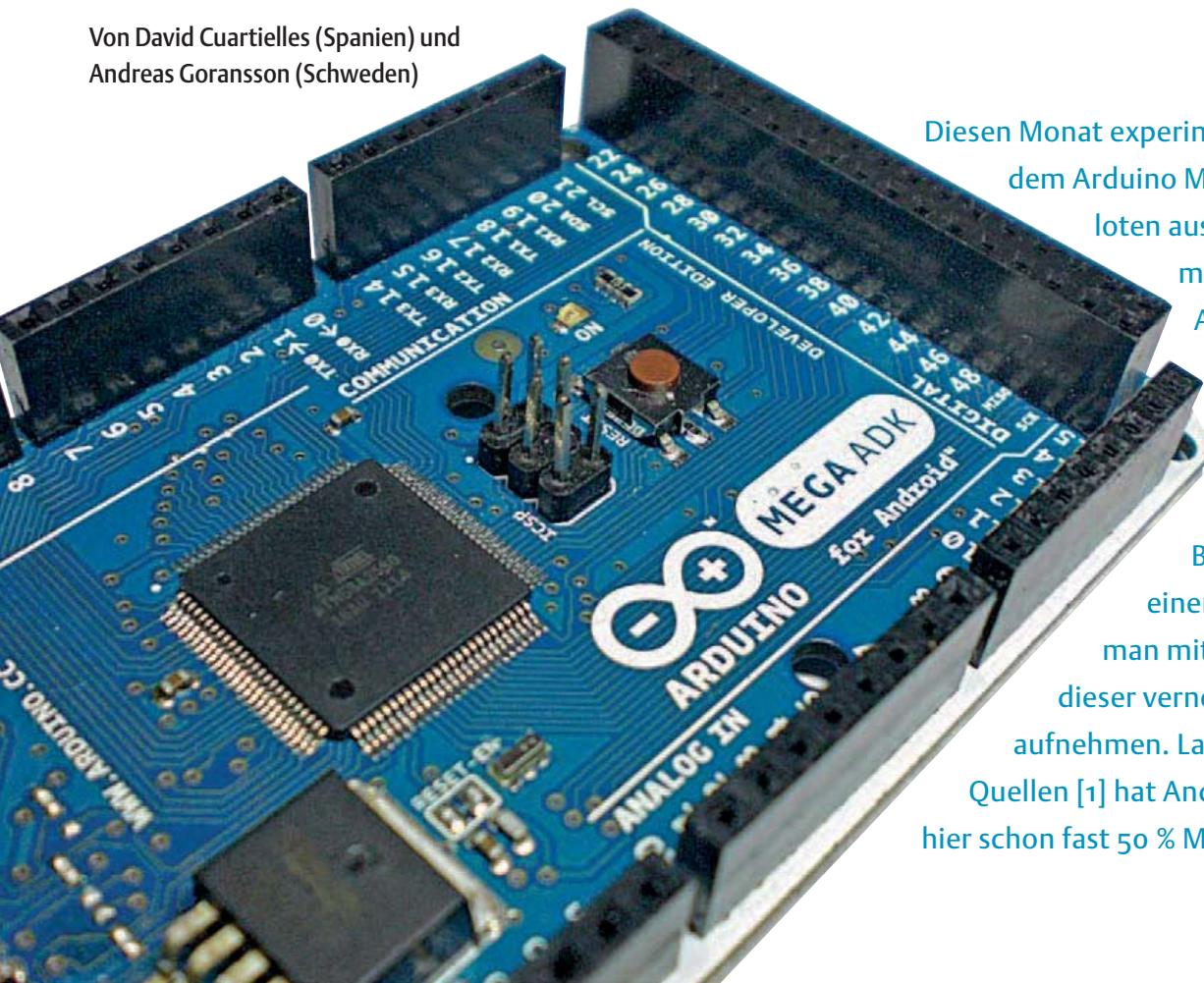
Bild 3. Der aktive Gleichrichter kann genutzt werden, um einen Kondensator über eine Solarzelle zu laden.

Er weist einen geringen Spannungsabfall auf und schützt gleichzeitig vor einem Rückfluss der gespeicherten Energie bei Dunkelheit.

# Arduino auf Kurs (3a)

## Teil 3a: Android Smartphone mit der realen Welt verbinden

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



Diesen Monat experimentieren wir mit dem Arduino Mega ADK und loten aus, wie wir uns mit einem unter Android laufenden Smartphone oder Tablet verbinden können. Mit Android als Betriebssystem auf einem Mobilgerät kann man mit nahezu allem in dieser vernetzten Welt Kontakt aufnehmen. Laut unabhängigen Quellen [1] hat Android im letzten Jahr hier schon fast 50 % Marktanteil erreicht.

Mit Android ist es für jeden möglich, Hardware-Erweiterungen für Smartphones zu bauen. Dieser Artikel bietet eine Anleitung, wie man sein System so aufsetzt, dass man dafür passende Erweiterungen und Apps entwickeln kann.

### Notwendige Hardware

Hier der Einkaufszettel für die folgenden Experimente:

- Arduino Mega ADK (siehe obiges Foto) oder Arduino Uno mit USB-Host-Shield
- Gerät mit Android Version 2.3.4 oder neuer
- USB-Kabel und Mikro-USB-Kabel

Hinweis: In diesem Artikel geht es vor allem darum, das System so vorzubereiten, dass man damit Arduino-Applikationen entwickeln kann, in die sich ein Smartphone einbinden lässt. Außerdem stellen wir (in der nächsten Folge) noch ein paar fertige Apps samt Source

zur Verfügung, mit denen man zwischen Smartphone und Board Daten austauschen kann.

Tipp: Bevor Sie weiterlesen sollten Sie zuerst allen notwendigen Code herunterladen, da wir aufgrund des Umfangs hier nur Ausschnitte präsentieren können.

### Einführung in Android

Android ist hauptsächlich als OS für mobile Geräte konzipiert. Wie viele andere Derivate basiert es auf Linux. Wichtig ist auch seine Java-ähnliche virtuelle Maschine namens Dalvik.

Da Android fürs Handy gedacht ist, haben typische Telefoniefunktionen Priorität. Ein eingehender Anruf beispielsweise unterbricht jede andere Aktivität. Weiter wird vom Vorhandensein einer Reihe von Sensoren z.B. für Helligkeit etc. ausgegangen.

Außerdem gibt es mehrere Wege, die ins Internet (oder zu anderen Geräten) führen: GPRS/3G/4G, WLAN, Bluetooth, USB-Kabel etc.

Schließlich programmieren wir keine nativen Applikationen, sondern Code, der unter Dalvik ausgeführt wird. Bezuglich Syntax sind die Programme äquivalent zu Java. In vielen Fällen kann man sogar Java-Libraries (als Source) direkt in die Apps importieren. Android sorgt für einen bestimmten Grad an Portabilität zwischen den „Terminals“ verschiedener Hersteller. Mit anderen Worten: Eine für Android kompilierte App sollte auf unterschiedlichen Smartphones laufen, solange sie die gleiche OS-Version haben.

### Add-ons für Android

Das Android-Entwicklungs-Team ermöglichte 2011 so genannte „accessory devices“ (= Erweiterungslösungen) mit Hilfe von Open-Source-Tools. Das bekannte Google Accessory Development Kit [2] besteht aus einigen Hardware Reference Designs und dem AOAP (Android Open Accessory Protokoll).

Bei den Hardware Designs handelt es sich um Varianten unterschiedlicher Arduino-Boards. In diesem Beitrag wird das Arduino Mega ADK eingesetzt, da es mit der ersten Version von AOAP kompatibel ist. Der in diesem Artikel verwendete Code sollte auch auf neueren Boards wie Arduino Due etc. laufen, die ADK-kompatibel sind. Es sollte auch möglich sein, die hier beschriebenen Experimente mit einem Arduino Uno plus USB Host Shield wie in **Bild 1** nachzuvollziehen.

Eine der zentralen Eigenschaften von AOAP ist das „App Wakeup“ bei einer Accessory-Detektion. Wenn so ein Accessory an ein Smartphone oder Tablet angeschlossen wird, dann triggert das einen Aufruf der App, deren Name durch das Accessory bestimmt wird. Falls die App noch nicht installiert ist, informiert das Accessory über eine URL, über die man die App downloaden kann.

Zur Zeit der Arbeit an diesem Artikel gab es schon eine ganze Reihe Boards außer dem Arduino Mega ADK, die mit dem Google ADK kompatibel sind. Leider gibt es aber noch keine Boards außer dem Arduino Due, die mit Googles ADK2 kompatibel wären.

### Arduino-Setup

Der Einfachheit halber starten die Experimente mit der Vorbereitung der Arduino IDE dahingehend, dass damit AOAP-kompatibler Source-Code für Arduino Mega ADK compiliert werden kann. Neben der richtigen Hardware benötigt man noch die Installation von:

- Arduino 1.0.1 oder neuer
- Die USB-Host-Library für Arduino (enthält die Klasse `AndroidAccessory`).

Wenn sie noch nie eine Library für Arduino installiert haben, dann halten Sie sich einfach an die Anleitung des nächsten Abschnitts. Dank der Arbeitsweise von AOAP wird Android dann, wenn der richtige Code auf dem Arduino Mega ADK läuft, vom Accessory profitieren, selbst wenn die passende App nicht auf dem Smartphone installiert ist. Dann nämlich wird wie in **Bild 2** die passende Webseite zum Download der passenden App vorgeschlagen.

Hinweis: Damit diese automatische App-Installation funktioniert, muss man beim Smartphone die entsprechende Option aktivieren,

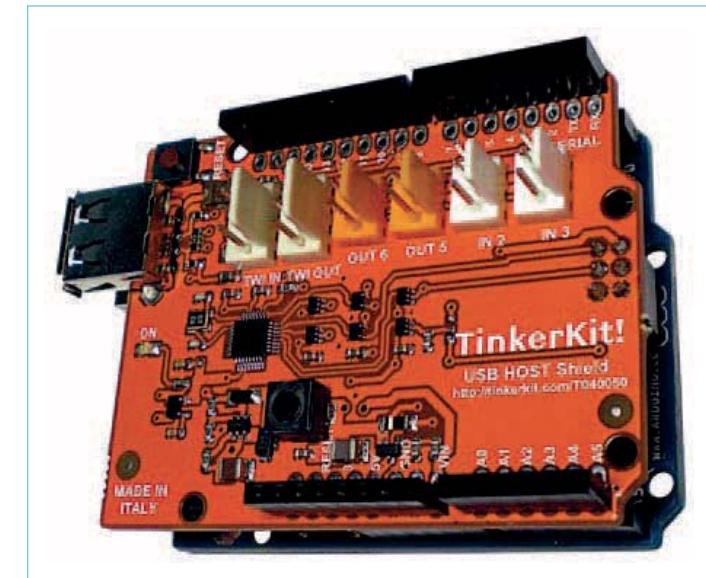


Bild 1. Arduino Uno mit USB-Host-Shield.

damit man auch Apps von außerhalb des offiziellen Google Play Market installieren kann. Via Einstellungen -> Sicherheit sollte man die Option „Unbekannte Herkunft“ aktivieren und für eine Internetverbindung über WLAN oder Mobilfunk sorgen, damit die App auch von unseren Servern herunter geladen werden kann.

### USB-Host-Library

Als dieser Artikel vorbereitet wurde, enthielt die Arduino IDE die vorgestellte USB-Host-Library noch nicht. Von daher sieht man im Menü „Sketch / Import Library“ noch keine Library namens „USB Host“ in der Liste – man muss dann die Library von der offiziellen Arduino-Projektwebseite [3] downloaden und installieren. Das Hinzufügen einer neuen Library besteht im Kern darin, dass man ein Verzeichnis namens „libraries“ innerhalb des Sketchbooks anlegt und darin direkt die Datei von [3] entpackt. Anschließend startet man die Arduino-IDE neu und die Library zeigt sich im zuvor erwähnten Menü.

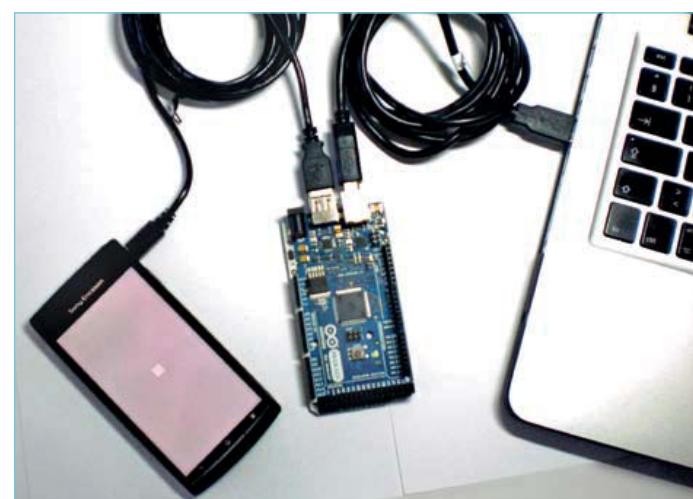


Bild 2. Arduino Mega ADK, angeschlossen an einen PC und ein Handy.

Bei einer älteren Version dieser Library (oder wenn sie von einer anderen Quelle als der angegebenen stammt) empfiehlt sich deren Deinstallation, bevor man sie auf die aktuelle Version bringt, da dies sonst zu Inkompabilitäten führen kann.

