

Neu: 8 Seiten mehr!

magazine

November 2014 | Nr. 527

elektor

Konverter USB ↔ 4 x UART

RS232 und RS422/RS485
plus Hub für 3 USB-Ports

Ohne Mikrocontroller
Ohne Software

40-V-Labornetzgerät mit zweistufiger Regelung | **C-Software für Elektor Extension-Shield**

Rauchmelder als Universaldetektor | Recycling für Klopfensoren

Arduino mit Atmel Studio | **RPi Compute Module** | Mikrocontroller für Einsteiger: **I²C-Bus**

Bauteil-Tipp: Strommonitor-ICs | DesignSpark Tipps & Tricks

Retronik: Funktelefon von 1965



4 198630 307906

ARM® Software-Entwicklungswerkzeuge

Die Software-Entwicklungsumgebungen von ARM enthalten alle Werkzeuge, die Sie für die Arbeit mit ARM-basierten Systemen benötigen, vom Microcontroller bis zum Applikationsprozessor.

Keil® MDK-ARM™ Version 5

Keil MDK ist die umfassende Entwicklungsumgebung für Microcontroller basierend auf ARM Cortex-M-Prozessoren. Keil MDK enthält die renommierte µVision-Umgebung mit Debugger, die ARM Compilation Tools und den neuen Pack Installer, mit dem Softwarekomponenten wie Chipunterstützung, CMSIS und Middleware installiert und verwaltet werden können.

Keil ULINK Debugadapter

Die Keil ULINK USB-Debugadapter stellen schnell und problemlos die Verbindung zum Zielsystem her. Das Spitzenmodell ULINKpro bietet Streaming Trace-Technologie für umfassende Ausführungsanalyse und vollständige Testabdeckung.

ARM DS-5 Development Studio

DS-5 ist eine vielseitige Lösung für die Arbeit an Embedded-Applikationen, erstellt von ARM für die Entwicklung robuster, hochoptimierter Produkte auf ARM-basierten Prozessoren. Egal, ob Sie ein SoC-Designer, ein Firmware-Ingenieur oder ein Android App-Entwickler sind: DS-5 hilft Ihnen dabei, alle Vorteile der ARM-Architektur voll zu nutzen.

Besuchen Sie unseren
Vortrag auf der



electronica 2014
inside tomorrow

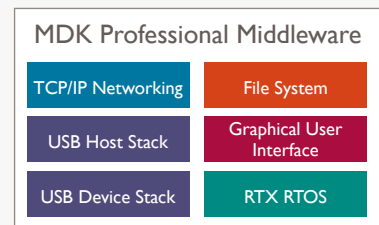
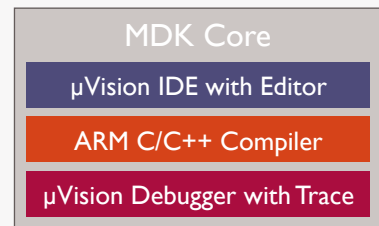
Make the most out of ARM microcontrollers with the
Cortex Microcontroller Software Interface Standard v4

ARM® KEIL®
Microcontroller Tools

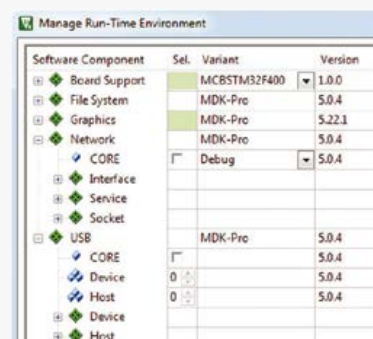
MDK-ARM® Version 5

Das komplette Software-Entwicklungssystem für alle ARM®-basierten Mikrocontroller.

MDK unterstützt alle Cortex-M Prozessoren und führt Softwareentwicklung mit Komponenten ein.