Mit dieser Library installiert man nicht nur den Code zur Ausführung verschiedener Kommandos (wie bei einem Modem), sondern auch eine ganze Reihe von Beispielen, mit denen man folgendes tun kann:

- Test auf korrekte Funktion des ADK-Modus
- Senden von digitalen/analogen Werten vom Board an das Handy
- Empfangen von Werten vom Handy durch das Board
- Debugging von unterschiedlichen an Arduino Mega ADK angeschlossenen USB-Geräten

Der Zugriff auf diese Beispiele ist sehr einfach, man navigiert einfach per Menü nach „File / Examples / USB Host“.

## App starten

Zum Test, ob die Sache sauber arbeitet, eignet sich das einfachste mögliche Beispiel. Man muss lediglich ein Programm auf Arduino laden, welches das Smartphone dazu veranlasst, eine einfache App namens Elektor\_MIAU herunter zu laden und etwas auf dem Bildschirm anzuzeigen (*Listing 1*).

Wenn man diesen Code auf Arduino Mega ADK gebracht hat, sollte man den „Serial Port Monitor“ in der IDE öffnen, um zu inspizieren, was da passiert. Wenn Sie Ihr Smartphone per Mikro-USB-Kabel an das Arduino-Board anschließen, wird der Serial Port Monitor etwas Ähnliches wie in **Bild 3** anzeigen.

Gleichzeitig wird Android das Arduino-Board detektieren. Da noch keine App zur Behandlung der Daten installiert ist, erhält man eine Meldung darüber, wo man die richtige App im Internet herbekommt. Man sollte so etwas wie in **Bild 4** sehen können.

Es ist Ihnen sicher aufgefallen, dass der angezeigte URL exakt dem im obigen Listing entspricht. Man gibt also im Arduino-Code an, wo im Internet die App steckt. In diesem Fall wird man auf die Webseite geleitet, die wir extra dazu erstellt haben, damit Sie die angeführten Beispiele testen können.



Bild 3. Screenshot des Serial Port Monitor bei der Verbindung eines Android-Geräts mit dem Arduino Mega ADK.

fene „Hello World“ auf dem Bildschirm lesen zu müssen. Auf jeden Fall nutzt diese App keine Informationen vom Arduino-Board und schickt diesem auch keine Daten. Alles was sie macht ist, bei Ihrem Android-Smartphone oder Tablet zu überprüfen, ob der Accessory-Mode unterstützt wird.

Hinweis: Prinzipiell ist es möglich, mehr als eine App zu haben, die auf die Daten eines Accessory zugreift. In diesem Fall bietet ein Dialog-Fenster mehr als eine Auswahlmöglichkeit.

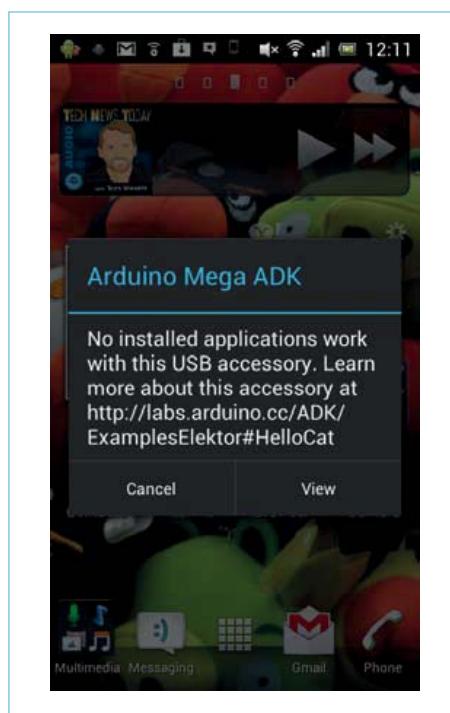


Bild 4. Screenshot eines Android-Smartphones mit der Warnung, dass die vom Accessory verlangte App fehlt.

Im Browser Ihres Android-Geräts sollte sich daher die Webseite von **Bild 5** zeigen.

Man kann die App downloaden und installieren, indem man auf den Link unter dem Bild klickt. Zuerst wird die App herunter geladen und auf das Handy übertragen. Als nächsten Schritt muss man nur noch auf die Datei „Elektor\_MIAU.apk“ klicken, damit die App installiert wird. Wenn die App einmal installiert ist, erscheint beim nächsten Anstecken des Boards eine Meldung mit dem Angebot, die App zu starten (siehe **Bild 6**).

Wenn der App das Booten gestattet wird, präsentiert sie ein GIF-Bild in einer Schleife (siehe **Bild 7**). Wir waren der Meinung, dass es für Sie interessant ist, wenn die grafischen Fähigkeiten Ihres Android-Terminals genutzt werden, statt schon wieder dieses abgegrif-

fene „Hello World“ auf dem Bildschirm lesen zu müssen. Auf jeden Fall nutzt diese App keine Informationen vom Arduino-Board und schickt diesem auch keine Daten. Alles was sie macht ist, bei Ihrem Android-Smartphone oder Tablet zu überprüfen, ob der Accessory-Mode unterstützt wird.

## Android-Setup

Die Erstellung von Apps für Android kann ziemlich kompliziert sein. Man muss dazu eine lange Liste an Software-Paketen aus verschiedenen Quellen installieren. Deshalb haben wir eine Anleitung namens *Getting Started Guide* [3] zusammengestellt, in der alle benötigten Zutaten abgehandelt werden, die für ein möglichst einfaches experimentelles Setup notwendig sind.

Der unkomplizierteste Weg zur Programmierung einer einfachen Android-App führt vermutlich über das Open-Source-Software-Tool *Processing* [4] von C. Reas und B. Fry. Diese Software bietet eine vereinfachte IDE, mit der man Java-Applikationen compilieren kann. Die neueste Version davon kann auch Code für Android-Geräte und Javascript für Webseiten verarbeiten. Wir haben für die Processing-IDE ein Add-on-Tool entwickelt, das Code compiliert und dann für Accessories auf das Handy überträgt.

Nachfolgend die Grundausstattung für das Nachvollziehen der hier beschriebenen Experimente:



UNIQUE  
RESOURCES FROM 

# COMING SOON A NEW ENGINEERING ECO-SYSTEM

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

## (Listing 1)

```
// libraries
#include <AndroidAccessory.h>

AndroidAccessory acc(
  "Arduino, SA",
  "Miau",
  "Arduino Mega ADK",
  "1.0",
  "http://labs.arduino.cc/ADK/
  ExamplesElektor#HelloCat",
  "0000000012345678");

// variable to detect whether the Android device was
// plugged recently
boolean wasConnected = false;
```

```
void setup()
{
  // initialize serial communications
  Serial.begin(115200);
  Serial.print("\r\nStart");

  // initialize the accessory object
  acc.begin();
}

void loop()
{
  if (acc.isConnected() && !wasConnected) {
    Serial.print("Accessory connected. ");
    wasConnected = true;
  }
}
```

- Android SDK mit allen Updates auf die neuesten Versionen der APIs
- Processing 2.0a8 oder neuer
- Arduino ADK Tool für Processing;
- Eclipse [5] + ADT-Plugin (beide optional und hier nicht verwendet)

Wie immer bei der Software-Erstellung gibt es auch andere Tools, mit denen man Apps entwickeln kann. Android SDK ist ein externes Tool zu einem beliebigen Code-Editor, das compiliert, Libraries linkt, simuliert, komprimiert und Android-Apps für eine bestimmte OS-Version signiert.

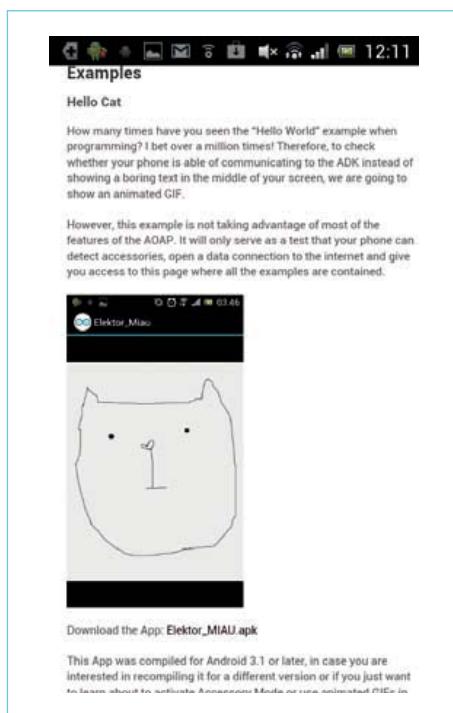


Bild 5. Android-Smartphone mit Information über eine App.

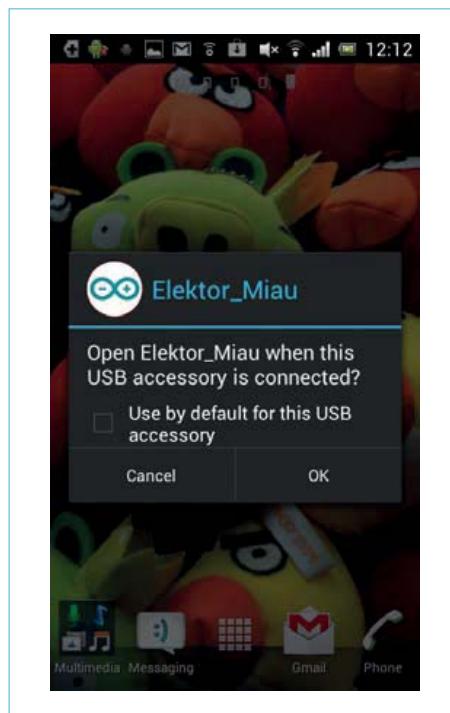


Bild 6. Das Android-Smartphone verlangt nach einer Bestätigung, bevor eine App für das Accessory gestartet wird.

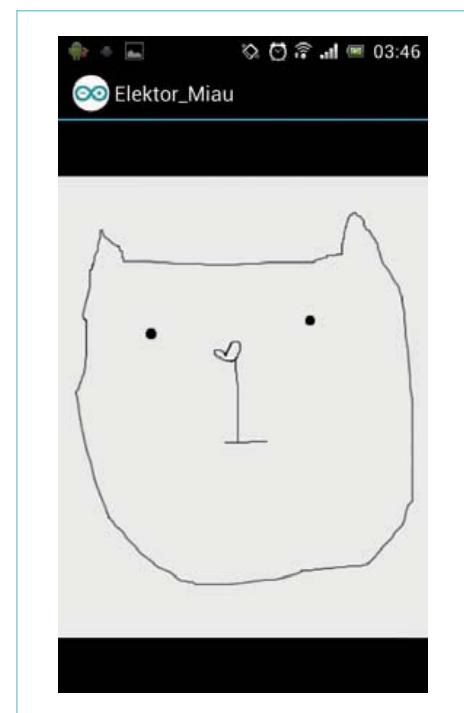


Bild 7: Die MIAU-App unter Android.

**(Listing 2)**

```
void setup() {
  // make sure the screen will have
  // fixed orientation
  orientation(PORTRAIT);
}

void draw() {
  // draw from the center of the shape
  rectMode(CENTER);

  // make a 50x50 pix square using the
  // default color scheme
  rect(width/2, height/2, 50, 50);
}
```

**(Listing 3)**

```
AndroidAccessory acc(
  "Arduino, SA",
  "Elektor_Rectangle",
```

```
"Arduino Mega ADK",
"1.0",
"http://labs.arduino.cc/ADK/
ExamplesElektor#Rectangle",
"000000012345678");
```

Trotzdem ist und bleibt Processing ein gutes Tool, wenn es um den Einstieg in die Programmierung geht. Von daher haben wir dafür auch ein Add-on gebaut, das im *Getting Started Guide* [3] erläutert wird. Doch wenn man Android-Apps auf professionellem Niveau entwickeln möchte, dann führt eigentlich kein Weg an Eclipse [5], der Open-Source-IDE und den zugehörigen Tools vorbei. Auf der Entwickler-Webseite von Android [6] wird Schritt für Schritt erklärt, wie man die Eclipse-basierte Toolchain installiert.

Im Weiteren wird davon ausgegangen, dass Ihnen die Installation von Processing samt Tools gelungen ist. Es handelt sich dabei um Cross-Platform-Software, weshalb sie mit fast jeder OS-Version laufen sollte. Nun zur Software für Android mit Hilfe von Processing.

**Die erste App**

Bevor man daran denken kann, eine App zur Steuerung eines Accessory zu erstellen, sollte man es erst einmal mit einer einfachen App versuchen, die irgendetwas auf den Bildschirm des Android-Geräts bringt. Zuvor muss man selbstverständlich das Android SDK installiert und Processing zur Compilierung von Android-Anwendungen konfiguriert haben (siehe **Bild 8**). Die IDE sollte ein grünes Farbschema aufweisen.

Nun zur App-Programmierung und Übertragung auf das Android-Gerät. Der folgende

Code sollte ein Quadrat in die Mitte des Bildschirms zaubern. Später kommt dann noch Code hinzu, der das Quadrat abhängig von den vom Arduino-Accessory gelieferten Sensordaten verändert (*Listing 2*).

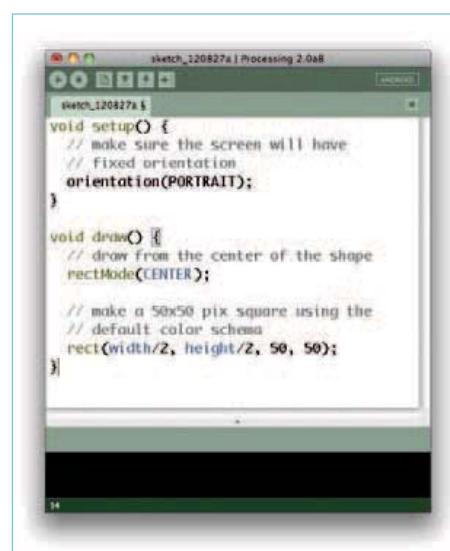


Bild 8. Screenshot von Processing im Android-Mode (siehe den Mode-Knopf in der rechten oberen Ecke).

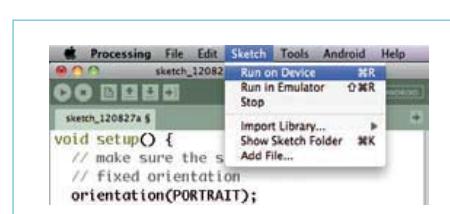


Bild 9. Die App läuft direkt auf einem Smartphone.

Processing ist vor allem für grafisch orientierte Künstler gedacht. Deshalb verwendet es ein Paradigma, in dem anstelle einer Schleife (loop) ein so genanntes „draw“ aufgerufen wird. Die Hauptfunktion des Programms muss daher „draw“ genannt werden. Mit der IDE von Processing wird Java-Code geschrieben, doch gleichzeitig werden alle komplexen Operationen vor dem Anwender verborgen. Um zu überprüfen, ob alles richtig installiert wurde, klickt man wie in **Bild 9** auf den Menü-Eintrag Sketch → Run on Device. Dann sollte der Code kompiliert und automatisch in das Android-Gerät übertragen werden.

Nach dem App-Start sollte sich etwas wie in **Bild 10** auf dem Bildschirm zeigen. Die App sollte direkt starten und dann aktiv bleiben. Man kann sie dann über das App-Menü jederzeit neu starten.

Hinweis: Zur Zeit der Arbeit an diesem Artikel war Processing noch nicht in der Lage, Apps zu signieren. Die Signierung von App validiert diese für die Verteilung auf andere Geräte. Alle mit Processing erstellten Apps laufen daher nur auf dem Gerät, auf das sie direkt hochgeladen wurden. Wenn man

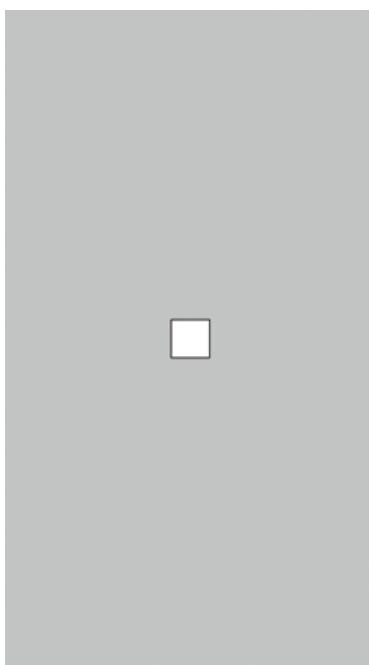


Bild 10. Screenshot der aktiven App auf einem Smartphone.

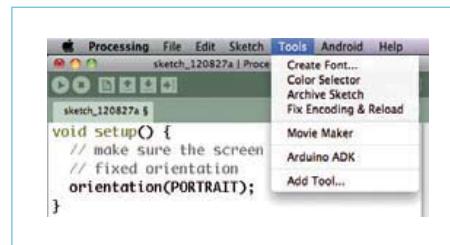


Bild 11. Screenshot des Tools-Menüs der IDE von Processing.



Bild 12. Dialog-Fenster zur Konfiguration der zum Accessory gehörenden App.

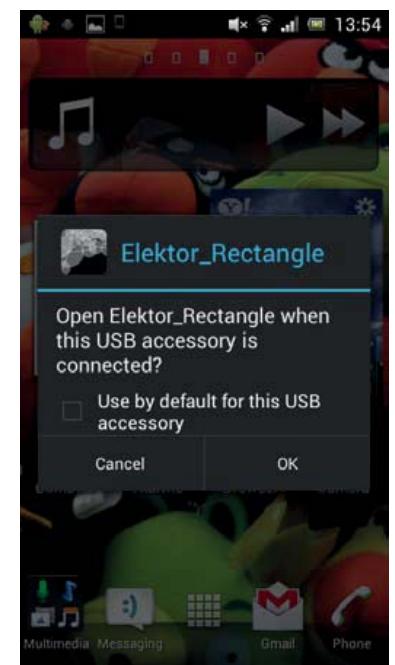


Bild 13. Dialog-Fenster mit der Benachrichtigung zum Öffnen der App mit dem Rechteck.

aber eine eigene App an einen Freund weiterleiten will, dann sollte man sie mit Eclipse compilieren, was aber sehr viel komplexer ist und daher hier nicht beschrieben wird. Die App „Elektor\_MIAU“ ist auf alle Fälle signiert, so dass sie jeder aus dem Internet laden und installieren kann.

## Apps durch Accessory-Anschluss starten

Als nächsten Schritt soll eine App automatisch starten, wenn ein Accessory angeschlossen wird. Hier kommt das Arduino-ADK-Tool für Processing zum Zug. Wenn es wie in [3] beschrieben installiert ist, sollten keine Probleme auftreten, wenn es über die IDE von Processing aufgerufen wird. Das neue für den Upload von Accessory-Code auf das Android-Gerät nötige Tool zeigt sich unter dem Menü-Eintrag Tools → Arduino ADK, wie in **Bild 11** zu sehen.

Wenn diese Option über das Menü ausgewählt wird, dann öffnet die IDE einen Dialog, mit dem man den Namen der App und den Accessory-Hersteller angeben kann (siehe **Bild 12**). Das sind die Infos, die Android-Accessories zur Identifikation der App benötigen, mit der sie sich verbinden sollen. Diese Angaben müssen selbstverständlich mit denen im Code von Arduino übereinstimmen, wie am nachfolgenden Listing 3 zu sehen ist.

Man kann nun den Code von Listing 1 mit der App HelloCat so modifizieren, dass eine neue App aufgerufen wird (*Listing 3*).

Listing 3 ist nur ein Code-Schnipsel zur Modifikation der entsprechenden Stelle in Listing 1, damit die Accessory-App für Android aufgerufen wird.

Wenn das geschehen ist und der Code auf das Arduino Mega ADK hochgeladen wurde, dann wird eine App gestartet, die ein Rechteck zeichnet, wenn man das Board ansteckt (siehe **Bild 13**).

Dies war eine kurze Einführung in die Erstellung von Accessories für Arduino samt passenden Apps für Android. In der nächsten Folge geht es dann weiter mit dem interessanten Thema, wie man Daten zwischen Arduino und Android austauscht.

(120539)

Vielen Dank an dieser Stelle an das Team von Circuits@Home für ihre Arbeit an der USB-Host-Library für Arduino. Ein spezieller Dank geht an Philip Lindsay für sein Debugging der initialen Sammlung offizieller Arduino-Libraries für Android.

Über den Co-Autor: Andreas Goransson ist Software- und UI-Designer aus Malmö. Im Moment lehrt er Software-Entwicklung an der Universität von Malmö und schreibt ein Buch über Android und Arduino (zusammen mit David Cuartielles).

## Weblinks

- [1] Weltweiter Smartphone-Markt nach Betriebssystemen: <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#smartphoneos>
- [2] Accessory Development Kit: <http://developer.android.com/tools/adk/index.html>
- [3] Arduino mit Android @ Arduino.cc: <http://labs.arduino.cc/ADK/AccessoryMode>
- [4] Processing: <http://processing.org>
- [5] Eclipse: <http://eclipse.org>
- [6] Installationsanleitung für das Android SDK mit Eclipse: <http://developer.android.com/tools/index.html>

# Von den Machern von Elektor!

**Elektor**

4 | 2013 [D] 17,50 € [A] 20,00 € [L] 20,00 € [B] 20,00 € CHF 30,00

# Special Project LEDs 4

## Leuchtdioden in Theorie und Praxis

Projekt

**Flash Box – RGB-Controller  
für LED-Stripes**

Praxis

**Kühlkonzept zur Aktivluftkühlung  
eines Hochleistungs-UV-LED-Moduls**

Theorie & Anwendung

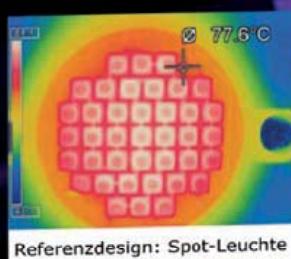
**Optische Charakterisierung  
von Leuchtdioden**

Schaltungstechnik

**LED-Treiber für Scheinwerfer**

News

**LEDs, LED-Treiber und EVA-Boards**



Jetzt neu  
am Kiosk!



Oder frei Haus unter [www.elektor.de/leds4](http://www.elektor.de/leds4) bestellen!

# Zurück zu den Wurzeln (9)

## Sinus-Oszillatoren

Wenn man immer nur bereits vorhandene Signale verarbeiten könnte, wäre die Elektronik um einen Aspekt ärmer: Das Erzeugen von Schwingungen - scheinbar aus dem Nichts. Oszillatoren sind wichtige Bestandteile vieler Geräte und besitzen die unterschiedlichsten Anwendungen. So lassen sich Töne erzeugen, aber auch Testsignale für Schaltungen und Baugruppen.

Von Burkhard Kainka (D)

## RC-Oszillatoren

Jeder kennt das unangenehme Pfeifen, das in Mikrofonanlagen entstehen kann. Das Phänomen tritt bei einer akustischen Rückkopplung zwischen Lautsprecher und Mikrofon auf. Es können Töne unterschiedlicher Höhe entstehen, die sich nur durch einen größeren Abstand der Anlagenteile oder eine geringere Verstärkung verhindern lassen.

Prinzipiell führt jede ausreichend starke Rückkopplung zu Schwingungen. Die Rückkopplung kann auch rein elektrisch erfolgen, indem ein Ausgangssignal an den Eingang zurückgekoppelt wird. Bedingung ist die richtige Phasenlage, wie sie bei einem zweistufigen Verstärker gegeben ist.

Die Schaltung in **Bild 1** ähnelt der eines Multivibrators, allerdings mit einstellbarer Rückkopplung. Während der Multivibrator immer Rechtecksignale liefert, können hier

auch Sinussignale oder andere Kurvenformen entstehen. Über den Lautstärkeregler lässt sich die Rückkopplung so einstellen, dass gerade schwache Schwingungen entstehen. Diese sind dann meist sinusförmig.

Aber es geht auch mit einem einzelnen Transistor, obwohl dieser die Phase um 180 Grad dreht. Die erforderliche zusätzliche Phasendrehung um 180 Grad lässt sich auch mit mehreren hintereinander geschalteten RC-Gliedern erreichen. Der Phasenschieber-Oszillator in **Bild 2** erzeugt ein Sinussignal von ca. 800 Hz, ideal ist dieser als Morse-Übungsgenerator oder als Signalgenerator für den Test von NF-Verstärkern. Auch mit einem Feldeffekttransistor BS170 funktioniert der Phasenschieber-Oszillator prima. Die Schaltung in **Bild 3** wurde sehr hochohmig ausgelegt und liefert eine Frequenz von 10 Hz. Es wird hier nur ein Betriebsstrom von ca. 30  $\mu$ A benötigt.

## Ringoszillatoren

Bisher wurden eine Stufe oder zwei Stufen verwendet. Aber was ist mit drei Stufen im

Emitterschaltung? Man würde eigentlich erwarten, dass aus der Rückkopplung eine Gegenkopplung wird, denn insgesamt wird die Phase um 180 Grad gedreht. Tatsächlich aber entstehen Schwingungen (**Bild 4**). Die erzeugte Frequenz steigt mit der Betriebsspannung und kann bis zu 1 MHz erreichen.

Was passiert hier? Man hat einen gegengekoppelten dreistufigen Verstärker mit sehr hoher Spannungsverstärkung. Jede der drei Stufen verursacht aber gleichzeitig eine kleine Zeitverzögerung. Alle drei Verzögerungen zusammen führen auf einer ganz bestimmten Frequenz zu einer zusätzlichen Phasenverschiebung von noch einmal 180 Grad. Für diese Frequenz wird dann aus der Gegenkopplung eine Rückkopplung. So entstehen Schwingungen. Will man eine solche Schaltung dagegen nicht als Oszillator, sondern als empfindlichen Verstärker einsetzen, muss man sich alle Mühe geben, eine mögliche Rückkopplung zu verhindern. Bei so hoher Verstärkung ist es nicht ganz einfach, Eigenschwingungen zu unterbinden.

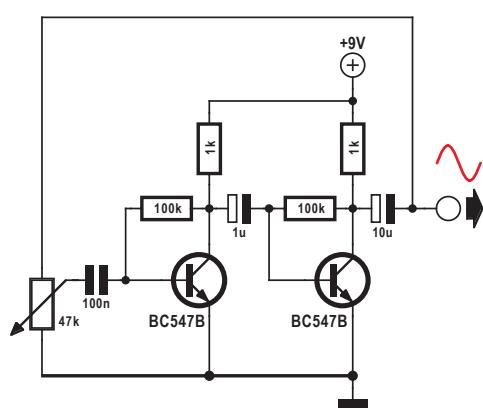


Bild 1. Schwingungszerzeugung durch Rückkopplung.

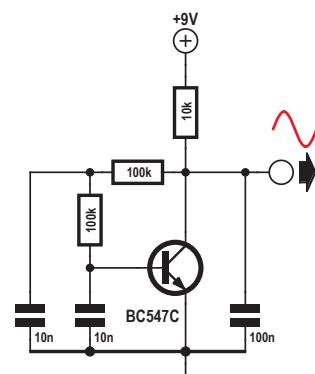


Bild 2. Oszillator mit Phasenschieber.