MDK Professional enthält Middleware und standardisierte Treiber. Diese Software-Komponenten werden im Run-Time Environment Window ausgewählt.



keil.com/mdk5

+49 89 456040-20



IFIXIT EUROPE



iFixit Pro Tech Toolkit



Precision Tweezers Set



Magnetic Project Mat

15% Rabatt
mit Coupon Code
ELKTR1114

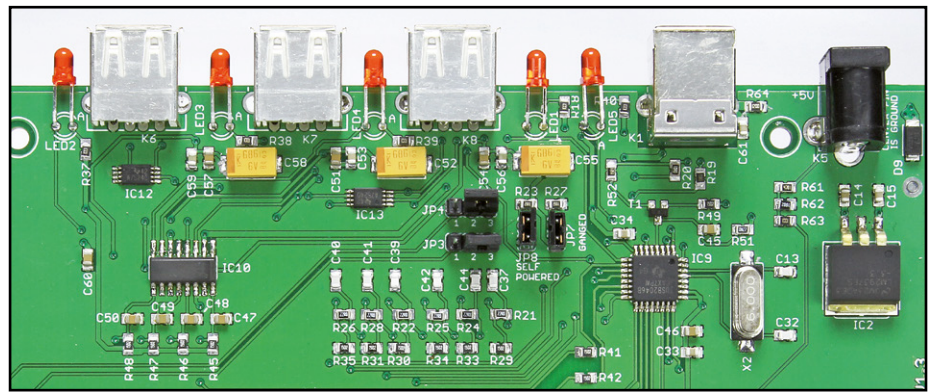
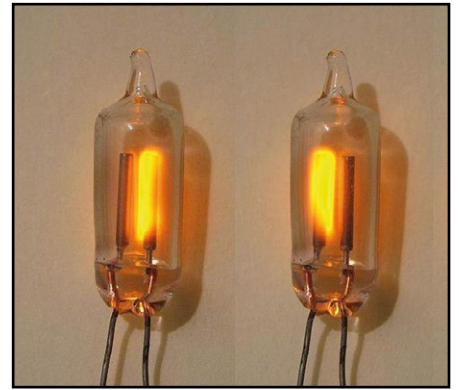
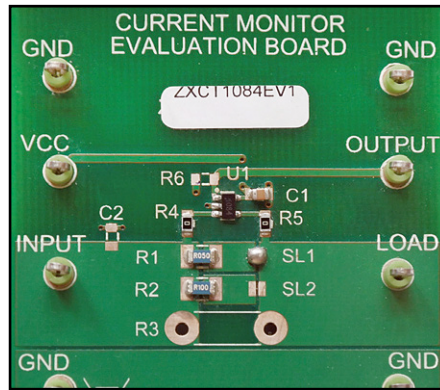
eustore.ifixit.com

iFixit. Das kostenlose Reparaturhandbuch für alles, geschrieben von allen.

Unsere Mission bei iFixit ist das Reparieren. Wir machen die Welt besser, ein Gerät nach dem anderen. Wiederverwenden, Reparieren und Aufarbeiten ist effektiver als Recyceln. Nur so verlängern wir die Lebensspanne von Produkten und schonen wichtige Ressourcen. iFixit ist eine weltweite Community von Tüftlern, die sich gegenseitig dabei hilft, Dinge zu reparieren. Wir teilen Reparaturanleitungen und Know-how online miteinander.

Erfahre mehr unter www.ifixit.com





● Labs

42 Make It in München

44 DesignSpark Tipps & Tricks

Neunummerierung von Bauteilen

46 Bauteil-Tipp: Strommonitor-ICs

Shunt-Widerstände sollen klein sein und der resultierende Spannungsabfall am besten massebezogen, wenn man Mikrocontroller zur Messung von Strömen einsetzt. Für diese Zwecke gibt es kleine Spezial-ICs, die dank integrierter Verstärkung beiden Anforderungen gerecht werden.

48 Seltsame Bauteile (9)

Glimmlampen

50 Neues von Elektor.Labs

52 Von Hinkelsteinen und Hügellgräbern

● Community

6 Impressum

8 Elektor Tomorrow

Elektor auf der electronica 2014

9 3D-Steuerung für µC und PC

Mit 3D-Touchpad und dem Microchip/Elektor-Entwicklungs-Kit

10 WIZnet

Connect the Magic Wettbewerb
Die Gewinner

● Projects

12 VariLab 402 Labornetzgerät

Das Labornetzgerät zählt zum wichtigsten Inventar jedes Elektronik-Labors. Es sollte weder der Ausgangsspannung Störsignale überla-