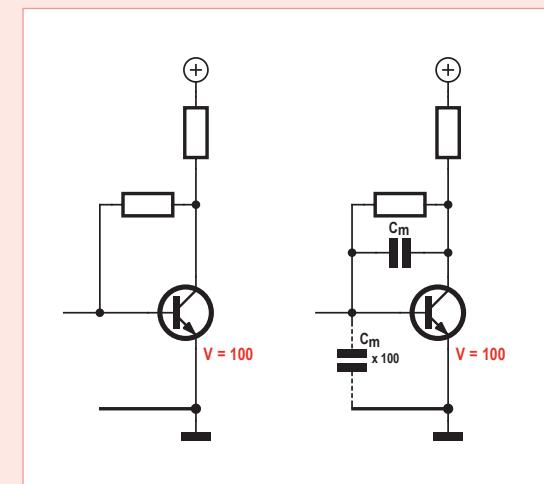
## Der Miller-Effekt

Ein Verstärker in Emitterschaltung hat eine typische Spannungsverstärkung von etwa 100-fach. Das gilt bis zu recht hohen Frequenzen, allerdings manchmal nicht so hoch wie man es sich wünschen würde. Während die Transitfrequenz eines BC547 bei etwa 300 MHz liegt (bei 300 MHz ist der Stromverstärkungsfaktor auf 1 gesunken), ist die Grenzfrequenz dieser Verstärkerschaltung sehr viel geringer. Das gilt besonders, wenn die Widerstände in der Schaltung recht hochohmig ausgelegt sind. Schuld daran sind die internen Sperrsichtkapazitäten des Transistors.

Besonders wirksam ist die Basis-Kollektor-Kapazität  $C_{bc}$ , obwohl sie beim BC547 nur etwa 5 pF beträgt. Das liegt am Miller-Effekt. Der Miller-Kondensator  $C_m$  (=  $C_{bc}$ ) zwischen Eingang und Ausgang des invertierenden Verstärkers wird nämlich von zwei Seiten aus umgeladen. Steigt z.B. die Basisspannung um 1 mV, dann sinkt die Kollektorspannung gleichzeitig um 100 mV. Es muss also etwa die 100-fache Ladung aufgebracht werden. Am Eingang liegt scheinbar ein Kondensator, der um den Spannungsverstärkungsfaktor größer ist als die Miller-Kapazität. In diesem Fall wären das etwa 500 pF. Je nach Innenwiderstand der angeschlossenen Signalquelle bildet sich damit ein Tiefpassfilter, das einem die hohe Grenzfrequenz verhagelt.

Für einen Verstärker heißt das: Wenn es auf eine hohe Grenzfrequenz ankommt, sollte man die Schaltung möglichst niederohmig dimensionieren. Außerdem kann es im Einzelfall sinnvoll sein, mit einer geringeren Spannungsverstärkung zu arbeiten, z.B. indem man den Ausgangswiderstand klein macht. Außerdem kann man mit speziellen HF-Transistoren viel erreichen, die sehr viel geringere Sperrsichtkapazitäten haben.

Für einen Oszillator bedeutet die Miller-Kapazität, dass man einen Oszillator ohne frequenzbestimmende Kondensatoren bauen kann, denn diese bringt der Transistor schon selbst mit.



Einen dreistufigen Oszillator zu bauen ist einfacher als einen dreistufigen Verstärker. Je kleiner der mittlere Kollektorstrom ist, desto hochohmiger wird die Schaltung, und desto stärker wirken sich die internen Kapazitäten des Transistors aus. Deshalb wird bei kleiner Betriebsspannung die Zeitverzögerung größer und die erzeugte Frequenz kleiner.

Eine solche Schaltung bildet einen Ring aus einzelnen Verstärkerstufen und wird deshalb auch als Ringoszillator bezeichnet.

Er funktioniert auch mit fünf, sieben oder neun Stufen. Die Bedingung ist nur, dass für Gleichspannung eine Gegenkopplung besteht. Bei einer geraden Anzahl von Stufen würde dagegen immer ein statisches Flipflop entstehen.

Ein dreistufiger Ringoszillator kann sehr hochohmig und damit sehr sparsam betrieben werden. Mit drei Kollektorwiderständen von 1 MΩ läuft der Oszillator schon mit 0,5 V und braucht weniger als 1 µA. Deshalb reicht sogar schon eine Photodiode

BPW34 im Sonnenlicht als kleine Solarzelle, um den Oszillator zu versorgen (Bild 5). Am Ausgang findet man eine Frequenz von ca. 5 kHz. Die Frequenz steigt mit der Beleuchtung, vielleicht könnte man die Schaltung sogar sinnvoll als Lichtsensor einsetzen.

Nur 5 kHz, und das ganz ohne Kondensatoren? Seltsam, wo doch ein Transistor nur wenige Pikofarad hat. Hier wirkt sich die sogenannte Millerkapazität (siehe Kasten) aus, welche die Kollektor-Basiskapazität um den Faktor der Spannungsverstärkung ver-

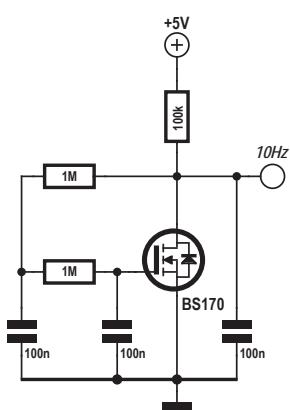


Bild 3. Phasenschieberoszillator mit FET.

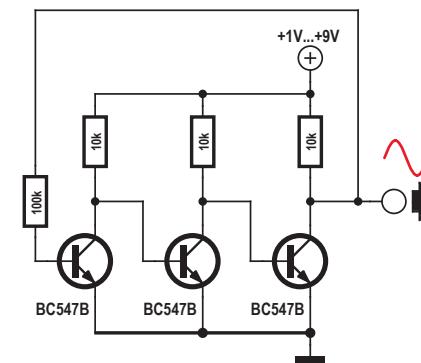
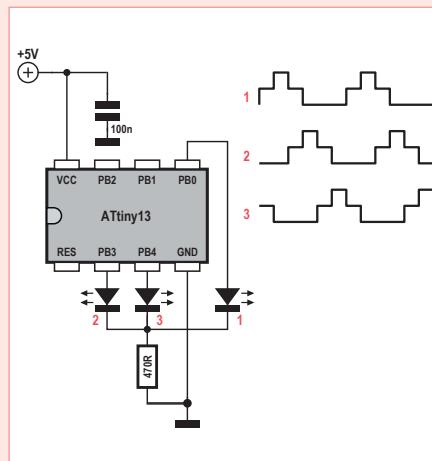


Bild 4. Ein Oszillator ohne Kondensatoren.

## Dreiphasen-Blinklicht



Dreiphasen(Drehstrom)-Signale kann man auch sehr einfach mit einem Mikrocontroller erzeugen. Es gibt dabei insgesamt sechs Umschaltpunkte im gleichen

Zeitabstand. Das Ergebnis an drei LEDs ist ähnlich wie bei einem Dreiphasen-Ringoszillator, nur eben deutlich digitaler.

Eine Besonderheit der Schaltung ist, dass drei LEDs sich hier einen Vorwiderstand teilen. Das führt dazu, dass jede LED mit zwei Helligkeitsstufen arbeitet. Für eine einzelne LED gibt es drei An-Zustände. Im ersten Zustand leuchtet sie zusammen mit der zu ihrer linken und muss sich mit dieser den Strom teilen. Im letzten gilt das gleiche für die LED zu ihrer rechten.

Aber im mittleren Zeitraum leuchtet sie allein und bekommt daher den vollen Strom. Jede LED durchläuft damit die Zustände: Halb, Voll, Halb, Aus, Aus, Aus. Und das jeweils mit einem Phasenunterschied von 120 Grad zur Nachbar-LED.

```
'Dreiphasen-Blinker 1500 ms, 0,67 Hz
$regfile = "attiny13.dat"
$crystal = 1200000
Config Portb = Output
```

```
Do
  Portb.0 = 1
  Waitms 250
  Portb.3 = 0
  Waitms 250
  Portb.4 = 1
  Waitms 250
  Portb.0 = 0
  Waitms 250
  Portb.3 = 1
  Waitms 250
  Portb.4 = 0
  Waitms 250
Loop
```

End

größert an den Eingang legt. Nach diesem Prinzip kann man leicht noch eins drauflegen und mit zusätzlichen Kondensatoren zwischen Basis und Kollektor eine sehr tiefe Frequenz erzeugen (Bild 6). Mit dreimal 100 nF kommt man nun auf ca. 1 Hz.

der Schaltung in Bild 7 gemeinsam eingesetzt werden und ein angenehmes Flackern erzeugen. Es handelt sich um einen Dreiphasen-Oszillator, bei dem jede der drei LEDs zeitversetzt leuchtet. Der LED-Strom ist näherungsweise sinusförmig, sodass sich sanfte Übergänge ergeben.

Strom betreibt, ist das Flackern schnell und hell oder langsamer und weniger hell. Denn auch hier wieder ist die Frequenz stark vom Betriebsstrom abhängig.

(120009)

### Dreiphasen-LED-Blinker

Bei so kleinen Frequenzen lassen sich schöne Lichteffekte erzielen. Drei LEDs sollen in

Je nachdem ob man die Schaltung direkt an 9 V oder über das Poti mit gebremstem

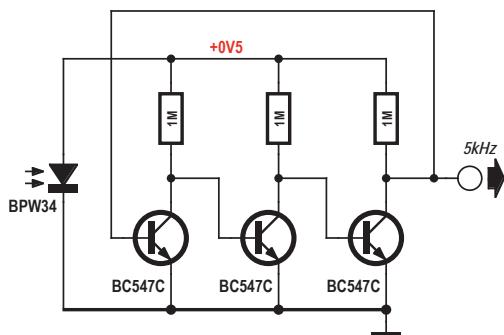


Bild 5. Ringoszillator mit Solarzelle.

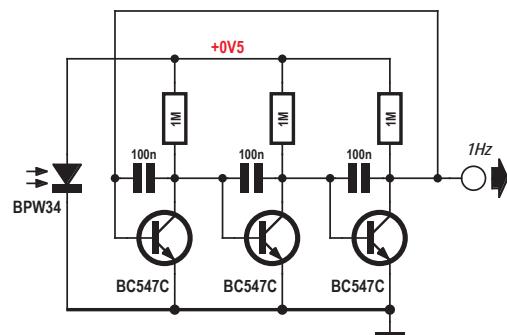


Bild 6. Langsamer und sparsamer Ringoszillator.



die welt braucht  
innovative  
elektronik.  
hier wird sie  
gezeigt.

## Quiz-Lösung aus dem Septemberheft

Der richtige Lösungscode lautet „CFG“. Hier die Erklärung:

### Lösung 1:

Eine vollständige Periode dauert  $T = 2 \times 0,65 \text{ ms} = 1,3 \text{ ms}$ .  
 $f = 1 / T = 1 / 1,3 \text{ ms} = 0,769 \text{ kHz}$ , also ca. 770 Hz (Lösung C ist richtig).

### Lösung 2:

Die Basisspannung sinkt weniger tief ab und erreicht schneller wieder +0,6 V. Die Frequenz steigt also (F).

### Lösung 3:

Der Emitterstrom soll größer werden, also wird der Emitterwiderstand verkleinert (G).

Damit steigt auch der Strom durch den Lautsprecher.

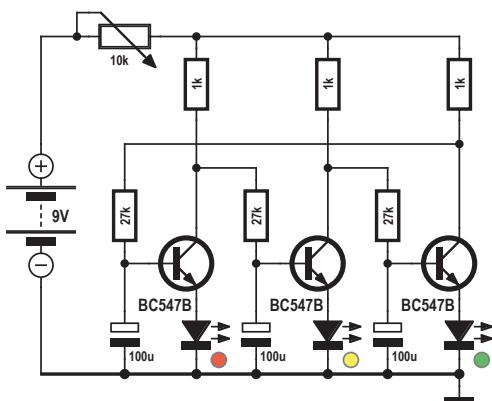
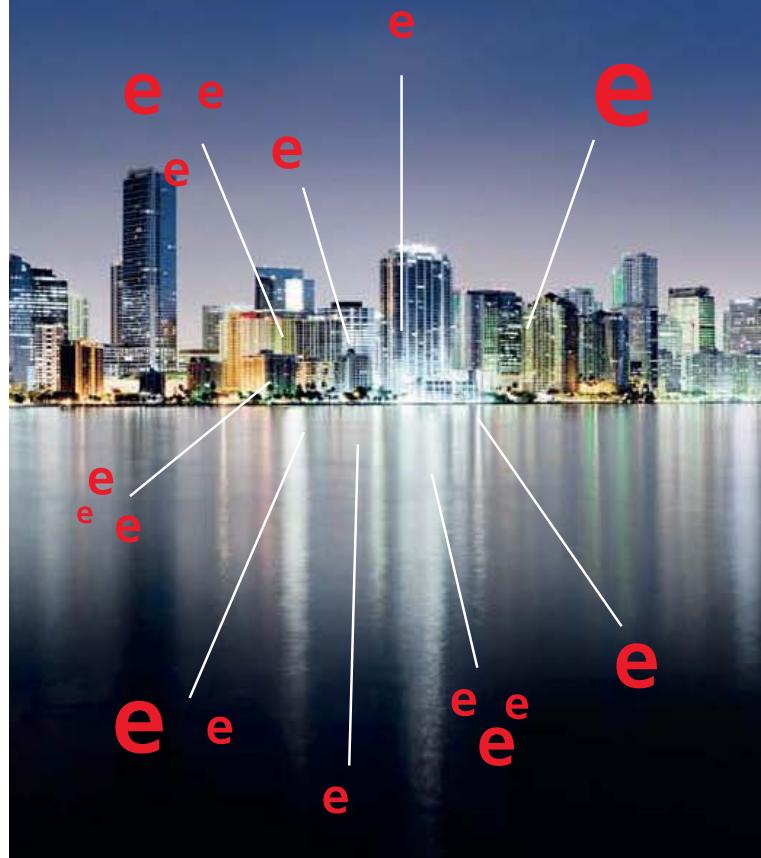


Bild 7. Dreiphasen-Lichteffekte.



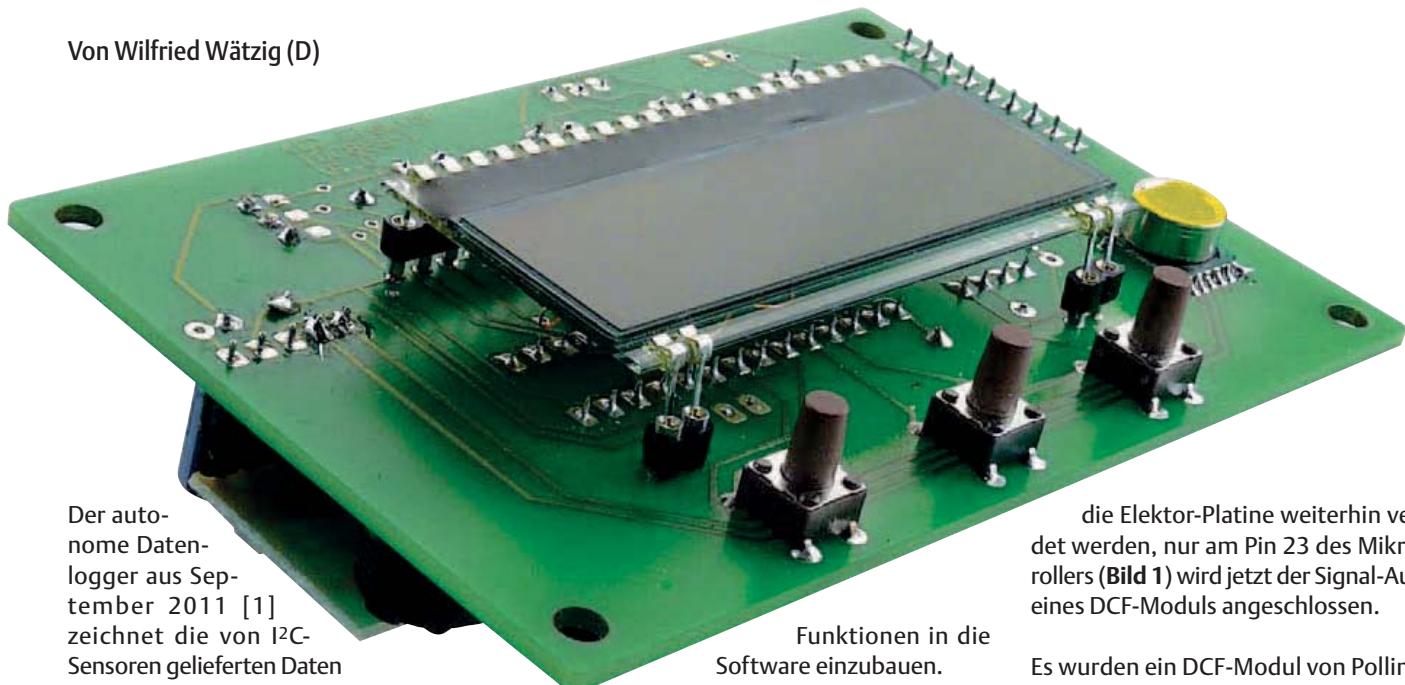
**electronica** 2012  
inside tomorrow

25. Weltleitmesse für Komponenten,  
Systeme und Anwendungen der Elektronik  
Messe München  
13.–16. November 2012  
www.electronica.de

# USB-Wetterlogger mit DCF-Modul

Auf Nachfrage von mehreren Elektor-Lesern hat der Autor des USB-Wetterloggers mit Langzeitspeicher (Elektor September 2011) eine erweiterte Version mit einem DCF-Modul erstellt.

Von Wilfried Wäting (D)



Der autonome Datenlogger aus September 2011 [1] zeichnet die von I<sup>2</sup>C-Sensoren gelieferten Daten

für Luftdruck, Temperatur und Feuchte auf und zeigt diese auf einem LC-Display an. Die Ergebnisse lassen sich über USB auslesen und mit GNUpot auf einem PC grafisch darstellen. Weil digitale Sensormodule verwendet werden, ist der Hardwareaufwand gering und ein Abgleich nicht erforderlich. Die Messwerte werden in regelmäßigen Zeitabständen erfasst und in einem serialen EEPROM gespeichert, so dass sie auch ohne Betriebsspannung erhalten bleiben. Das Messgerät funktioniert sechs bis acht Wochen mit drei Mignonzellen. Über ein Seriell-USB-Modul ist das Auslesen in einen auswertenden PC möglich.

Sehr viele Leser haben seit der Veröffentlichung im September 2011 diesen Wetterlogger nachgebaut (Elektor liefert dazu ein Kit, bestehend aus Platine, programmiertem Controller, Feuchte- und Luftdrucksensor). Verschiedene Leser haben beim Autor den Wunsch geäußert, nachträglich noch ein DCF-Modul anschließen zu können. Das erforderte aber auch eine Anpassung der Software. Der Autor hat diese Modifikation gleich dazu genutzt, noch einige Extra-

Funktionen in die Software einzubauen. Auch nach der Modifikation kann

die Elektor-Platine weiterhin verwendet werden, nur am Pin 23 des Mikrocontrollers (Bild 1) wird jetzt der Signal-Ausgang eines DCF-Moduls angeschlossen.

Es wurden ein DCF-Modul von Pollin (Kontakt PON an Masse) und ein DCF-Modul

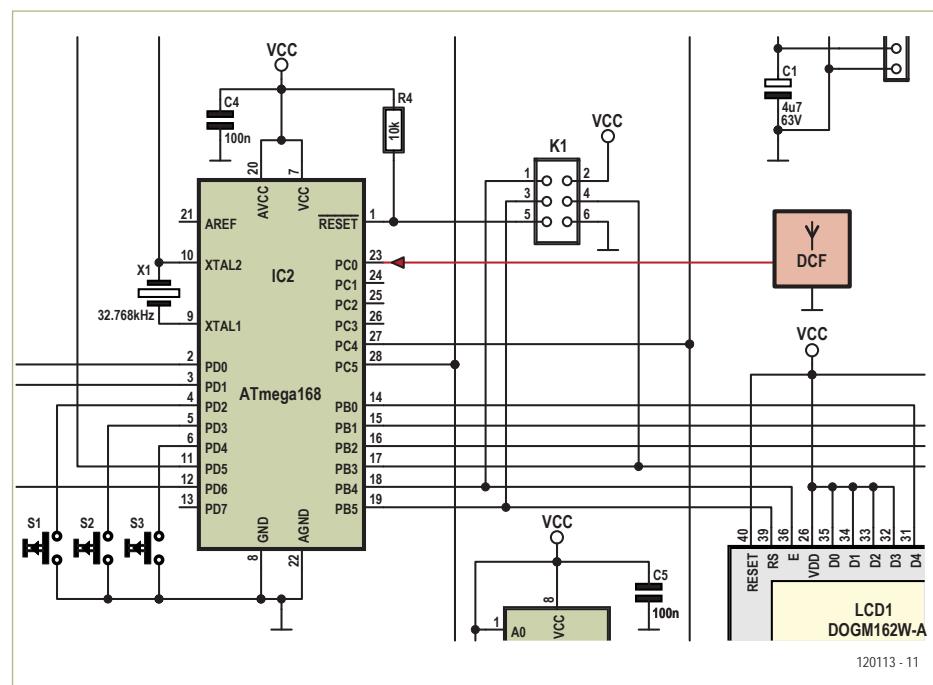


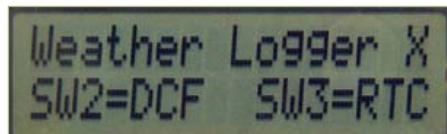
Bild 1. Update: Neuer Mikrocontroller mit DCF-Modul.

von Conrad getestet. Beide Module reagieren empfindlich auf Störungen, die durch Schaltnetzteile, Energiesparlampen etc. hervorgerufen werden. Man muss mit mehreren Minuten rechnen, bis das Modul ein sauberes Signal liefert, sodass das Programm die DCF-Informationen entschlüsseln kann. Alternativ kann der Mikrocontroller auch die integrierte Realtime-Clock (RTC) nutzen.

### Erweiterte Funktionen

Wegen des erweiterten Programm-Codes (jetzt fast 12 KB) muss der **ATmega88** durch einen (pinkompatiblen) **ATmega168** ersetzt werden. Das neue Programm (Version 1.1) für den Mikrocontroller kann von der Elektor-Projektseite [2] heruntergeladen werden. Auch ein vorprogrammierter ATmega168 ist lieferbar (Bestellnummer 120113-41 [2]). Die Fuses des ATmega168 bleiben dieselben wie beim ATmega88 (siehe **Tabelle 1**).

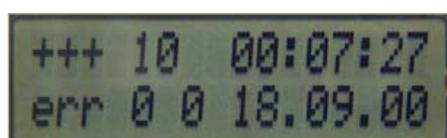
Nach dem Einschalten des Wetterloggers wird der Zeit-Modus abgefragt:



#### DCF-Uhr:

Während der Synchronisation auf das DCF-Signal wird der 1-Sekunden-Impuls der Realtime-Clock benutzt.

Ausgabe auf dem Display:



+++ xx hh:mm:ss xx=DCF-Zähler,  
hh:mm:ss=Uhrzeit  
err c b DD:MM:YY c=DCF-Error-Counter,  
b=DCF-Bit, DD:MM:YY=Datum  
Wenn die DCF-Uhr synchronisiert ist,  
erscheint das Symbol „DCF“ statt „+++“.

Mit S1 geht's zurück zum Hauptprogramm. **Tabelle 2** zeigt eine Übersicht aller Einstellungen und Funktionen (zugänglich mit den

Tasten S1...S3).

Bei den interaktiven Kommandos zur Datenabfrage gibt es jetzt ein weiteres Ausgabe-Format, bei dem der Monat/Tag/Stunde/Minute ausgegeben werden (**Tabelle 3**). (120113)

### Weblinks

- [1] [www.elektor.de/100888](http://www.elektor.de/100888)
- [2] [www.elektor.de/120113](http://www.elektor.de/120113)

**Tabelle 1. Einstellung der Fuses für den ATmega168**

Fuses:	EXT.	0xF9	8 MHz internal oscillator : 8 => CPU-CLOCK = 1 MHz
	HIGH	0xDF	CKDIV8 enabled, brown-out disabled
	LOW	0x62	65 ms startup

**Tabelle 2. Einstellungen und Funktionen (Taster S1...S3)**

Funktionsauswahl mit S1	Funktion S2	Funktion S3
	Wahl DCF-Uhr	Wahl Realtime-Uhr
0: normale Anzeige	zeigt Messwert Druck	zeigt Messwert Feuchtigkeit
1: Stellen der Uhr	erhöht Stunden	erhöht Minuten
2: Setzen M/N	erhöht M: 0..6	löscht N
3: UART-Kontrolle	weiter	Exit
4: Anzeige Messwerte	weiter	Exit

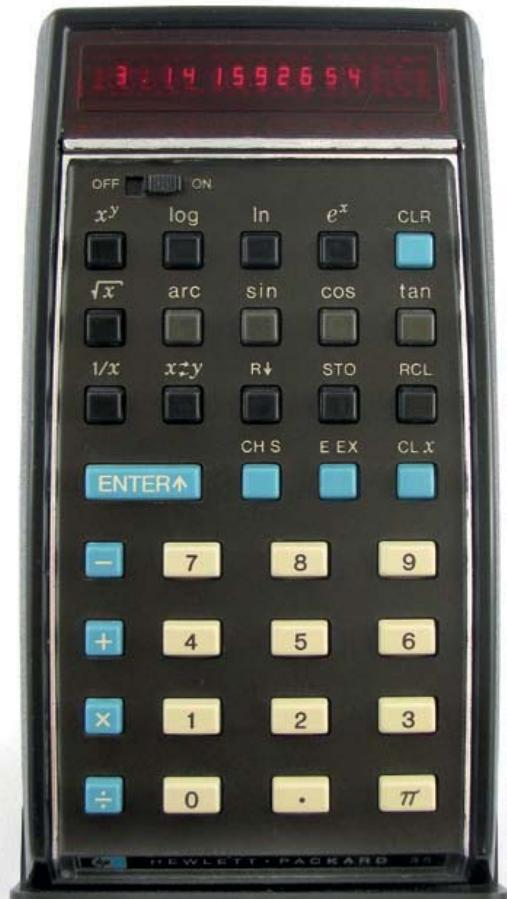
M = Messungen pro Stunde; N = Anzahl der Messungen

**Tabelle 3. Interaktive Kommandos zur Datenabfrage**

Befehl	Funktion
h =help	drückt die verfügbaren Kommandos, z.B. # h=help/a=show-p/p#=print#/m#=set-fm#/c=clear/x=exit
a =show-p	drückt die Anzahl der Messungen
p# =print#	p0 drückt die Messwerte mit Bezeichnung, z.B. 123 12:30:00 T= 25.6 degC H=43% P= 987.6 hPa p1 drückt nur die Zahlen, z.B. 123 12:30:00 256 43 9876 p2 drückt die Zeitachse in Stunden und die anderen Werte ohne Bezeichnung im Eingabeformat für GnuPlot, z.B. 68.50 256 43 9876 p3 drückt Datum (Monat/Tag) und Uhrzeit (Stunde/Minute) mit den Werten (für Excel), z.B. 10 23 12 30 256 43 9876
m# =set-fm#	setzt die Anzahl der Messungen pro Stunde #=0..6 (60/30/20/15/12/10 Minuten) m0 keine Messung m6 Messung 6 Mal pro Stunde, d.h. alle 10 Minuten
c =clear	löscht die Anzahl der Messungen N
x =exit	beendet die serielle Verbindung

# HP-35: Revolution in der Tasche (1972)

Von Dipl.-Inf. Karl-Ludwig Butte (D)



1

Das war 1968. Mit dem HP-9100A hat Hewlett-Packard Neuland im Bereich elektronischer Rechner betreten. Doch beim damaligen Preis von 4.900 \$, was nach heutiger Kaufkraft [1] etwa 33.000 \$ entspricht, musste ein elektronischer Rechner für viele Ingenieure ein Traum bleiben. Sie mussten also weiter mit dem Rechenschieber vorlieb nehmen.