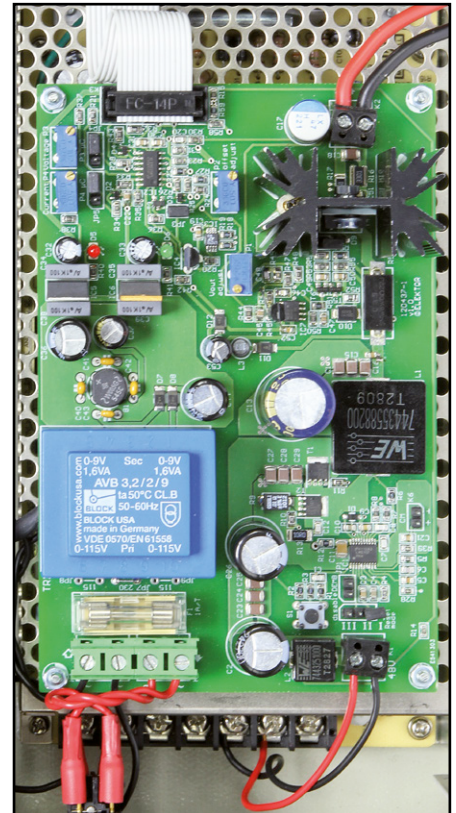
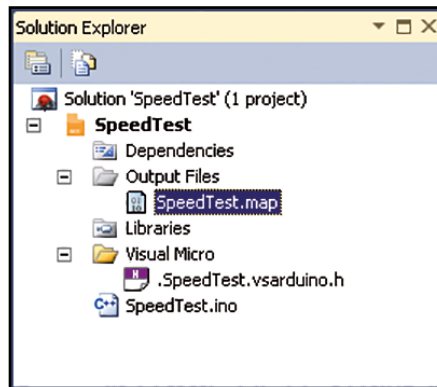
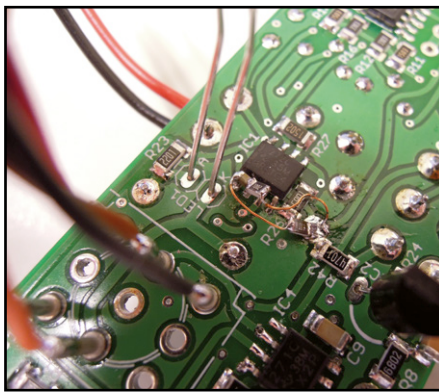
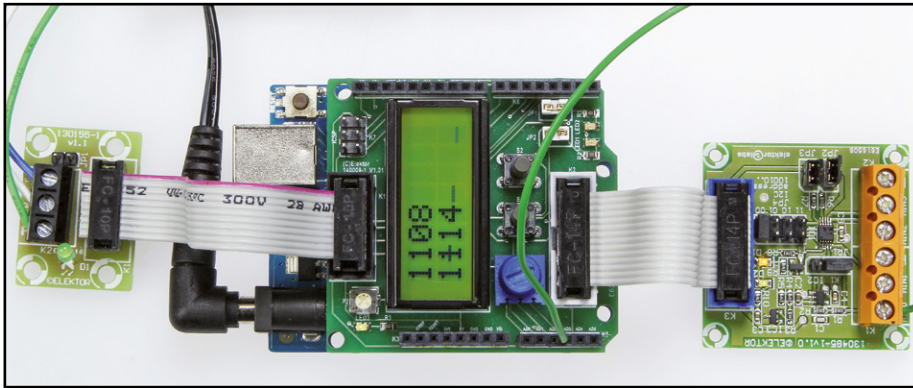
gern noch zu Instabilitäten bei Lastwechseln neigen. Das VariLab 402 aus dem Elektor-Labor arbeitet mit einer zweistufigen, schaltenden und analogen Regelung. Die Spannung ist im Bereich 0...40 V einstellbar.

22 USB-Hub mit RS232/RS485

Hier kommt ein ganz besonderer USB-Hub: Drei USB-Ports werden von zwei voll duplex arbeitenden RS232-Schnittstellen und zwei RS422/485-Interfaces unterstützt. Denn viele bewährte periphere Geräte, die durchaus noch auf dem Stand der Zeit sind, nutzen immer noch die serielle RS232-Schnittstelle zur Kommunikation.

34 C-Module

Mit modularer Hardware kommt man schnell zu einem passenden Prototypen, weil man fertig entwickelte Elemente nur noch zusammenfügen muss. Die Programmiersprache C bietet sich dafür an, dieses Prinzip auf Software



zu übertragen. Für unser Elektor Extension-Shield und drei Erweiterungsboards stellen wir passende Softwaremodule und Demo-Anwendungen vor.

54 Mikrocontroller für Einsteiger (7)

Der I2C-Bus ist eine 2-Draht-Datenverbindung und wurde ursprünglich von Philips für die Unterhaltungselektronik entwickelt. Viele ICs können am selben Bus angeschlossen sein, ohne sich gegenseitig zu stören. In der letzten Folge unseres Mikrocontrollerkurses zeigen wir, wie man Porterweite-

rungen und andere I2C-Chips mit BASCOM ansteuert.

62 Recycling für Klopfensoren

65 Allesmelder

Mit nur einigen externen Komponenten wird ein preiswerter Rauchmelder zu einem universellen Allesmelder. Es können praktisch unbegrenzt Schaltungen und Sensoren gleichzeitig angeschlossen werden. Der Rauchmelder bleibt dabei voll funktionsfähig.

68 Rpi Compute Module = Super-RPi

Der Raspberry Pi bietet allen Programmier-Enthusiasten einen Zugang zu einer preisgünstigen und leistungsfähigen Embedded-Computing-Plattform. Die Raspberry Pi Foundation erweitert die Plattform nun. Das erste Ergebnis ist das Raspberry Pi Compute Module und ein (optionales) Host-Board, mit dem man leicht auf alle I/Os zugreifen kann.

72 Arduino mit Atmel Studio

Wer bisher Mikrocontroller der AT(x)mega-Reihe mit Atmel Studio programmiert hat und die Vorteile eines Debuggers kennt, wird diesen Komfort beim Umstieg auf den ARM-Cortex-Controller des Arduino Due und die Arduino-IDE schmerzlich vermissen. Diesem Manko wird nun abgeholfen.

Magazine

80 Retronik XXL

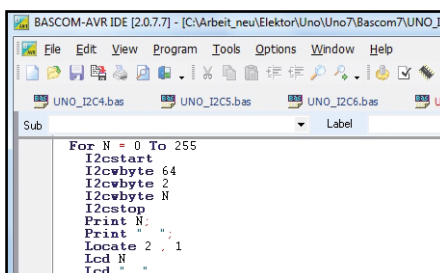
Telefunken-Funktelefon von 1965

87 Hexadoku

Sudoku für Elektroniker

90 Vorschau

Nächsten Monat in Elektor



Impressum

45. Jahrgang, Nr. 527 November 2014

Erscheinungsweise: 10 x jährlich

(inkl. Doppelhefte Januar/Februar und Juli/August)

Verlag

Elektor-Verlag GmbH

Süsterfeldstraße 25

52072 Aachen

Tel. 02 41/88 909-0

Fax 02 41/88 909-77

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

Hauptsitz des Verlags

Elektor International Media

Allee 1, NL-6141 AV Limbricht

Anzeigen:

Margriet Debeij (verantwortlich)

Tel. 02 41/88 909-13 / Fax 02 41/88 909-77

Mobil: +31 6 510 530 39

E-Mail: m.debeij@elektor.de

Julia Grotenrath

Tel. 02 41/88 909-16 / Fax 02 41/88 909-77

E-Mail: j.grotenrath@elektor.de

Es gilt die Anzeigenpreisliste Nr. 44 ab 01.01.2014

Distribution:

IPS Pressevertrieb GmbH

Postfach 12 11, 53334 Meckenheim

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

E-Mail: elektor@ips-pressevertrieb.de

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

© 2014 elektor international media b.v.

Druck: Senefelder Misset, Doetinchem (NL)

ISSN 0932-5468

Mit Partnern für Sie

In dieser Ausgabe finden Sie wieder einen Mix aus Schaltungsprojekten und interessanten Hintergrundartikeln. Natürlich gehören auch Anzeigen zu unserem Magazin. Wie bei anderen Zeitschriften auch sorgen die Einnahmen dafür, dass wir Ihnen unsere Hefte zu einem für jedermann bezahlbaren Preis anbieten können. Im Gegensatz zu fast allen anderen Elektronik-Zeitschriften sind bei uns nicht nur Redakteure, sondern auch Entwickler im Labor mit der Erstellung des Inhalts beschäftigt.