Nur vier Jahre später war alles anders. Mit Hilfe der von Mostek [2] entwickelten stromsparenden ICs gelang es Hewlett-Packard, am 1.2.1972 den ersten wissenschaftlichen Taschenrechner der Welt

## Damals in den wilden Siebzigern...

Der HP-35 erschien in einer aufregenden Zeit. Den Anfang des Jahres 1972 markierten die Apollo-Mond-Missionen 16 und 17 der NASA. Mariner 9 schickte Bilder vom Mars und Pioneer 10 war gerade Richtung Jupiter und Asteroidengürtel unterwegs. Im September wurde die Öland-Brücke in Schweden eröffnet [3], die damals die längste Brücke Europas war. Und natürlich brachte der Anfang der 1970er Jahre große Fortschritte in Sachen ICs und in deren Folge die Taschenrechner mit sich [4].

zu präsentieren. Das Team von Tom Osborne benötigte lediglich 1,5 Jahre für eine Antwort auf Bill Hewletts Herausforderung. Sie entwickelten in dieser Zeit tatsächlich einen wissenschaftlichen Rechner im Taschenformat. Trotzdem war er zu Anfang bei einem damaligen Preis von 395 \$ (etwa 2.600 \$ nach heutigem Wert) lediglich für Ingenieure in größeren Firmen erschwinglich. Doch die Überlegenheit des Taschenrechners und der Niedergang des Rechenschiebers waren damit unumkehrbar eingeläutet.

## Die Hardware

**Bild 1** zeigt den Rechner mit den Abmessungen 147 x 81 x 33 mm und einem Gewicht von nur 246,6 g. Unter dem einzeiligen Display mit 15 Stellen befanden sich der Ein-/Aus-Schalter und eine Tastatur mit 35 Tasten. Nach dem Willen von Bill Hewlett war die Tastenzahl namensgebend für den Rechner.

Das Sieben-Segment-Display enthält für jede Stelle eine kleine Linse, womit sich die Lesbarkeit durch Vergrößerung verbessert. LED-Anzeigen glänzen nicht gerade durch geringen Stromverbrauch. Deshalb wurde der HP-35 mit einem Akku-Satz aus drei AA-Zellen betrieben, die in einem eigenen Kunststoffgehäuse steckten (**Bild 2**). Die Kontaktflächen für den positiven und negativen Anschluss wurden unsymmetrisch an der Seite des Gehäuses platziert, sodass man das Paket nicht irrtümlich verpolen konnte. Der Rechner kam mit einer Transporthülle, einem Netzteil und einem Handbuch. Alle Teile zusammen waren in einer beeindruckend gro-

Das Modell HP-35 war der erste wissenschaftliche Taschenrechner der Welt. Schlagartig gehörten nervige und ungenaue Berechnungen mit Hilfe von Tafeln und Tabellen für Trigonometrie und Logarithmen der Vergangenheit an, denn der HP-35 konnte nicht nur Wurzelziehen, sondern verfügte auch über Winkelfunktionen und konnte sogar Logarithmen berechnen – und das alles nicht nur blitzschnell auf Knopfdruck, sondern auch noch mit einer Genauigkeit von zehn Stellen!

## Der Rechenschieber ist tot: Lang lebe der HP-35!

„Das sollten wir mit einem Zehntel des Volumens, zehn Mal so schnell und zu einem Zehntel des Preises hinbekommen.“ Dieser Spruch wird Bill Hewlett als Reaktion auf die Demonstration des Tischrechners HP-9100A zugeschrieben, der mit gut 18 kg ein Monster von Apparat war. Die Worte waren als Glückwunsch an das Entwicklungs-Team gedacht!

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

# Der erste wissenschaftliche Taschenrechner von Hewlett-Packard wird 40

Ben Schachtel untergebracht. Leider fehlen mir genau die letzten beiden Dinge in meiner umfangreichen Sammlung (Bild 3).

Bild 4 zeigt das Innenleben des HP-35 (Mainboard rechts, Tastatur und Display-Platine links). In Bild 5 ist die Blockschaltung zu sehen. Es sitzen insgesamt fünf MOS-LSI-ICs auf dem 70 x 75 mm messenden Mainboard. Drei davon sind ROMs mit der Firmware des Rechners. Die restlichen beiden Chips enthalten die Arithmetik- und Registereinheit sowie die Steuer- und Timing-Einheit. Das andere Board enthält die Tastatur sowie das Display samt Treiber-Elektronik. Die Arithmetik- und Registereinheit besteht aus fünf Blöcken: Befehlsspeicher und Decodierer, Timer, sieben Register mit je 56 bit, Addierer/Subtrahierer und Display-Decoder [5].

Die Steuer- und Timer-Einheit ist für die Koordination der internen Operationen zuständig. Hierzu gehören die Abfrage der Tastatur, die allgemeine System-Synchronisation und die Erzeugung der Befehlsadressen [5].

In den drei ROMs stecken die Routinen zur Berechnung mathematischer Funktionen. Jedes ROM fasst 256 Befehle mit 10 bit Breite. Außerdem enthält jedes ROM einen Decodierer. Der Gesamtspeicher für alle Rechnerfunktionen umfasst somit 7,5 kbit. Wie konnten Tom Osborne und sein Team so viel Logik in so wenig Speicherplatz quetschen? Nachfolgend die Antwort auf diese Frage.

## Die Software

Zwei wesentliche Aspekte erlaubten einen so kompakten Code: Der erste Punkt war die Verwendung von UPN (Umgekehrte Polnische Notation) für die Eingabe der zu berechnenden Parameter und Operationen. Der zweite Punkt war die Verwendung so genannter CORDIC-Algorithmen.

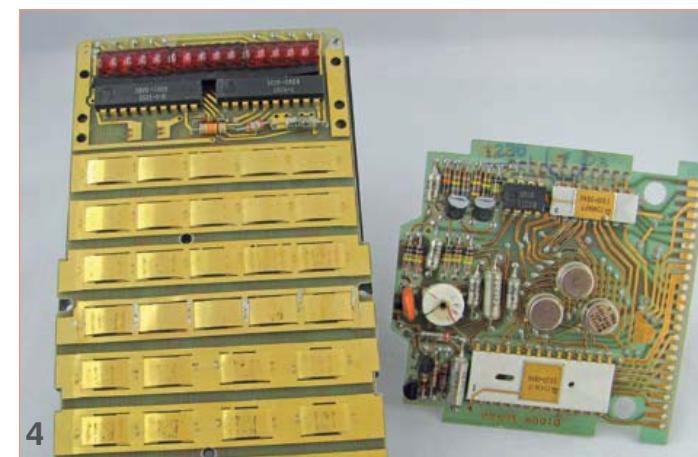
Die UPN ist eine Abwandlung der so genannten „polnischen Notation“ (eine Entwicklung des polnischen Mathematikers Jan Łukasiewicz), welche die klammerfreie Eingabe von Ausdrücken ermöglicht [6]. Normalerweise wird zuerst die Operation gefolgt von den beiden Operanden eingegeben. Beim HP-35 und der UPN wird genau anders herum verfahren: Zuerst kommen die beiden Operanden in den Vier-Ebenen-Stack des Rechners mit Hilfe der großen Enter-Taste, und dann erst wird die Taste mit der gewünschten Operation betätigt. Aus diesem Grund findet man auf der Tastatur des HP-35 weder ein Gleichheitszeichen noch Tasten für Klammern. So etwas ist bei UPN schlicht überflüssig. In den 1970ern – ganz besonders seit Erscheinen des Modells HP-65 – wurde die Debatte bei Taschenrechnern von den Vor- und Nachteilen von HPs UPN versus AOS (Algebraic Operating System) von Texas Instruments beherrscht, was zeitweise den Charakter quasi-religiöser Kriege annahm, ganz wie später bei den Diskussionen über Programmiersprachen (BASIC versus Pascal und später C). Im Internet finden sich viele Beschreibungen der eleganten UPN-Eingabe-Methode [7]. Mit UPN konnte man sich nicht nur die drei Tasten „=“, „(“ und „)“ sparen, sondern damit war auch eine Vereinfachung der Software verbunden, die Ausdrücke analysiert. Der Anwender wurde außerdem nicht mit Dingen wie der maximalen Verschachtelungstiefe von Klammerebenen belästigt. Weiter musste man sich als Programmie-



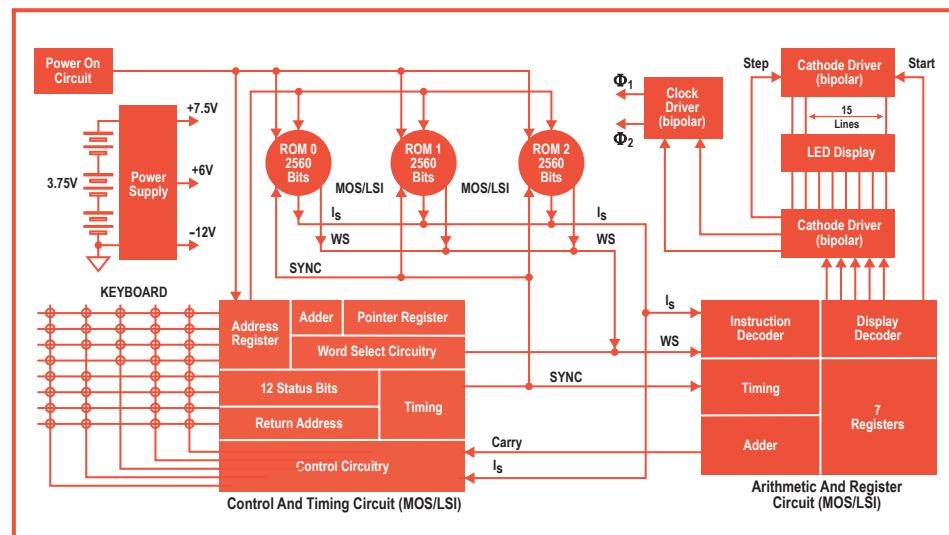
2



3



4



rer nicht mit einer begrenzten Stack-Tiefe beschäftigen.

Nun zum zweiten Punkt, den CORDIC-Algorithmen: CORDIC ist ein Akronym für „COordinate Rotation Digital Computer“ und bezieht sich auf eine Klasse effektiver iterativer Algorithmen für trigonometrische und andere Berechnungen, die von Jack E. Volder 1958 entwickelt wurden [8]. Volder gelang die Digitalisierung bislang unzuverlässiger Navigationsrechner im B58-Bomber von Convair, mit der dann die aktuelle Position in Echtzeit berechnet werden konnte. Echtzeitfähigkeit war schon allein deshalb von Relevanz, da es sich bei der B58 um einen Überschall-Flugzeug mit Strahltriebwerken handelte.

Tom Osborne hatte sich schon beim HP9100A mit CORDIC-Algorithmen beschäftigt und daher viel Know-How zu diesem Thema angesammelt. Er wusste also schon, wie man Multiplikation, Division und Trigonometrie auf einfache Operationen wie Addition und Shift-Operationen zurückführt. Die erzielte Genauigkeit braucht übrigens keinen Vergleich mit derjenigen moderner Rechner zu scheuen, auch wenn sich hie und da kleine Fehler in der letzten Stelle zeigen. Die Berechnung von  $4^3$  ergibt beim HP-35 beispielsweise 63.99999997 statt exakt 64.

## Bugs

Trotz der großen Sorgfalt schllichen sich damals einige Bugs in die Software ein. Der erste wurde entdeckt, nachdem schon mehr als 25.000 Exemplare des Rechners verkauft worden waren [9]. Die Funktion  $\text{eln } 2.02$  ergab 2 statt 2.02 und  $\tan^{-1} 0.0002$  ergab  $5.729577893 \times 10^{-3}$  statt 0.01145916 [10]. Das war vor der Erfindung überschreibbaren Flash-Speichers ein recht kniffliges Problem. HP konnte ja nicht einfach ein Firmware-Update ins Internet stellen und so das Problem aus der Welt schaffen, wie das heute üblich ist. In einer Krisensitzung brachte Dave Packard die Sache auf den Punkt: Was sollte HP mit den bereits ausgelieferten Rechnern machen? Der berühmte Vorschlag eines Kollegen „Nichts sagen!“ war natürlich nicht akzeptabel [9]. Stattdessen wurden alle Käufer über den Fehler informiert und es wurde angeboten, diesen kostenlos zu reparieren (siehe [11] auf Seite 6 und die Bilder 5 und 6). Interessanterweise kam nur rund ein Viertel der betroffenen Geräte zwecks Reparatur zurück. Die meisten Kunden behielten ihr Gerät trotz des Fehlerhinweises so wie es war. Manche bestellten einfach einen neuen HP-35 mit korrigierter Software.