Sie testen Schaltungen, routen Platinen und machen viele Projekte erst nachbausicher.

Leider sind immer weniger Unternehmen dazu bereit, klassische Anzeigen zu schalten.

Ein neuer Trend ist „Native Content“, bei der Redakteure für Unternehmen Auftrags-Artikel schreiben. Ein zweiter Trend ist das Sponsoring von Verlags-Events und interessanten Produkten für die Leser (Sampling).

Damit unsere Zeitschrift eine Zukunft hat, können auch wir uns diesem Umbruch im Verlagswesen nicht ganz verschließen. Doch haben wir die Zusammenarbeit mit Partnerfirmen an zwei Bedingungen geknüpft.

Erstens müssen Sie – liebe Leser – durch ein Sampling oder ein Sponsoring von Veranstaltungen einen echten Nutzen ziehen. Ein Beispiel ist der große, gut ausgerüstete Elektor-Messestand auf der kommenden electronica, der ohne Hilfe von Partnern nicht zustande gekommen wäre. Dort warten Messgeräte und weitere Tools auf Sie, die Sie anhand von eigenen Schaltungen ausprobieren können. Auch die angebotenen Workshops sind kostenlos.

Zweitens werden wir Sie natürlich auch in Zukunft mit redaktionell unabhängigen Reviews und Tests versorgen. Sie müssen deshalb als Leser klar erkennen können, welche Artikel sich auf Kooperationen mit externen Unternehmen beziehen. Alle Artikel, die es ohne ein Sponsoring nicht gegeben hätte, sind deshalb mit dem Stichwort „Advertorial“ in der Dachzeile gekennzeichnet.

Auf diese Weise werden wir Ihnen auch in Zukunft redaktionell unabhängige Projekte und Hintergrund-Artikel anbieten können. Bleiben Sie dran!

Jens Nickel

Chefredakteur Elektor



Unser Team

Chefredakteur:

Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)

Ständige Mitarbeiter:

Dr. Thomas Scherer, Rolf Gerstendorf

Leserservice:

Ralf Schmiedel

Korrekturen:

Malte Fischer

Internationale Redaktion:

Harry Baggen, Jan Buiting, Denis Meyer

Elektor-Labor:

Thijs Beckers, Ton Giesberts, Wisse Hettinga,
Luc Lemmens, Mart Schroyen, Jan Visser,
Clemens Valens, Patrick Wielders

Grafik & Layout:

Giel Dols



Germany

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



United Kingdom

Carlo van Nistelrooy
+44 20 7692 8344
c.vannistelrooy@elektor.com



Netherlands

Ferdinand te Walvaart
+31 46 43 89 444
f.tewalvaart@elektor.nl



France

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



USA

Carlo van Nistelrooy
+1 860-289-0800
c.vannistelrooy@elektor.com



Spain

Jaime González-Arintero
+34 6 16 99 74 86
j.glez.arintero@elektor.es



Italy

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



Sweden

Carlo van Nistelrooy
+31 46 43 89 418
c.vannistelrooy@elektor.com



Brazil

João Martins
+31 46 4389444
j.martins@elektor.com



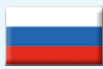
Portugal

João Martins
+31 46 4389444
j.martins@elektor.com



India

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



Russia

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



Turkey

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



South Africa

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



China

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

Unser Netzwerk



VOICE COIL



audioXPRESS



Die Elektor-Community



Unsere Partner und Sponsoren



ARM

www.keil.com/mdk5 2



Batronix

www.batronix.com/go/41 83



Beta Layout

www.beta-estore.com 33



CadSoft

www.cadsoft.de 64



Circuit Design

www.circuitdesign.de 67



Conrad

www.electronica.conrad.de 21



Eurocircuits/PCBservice

www.elektorpcbsevice.com 53



HAMEG

www.hameg.com 92



iFixit

www.ifixit.com 3



LeitOn

www.leiton.de 64



Microchip

www.microchip.com/gestic 91



Pico

www.picotech.com/PS364 77



Reichelt

www.reichelt.de 49, 51



Schaeffer AG

www.schaeffer-ag.de 64

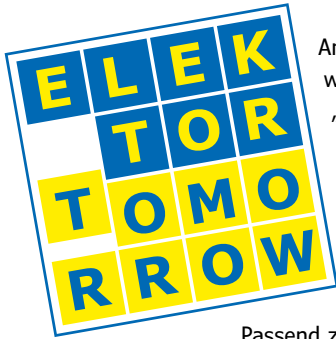


TBP Electronics

www.tbp.eu 61

Sie möchten Partner werden?

Kontaktieren Sie uns bitte unter m.debeij@elektor.de (Tel. 02 41/88 909-13).



Am 11. November ist es so weit: Die weltgrößte Elektronikmesse – die „electronica“ – öffnet in München wieder für vier Tage ihre Pforten. Natürlich wird auch Elektor vor Ort sein und für dieses Jahr haben wir uns etwas ganz Besonderes ausgedacht. In Halle A6, Stand 380, werden wir einen „Maker Space“ einrichten.

Passend zu Elektor gehen wir hier ganz praktisch zu Werke. Besucher sind herzlich eingeladen, eigene Projekte mitzubringen, zu testen, zu messen, zu löten und natürlich mit Entwicklern aus unserem Labor zu fachsimpeln.

Für den „Elektor Maker Space“ haben wir die Unterstützung namhafter Unternehmen aus der Elektronikwelt gewonnen.

- Rohde & Schwarz wird einige Arbeitsplätze mit Oszilloskopen, Multimetern und Netzgeräten ausstatten. Unter anderem kann man in München das HMO3000 Complete-Bundle ausprobieren. Zum Sonderpreis eines voll ausgestatteten Mixed-Signal-Oszilloskops der Hameg HMO3000-Serie packt der Hersteller zusätzlich zwei Logikstastköpfe sowie alle verfügbaren Software-Optionen gratis dazu.
- Der bekannte Elektronikversender Conrad stellt unter anderem Lötstationen zur Verfügung. Voraussichtlich kann man am Elektor-Stand bereits mit der brandneuen digitalen 100-W-Lötstation arbeiten. Auch analoge Lötstationen, eine Heißluft-SMD-Lötstation und weitere Tools hat der Distributor im Gepäck.
- iFixit ist eine weltweite Gemeinschaft von Leuten, die sich gegenseitig helfen, Dinge zu reparieren. Das Unternehmen liefert nicht nur Reparatur-Anleitungen für eine große Anzahl von Tablets und Smartphones, sondern auch spezielle Reparatur-Toolkits. Diese kann man am Elektor-Stand ausgiebig testen.
- Ein weiterer Partner ist der bekannte Toolhersteller Keil/ARM. Das Unternehmen hat unter anderem C/C++-Compiler, Debugger, Echtzeitbetriebssysteme und Eval-Boards im Angebot. Eines der Highlights ist die neue Entwicklungsumgebung Keil MDK Version 5, die sich für eine breite Palette von Cortex-M- und Cortex-R-Mikrocontrollern eignet. Integriert sind ein Debugger, ein ARM C/C++ Compiler und eine große Anzahl von Bibliotheken.

Diese Liste ist nicht vollständig, weitere Messtechnik- und Toolhersteller haben ihr Kommen angekündigt. Am Elektor-Stand kann man die Sachen nicht nur selbst in die Hand nehmen, es sind auch Workshops geplant.

Übrigens: Kaffee, kostenloses WLAN und Lademöglichkeiten für Laptop, Handy und Co. gibt es an unserem Stand natürlich auch. Wir sehen uns in München!

(140385)



3D-Steuerung für μ C und PC

Mit 3D-Touchpad und dem Microchip/Elektor-Entwicklungs-Kit

Microchip Technology Inc. und Elektor International Media stellen gemeinsam ein Entwicklungs-Kit für alle Entwicklungs-Ingenieure und Programmierer vor, die sich für 3D-Sensorik, Touch Control und Gestensteuerung für Mikrocontroller und PCs begeistern.

**MADE IN
MUNICH**

Schlüssel zu dem Projekt ist der **MGC3130**, Microchips erstes IC auf Basis

der GestIC®-Technologie, das kapazitive Veränderungen eines elektrischen Feldes im Femtofarad-Bereich erfassen kann (1 fF = 10^{-15} F).