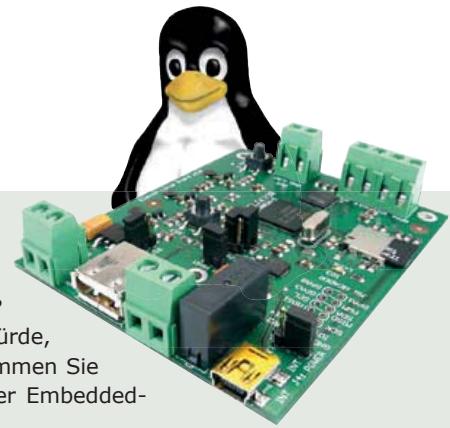
## Der Siegeszug des Taschenrechners

Beim HP-35 handelt es sich um eines der erfolgreichsten Produkte in der Geschichte von HP. Am 14.4.2009 erhielt HP dafür den begehrten Award „IEEE Milestone in Electrical Engineering and Computing“ [12]. Ein Jahr nach dem HP-35 erschien sein Nachfolger HP-45 und schon 1974 kam das Modell HP-65 auf den Markt, bei dem es sich um den ersten programmierbaren Taschenrechner der Welt handelte, der seine Programme sogar auf Magnetkarten speichern konnte. Insgesamt entwickelte HP mit der Zeit mehrere hundert unterschiedliche Taschenrechner und diese wurden auch von vielen Nobelpreis-Trägern verwendet. Selbst wenn der Hype um Taschenrechner seit der allgemeinen Verbreitung von PCs komplett abgeklungen ist, werden doch immer wieder neue Modelle mit besseren und grafischen Fähigkeiten entwickelt, die sogar symbolische Manipulationen erlauben. Auch außerhalb der Wissenschaft wurden besonders für das Finanzwesen spezialisierte Modelle entwickelt. Schon 1973 wurde von HP das Modell HP-80 als „Business-Rechner“ vorgestellt. Mit dem HP-12C gelang HP ein weiterer Rekord im Bereich Taschenrechner: Dieser Finanz-Rechner wird ununterbrochen seit 1981 produziert und das sind für ein technisches Produkt geradezu unglaubliche 31 Jahre!

(120454)

## Weblinks und Referenzen

- [1] „What is a dollar worth?“ [www.minneapolisfed.org](http://www.minneapolisfed.org)
- [2] Malone, Michael S.: „Bill & Dave“, Portfolio (Penguin Group) New York, 2007
- [3] <http://en.wikipedia.org/wiki/1972>
- [4] [http://en.wikipedia.org/wiki/Integrated\\_circuit](http://en.wikipedia.org/wiki/Integrated_circuit)
- [5] HP Journal, Juni 1971, Seite 2 ff.: [www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1972-06.pdf](http://www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1972-06.pdf)
- [6] [http://en.wikipedia.org/wiki/Polish\\_notation](http://en.wikipedia.org/wiki/Polish_notation)
- [7] <http://h41111.www4.hp.com/calculators/uk/en/articles/rpn.html>
- [8] <http://en.wikipedia.org/wiki/CORDIC>
- [9] [www.hpmuseum.org/hp35.htm](http://www.hpmuseum.org/hp35.htm)
- [10] [www.jacques-laporte.org/HP35%20ROM.htm](http://www.jacques-laporte.org/HP35%20ROM.htm)
- [11] <http://hhuc.us/2007/Remembering%20The%20HP35A.pdf>
- [12] [www.ieee.org/about/news/2009/14april\\_1.html](http://www.ieee.org/about/news/2009/14april_1.html)



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

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

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

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

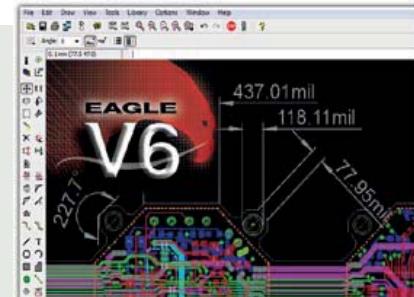
In diesem 3-tägigen Kurs werden Sie die Programmierung von „Eingebetteten Systemen“ in der Programmiersprache C kennenlernen. Dazu wird auf die PIC-Mikrocontroller-Familie der Firma Microchip anhand des C18-/C30-Compilers eingegangen. Das Hauptaugenmerk liegt auf der C-Sprache, um sich besser auf die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

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

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

### Eagle PCB und Design (2-tägiges Seminar)

In diesem 2-tägigen Kurs werden Sie lernen, wie man mit dem Programm Eagle der Firma Cadsoft GmbH Leiterplatten entflechten kann. Begonnen wird mit dem Zeichnen von Schaltplänen unter Verwendung von Standard-Eagle-Bibliotheken. Sie lernen, wie man Schaltpläne über mehrere Seiten hinweg zeichnet und wie man eigene Bibliotheken und Bauteile erstellt. Anschließend werden Sie lernen, wie man aus dem Schaltplan eine Leiterplatte definiert und diese dann entfleitet (layoutet). Dabei werden auch Spezialkenntnisse zum Layouten von HF-Schaltungen, wie Leitungswellenwiderstand, vermittelt. Nach erfolgreichem Layout werden Produktionsdaten erzeugt, die man benötigt, wenn man die Platine fertigen lassen möchte. Dabei wird auch auf die verschiedenen Produktionsarten wie fräsen und ätzen eingegangen. Zum Abschluss gibt es Tipps und Tricks zum Umgang mit Eagle.



Referent: Prof. Dr.-Ing. Francesco P. Volpe – Teilnahmegebühr: 849,00 € (inkl. MwSt.)

# Hexadoku

## Sudoku für Elektroniker

Hier kommt wieder eine neue hexadezimale Herausforderung – für all jene, die Abwechslung von all den Schaltplänen und Listings brauchen. Tun Sie also mal etwas anderes für Ihre kleinen grauen Zellen – und füllen Sie in die grauen Zellen auf dem Papier die richtigen Zahlen ein. Denn hier warten vier Gutscheine auf die Gewinner!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist.

Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass **alle** Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert

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

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

### Mitmachen und gewinnen!

Unter allen internationalen Einsendern mit der richtigen Lösung verlosen wir

einen 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. November 2012!

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

Die richtige Lösung ist: 3F126.

Der Elektor-Gutschein über 100 € geht an: William Neumann aus Leduc (Kanada).

Einen Elektor-Gutschein über je 50 € haben gewonnen: Yves Printemps, Eugene Stemple und Jean-Louis Vidaud.

Herzlichen Glückwunsch!

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

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

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

 <p>Entwicklung industrietauglicher Software und Hardware sowie Elektronik 03303/212166 oder <a href="http://www.jasys.de">www.jasys.de</a></p>	  <p>Gravuren / Schilder / Frontblenden <a href="http://www.webgrav.de">www.webgrav.de</a></p>	 <p>LCDs und mehr <a href="http://www.lcd-store.de">www.lcd-store.de</a> <a href="http://www.LC-Design.de">www.LC-Design.de</a> <a href="http://www.crystalfontz.de">www.crystalfontz.de</a></p>	<p>Leiterplatten für Pragmatiker!</p>  <p><a href="http://wwwpcb-joker.com">wwwpcb-joker.com</a></p>
<p><b>Alles Spule!</b></p> <p>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.</p> <p><b>MM</b> Menting Mikroelektrik <a href="http://www.spulen.com">www.spulen.com</a></p>	<p><b>Bausätze zu ELEKTOR</b> 1986 bis heute!</p> <p>Teilesätze, Platinen, programmierte Kontroller sowie Cds zu fast allen Elektor-Projekten vom Spezialist. Alle Elektor-Artikel zum Verlagspreis.</p> <p>Ihr zuverlässiger Partner für aktive und passive elektronische Bauteile und Komponenten:</p> <p><b>Geist Electronic- Versand GmbH</b> Tel.: 07720/36673 Fax: 07720/36905 Mail: <a href="mailto:info@geist-electronic.de">info@geist-electronic.de</a> Shop: <a href="http://www.geist-electronic.de">www.geist-electronic.de</a></p>	<p><b>LOETRONIC</b> Embedded MP3 Module <a href="http://www.loetronic.com">www.loetronic.com</a></p> <p><b>SCOPES</b> und mehr</p> <p><b>HAMEG</b>® Instruments A Rohde &amp; Schwarz Company</p>	<p><b>MESSTECHNIK</b> zum fairen Preis</p>
<p>Auch Ihr Unternehmen ist eine Anzeige wert!</p>			<p><a href="http://www.antronic.de">www.antronic.de</a> ab 1 Stck. <b>ANT</b> TRONIC Leiterplatten zu TOP-Preisen!!</p>
<p><b>Günstige Platinen clever bestellen!</b></p>  <p><a href="http://wwwpcb-joker.com">wwwpcb-joker.com</a></p>	<p><b>DEIWO</b>® Dienstleistungen für die Elektronik-Industrie</p> <ul style="list-style-type: none"> <li>- Bestückung SMD und THT             <ul style="list-style-type: none"> <li>- Verdrahtung</li> <li>- Kabelkonfektionierung</li> <li>- Baugruppenreinigung</li> <li>- Verguss, Blech, Biegeteile</li> <li>- mechanische Baugruppen</li> <li>- CNC Gehäuse-Bearbeitung</li> <li>- Montage und Endprüfung</li> </ul> </li> </ul> <p><a href="http://www.deiwo.de">www.deiwo.de</a> Fertigung in Tschechien, Deutsche Leitung, ISO 9001 zertifiziert, <b>FAIRE PREISE!</b></p> <p>Ansprechpartner: Herr Jornitz Telefon: 00420 475 205 071, Telefax: 00420 475 226 141 eMail: <a href="mailto:info@deiwo.cz">info@deiwo.cz</a>   Anschrift: Krásný Les 341, 40337 Petrovice, Tschechien</p>	<p><b>USB-ISOLATOR</b></p>  <p>Galvanische Trennung Störfreie Messungen Sicherheit und Schutz</p> <p><b>CESYS</b> <a href="http://www.cesys.com">www.cesys.com</a> Elektronik · Messtechnik · FPGA</p> <p><b>W2micro</b> <a href="http://www.w2micro.de">www.w2micro.de</a></p> <ul style="list-style-type: none"> <li>• diverse SMD-Adapter</li> <li>• RS232-TTL-Pegelwandler</li> <li>• Sonderelektronik (z.B. Audioversteiler)</li> <li>• Leiterplattenbestückung und Kabelkonfektionierung ab 1 Stück</li> </ul>	<p>Kleinrechner mit FPGA</p> <p><a href="http://www.bomerenzprojekt.de">www.bomerenzprojekt.de</a></p> <p><b>Ausgabe:</b> Dezember 2012</p> <p><b>Anzeigenschluss:</b> 30. Oktober 2012</p> <p><b>Erscheinungstermin:</b> 28. November 2012</p>

**www.elektor.de**

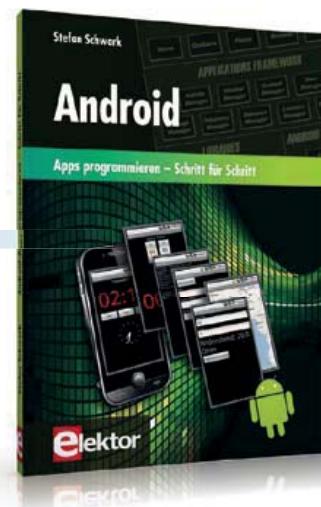
 <p><b>PCBs</b> Muuuuch Cheaper...</p> <p>No-frills policy</p>	<p><b>17.22 EURO*</b></p> <p>5 pcbs, 100 mm x 100 mm *per piece, incl. VAT (23%) + shipping costs e. g. Germany 10.89 EURO</p>	 <p><a href="http://www.jackaltac.com">www.jackaltac.com</a></p>
--	--	---



1



2



3

## Einstieg in die Praxis

### 1 LabVIEW 1

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

240 Seiten (kart.) • ISBN 978-3-89576-253-6

€ 34,80 • CHF 43,20

übersichtlich gestaltete Benutzeroberfläche. Mit der Elektor-DVD 2011 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 978-90-5381-276-1 • € 27,50 • CHF 34,10

### 2 Apps programmieren – Schritt für Schritt

#### 3 Android

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

Funktionsweise des Android-Systems und Schritt für Schritt die Programmierung von Applikationen.

256 Seiten (kart.) • ISBN 978-3-89576-252-9

€ 34,80 • CHF 43,20

## Bestücke und getestete Platine

### 4 Elektor-Linux-Board

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

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

## Kompletter Elektor-Jahrgang 2011 auf DVD

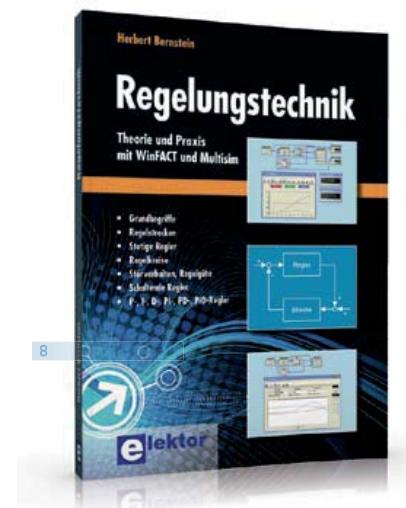
### 5 Elektor-DVD 2011

Die neue Elektor-Jahrgangs-DVD enthält alle Artikel des Jahrgangs 2011. Sie verfügt über eine sehr

## Band 1: 35 Einsteiger-Projekte in C

### 5 ARM-Mikrocontroller

Die Projekte in diesem Buch sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Das heißt nicht, dass diese Projekte einfach sind. Sie sind aber einfach zu verstehen. Es wird beispielsweise die USB-Verbindung zur Kommunikation benutzt, eine



Methode, die im mbed-Board so einfach integriert ist, dass sie sich auch für ein Einsteiger-Buch eignet. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung. Es bedeutet, dass man keinerlei Software auf seinem PC installieren muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC.

261 Seiten (kart.) • ISBN 978-3-89576-262-8

€ 39,80 • CHF 49,40

Von 0 und 1 zum FPGA

## 6 Digitale Logik selbst entwickeln

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

603 Seiten (kart.) • ISBN 978-3-89576-254-3

€ 49,00 • CHF 60,80

Bausatz mit allen Bauteilen & Platine(n)

## 7 TAPIR – E-Smog-Detektor

Der TAPIR (Totally Archaic but Practical Interceptor of Radiation) spürt "strahlendes Missverhalten" elektronischer Geräte in Ihrer Umgebung auf. Trotz der einfachen Schaltung handelt es sich hierbei um einen ultrasensitiven E-Smog-Detektor, der jede Quelle eines elektrischen oder – mit einer entsprechenden Antenne versehen – magnetischen Feldes aufspürt und dies akustisch signalisiert.

Art.-Nr. 120354-71 • € 14,95 • CHF 18,60

Theorie und Praxis mit WinFACT und Multisim

## 8 Regelungstechnik

Die heutige Regelungstechnik hat Verknüpfungspunkte mit fast jedem technischen Gebiet. Ihre Anwendungen reichen von der Elektrotechnik über die

Antriebstechnik und den Maschinenbau bis hin zur Verfahrenstechnik. Will man nun die Regelungstechnik anhand der fachlichen Regeln dieser einzelnen Gebiete erklären, so müsste man von einem Regelungstechniker verlangen, jedes Fachgebiet, in dem er Regelungen vornehmen will, fundiert zu beherrschen. Dies ist aber bei dem heutigen Stand der Technik nicht möglich. Bei der Regelung einer Antriebsaufgabe, einer Druck- oder einer Temperaturregelung tauchen Gemeinsamkeiten auf, die man mit einer einheitlichen Vorgehensweise beschreiben kann. Die Grundgesetze der Regelungstechnik gelten in gleicher Weise für alle Regelkreise, ganz unabhängig davon, wie verschieden sie im Einzelnen auch apparativ aufgebaut sein mögen. Dieses Buch richtet sich an den Praktiker, der gründlicher in die Regelungstechnik eindringen möchte, auf ausschweifende theoretische Exkursionen in die Mathematik aber gerne verzichten kann.

365 Seiten (kart.) • ISBN 978-3-89576-240-6

€ 49,00 • CHF 60,80

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

**www.elektor.de**

Elektor-Verlag GmbH

Süsterfeldstr. 25

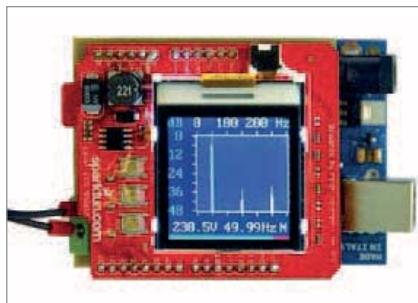
52072 Aachen

Tel. +49 (0)241 88 909-0

Fax +49 (0)241 88 909-77

E-Mail: bestellung@elektor.de

**elektor**



### Netzspannungsanalyser mit Arduino

Ein Arduino-Board ist universeller und leistungsstärker als vielen Anwendern bewusst ist. Dieses Projekt demonstriert, welche Möglichkeiten in einem 8-bit-Mikrocontroller stecken können, aber auch, wo seine Grenzen liegen: Wir bauen einen Spektrumanalysator für die vom Energieversorger gelieferte Netzspannung. Mit der FFT (Fast Fourier Transformation) werden die Harmonischen der Netzfrequenz analysiert und auf einem Display dargestellt. Außerdem erklären wir, wie der Mikrocontroller und die Peripherie möglichst effizient genutzt werden können und welche kostenlosen Tools zum Entwickeln verwandter Projekte verfügbar sind.



### USB-I/O-Schnittstellenkabel

In der Welt der PCs hat der USB-Port die klassische Parallelschnittstelle und den seriellen RS232-Port weitgehend verdrängt. Da wäre es schön, dieses Interface unkompliziert zum Messen und Schalten einsetzen zu können. Hier kommt die Lösung: Ein USB-TTL-Kabel von FTDI wird durch eine Mini-Platine ergänzt, auf der ein R8C-Mikrocontroller seinen Platz hat. Der Mikrocontroller stellt 24 I/O-Leitungen bereit, die der Anwender für seine Zwecke konfigurieren kann. Die Platine ist klein genug, um in das Gehäuse eines 25-poligen D-Sub-Steckers zu passen.



### Multifunktionale Super-Weckuhr

Die meisten Wecker, die heute in den Schlafzimmern stehen, lassen trotz allen technischen Fortschritts immer noch Wünsche offen. Die Zeit war reif, eine Super-Weckuhr zu entwerfen, der es an (fast) nichts mangelt. Auch wenn das eine oder andere „Gadget“ möglicherweise entbehrlich scheinen mag, sicher ist, dass sich unsere Super-Weckuhr von der Massenware aus dem Elektronikmarkt um Längen abhebt. Ein USB-Mikrocontroller AT89C5131 von Atmel macht's möglich, er verhilft dem Wecker sogar dazu, mit einem PC zu kommunizieren!

*Elektor Dezember 2012 erscheint am 28. November 2012.*

*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!*

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

## Elektor im Netz

Alle Magazin-Artikel ab 1996 können einzeln im PDF-Format heruntergeladen werden (gegen Elektor-Credits). Zu jedem Artikel findet man eine kurze Zusammenfassung, technische Daten und eine Stückliste (sofern vorhanden), was bei der Projektauswahl hilft. Dazu kommen eventuelle Updates/Berichtigungen und die Downloads zum Artikel (z.B. Software und Zusatzinfos). Auch die zum Projekt gehörenden Produkte wie Platinen, programmierte Controller und mehr lassen sich hier bestellen.

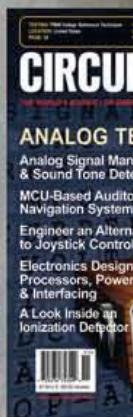
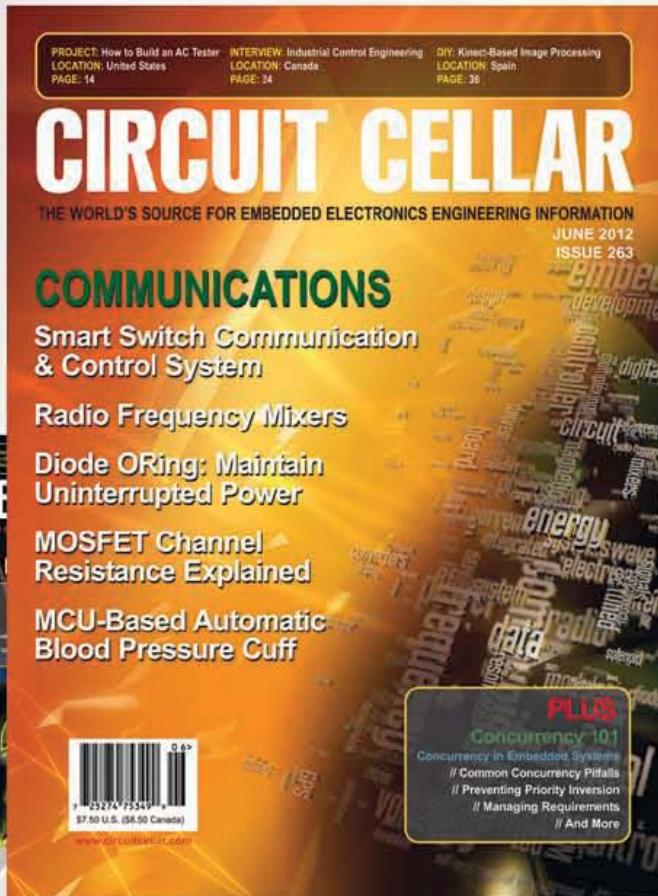
Im Elektor-Shop findet man alle anderen Angebote von Elektor, wie CD-ROMs, DVDs, Kits, Module, Software und Bücher. Mit der Suchfunktion können Sie die ganze Site nach Stichwörtern durchforsten.

### Ebenfalls unter [www.elektor.de](http://www.elektor.de):

- News aus der Elektronikwelt
- Leserforum
- Download des E-Magazins
- Spezialangebote (zeitlich begrenzt)
- FAQs, Inhaltsverzeichnisse und Kontaktformular

**SPECIAL: SAVE 50%    SPECIAL: SAVE 50%    SPECIAL: SAVE 50%**

# Celebrate Circuit Cellar's 25<sup>th</sup> Anniversary



**\$25 Print or Digital :: \$50 Combo**

Celebrate *Circuit Cellar's* 25<sup>th</sup> year of bringing readers insightful analysis of embedded electronics technology.

Visit [www.circuitcellar.com/el912](http://www.circuitcellar.com/el912) to take advantage of these great deals.

**BONUS OFFER! BONUS OFFER! BONUS OFFER! BONUS OFFER!**

Sign up today and you'll also receive the **Special 25<sup>th</sup> Anniversary Edition** with your subscription!



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



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

Bauelemente  
Werkstattbedarf  
Netzwerk- & PC-Technik

## Messtechnik

Sat- und TV-Technik  
Stromversorgung

### DUSPOL expert

Spannungs- und Durchgangsprüfer mit Drehfeldanzeige und Vibrationsalarm

- Spannung von 12V - 690V AC
- Vibrationsalarm zur sicheren Spannungserkennung
- Akustische, optische Durchgangsprüfung
- Drehfeld-/ Phasen-, Polaritätsprüfung
- 30 mA Fl-Auslösung über Drucktaster
- Schutzart IP 64
- zugelassen nach IEC/EN 61243-3 (DIN VDE 0682-401)
- Überspannungskategorie CAT IV 500 V



Bestellnummer:  
DUSPOL EXPERT

**54,45**

Weitere Produkte von BENNING finden Sie in unserem Online-Shop:

Einfach QR-Code per Smartphone scannen oder Kurzlink eingeben:

 <http://rch.lt/6f> 

### Gerätetester BENNING ST 750

Gerätetester zur Prüfung elektrischer und medizinisch elektrischer Geräte

- Für alle VDE 0701-702 und VDE 0751 Prüfungen
- Automatische und selbstkonfigurierbare Prüfbläufe
- Prüflings-/Kundendatenbank auf SD-Karte speicherbar
- Messergebnis mit gut/schlecht Anzeige
- Schnittstellen: 3x USB, 1x RS 232
- Schutzleiterwiderstand mit 200 mA DC und 10 A AC Prüfstrom
- Isolationswiderstand mit 50 V - 500 V Prüfspannung (einstellbar)
- Schutzleiter-/Berührungsstrom über Differenzstrom-, Ersatzableitstromverfahren und direkte Messung
- Funktionstest
- Prüfung von Geräteanschluss- und Verlängerungsleitungen
- Geräteableitstrom, Ableitstrom



Bestellnummer:  
BENNING ST 750

**1.335,00**

## BENNING



### BENNING PV 1 Photovoltaiktester

VDE 0126 (DIN EN 62446)

Photovoltaiktester für die Installations- und Wiederholungsprüfung von netzgekoppelten Photovoltaik-Systemen

- 200 Messwertspeicher für Strang-Strang-Vergleich
- Einfache und sichere Bedienung über Tastendruck
- Verständliche und eindeutige Anzeige aller Messergebnisse
- Schutzleiterwiderstandsmessung mit 200 mA Prüfstrom
- Leerlaufspannungsmessung bis 1.000 V DC
- Kurzschlussstrommessung bis 10 A DC
- Isolationswiderstandsmessung
- Funktionstest

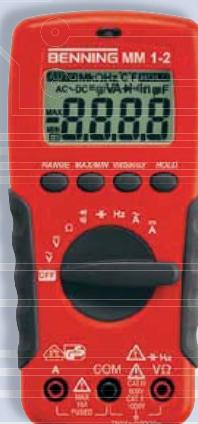
Bestellnummer:  
BENNING PV1

**1.119,00**

### Digital Handmultimeter

mit Voltsensor zur berührungslosen Erkennung der Phasenspannung

- 2.000 Counts
- Grundgenauigkeit: 0,5%
- Alle Grundmessarten
- Messkategorie: CAT III 600 V
- Automatische und manuelle Bereichswahl
- Durchgangsprüfung und Diodentest
- VDE-zugelassen



Bestellnummer:  
BENNING MM 1-2

**79,95**