Das Paket besteht aus einem **MGC3130-Entwicklungs-Kit von Hillstar** und einem **3D-TouchPad**. Das Entwicklungs-Kit umfasst ein MGC3130-Modul, eine I2C-USB-Brücke, eine Vierschicht-Referenzelektrode mit einem 95 x 60 mm² großen empfindlichen Bereich, ein „Hand Brick“-Set zur Selbstmontage (vier Schaumstoffblöcke, eine Kupferfolie) und ein USB-Kabel für den PC-Anschluss. Das Kit ermöglicht die Parametrierung des MGC3130, so dass die damit verbundene GestIC®-Technologie in allen Details untersucht werden kann.

Das **Aurea-Software-Tool** stellt den MGC3130 ein, protokolliert und stellt die Signale in grafisch ansprechender Form dar. Das Kit bietet auch Hilfestellung und Werkzeuge zur speziell auf Ihre Anwendung ausgerichteten Gestaltung der Elektroden.

Das im Paket enthaltene **3D-Touchpad**, ein kompletter 3D-Tracking-Gesture-Controller mit Mausfunktionalität, erweitert out of the box jeden PC, jedes Tablet und jedes Embedded-System mit USB um eine große hochempfindliche Touchpad-

fläche. Auf der internen Platine befinden sich sowohl die Elektroden für die 3D-Erfassung als auch die Touch-Sensor-Matrix für das Touchpad.

3D-Touchpad und MGC3130-Entwicklungs-Kit von Hillstar bieten eine solide Grundlage zur Entwicklung einer 3D-Steuerung für Embedded-Systeme einschließlich Arduino, RPi, BBB, T-Boards und anderen Plattformen. In Elektor werden in naher Zukunft einfache und kompliziertere Anwendungen der GestIC®-Technologie beschrieben. Der TuxRacer ist als Einstieg perfekt.

Das Entwicklungs-Kit mit Touchpad wird ausschließlich durch Elektor zum Sonderpreis verkauft.

Weitere Informationen und Ankündigungen zur Produktverfügbarkeit werden Sie im wöchentlichen Newsletter Elektor.POST, in der Elektor-Zeitschrift und auf der Elektor-Labs-Website finden. Oder Sie fragen am Elektor-Stand auf der Electronica 2014 in München nach!

(140408)





WIZnet Connect the Magic Wettbewerb 2014: Die Gewinner

Der Entwicklungswettbewerb Connect the Magic 2014 von WIZnet gab Elektronik-Begeisterten die Möglichkeit, mit dem Ethernet-Modul WIZ550io ein Projekt zu entwickeln. Dabei hatten sie die Chance, einen Teil des Preisgeldes von 15.000 Dollar einzustreichen. Schon bald nach der Einsendefrist am 3. August 2014 begann die Jury, die Einsendungen auf ihre technische Qualität, Originalität, Nützlichkeit, Wirtschaftlichkeit und Design-Optimierung abzuklopfen. Wir freuen uns, nun die Ergebnisse und Gewinner bekanntgeben zu können.

Sie können die kompletten Projekte (Dokumentation, Schaltpläne, Fotos, Code und mehr) bei <http://circuitcellar.com/wiznet2014/> einsehen. Vielen Dank allen Teilnehmern und herzlichen Glückwunsch den Gewinnern!

Hauptgewinn

Chimäre: Das Poly-Magneto-Phonische Theremin Hans Peter Portner (Schweiz)

Die Chimäre ist ein berührungsloser, netzwerkfähiger, polyphoner Musik-Controller als Open-Source-Hardware. Es ist ein gemischt analog/digitaler Klon des Theremins. Ein Array von analogen, linearen Hall-Effekt-Sensoren bildet eine kontinuierliche, zweidimensionale „Spielfläche“. Die Sensoren werden mit Neodym-Magneten an den Fingern des Künstlers gesteuert. Das Gerät verfolgt und berech-



net Position und Entfernung mehrerer Magneten in der Nähe des Sensorarrays, um aus diesen Informationen Signale zu erzeugen. Die Signale sind als Open Sound Control kodiert und werden über UDP/TCP an einen Software-Synthesizer übertragen. Die DSP-Einheit basiert auf einem ARM Cortex M4-Mikrocontroller, der mit dem WIZnet W5500-Chip verbunden ist, welcher sich wiederum um die Netzwerkkommunikation via UDP/TCP kümmert.

Zweiter Preis

LCDTV-Server: Streaming Media mit Ethernet/ USB-Adapter

Lindsay Meek (Australien)

Das Server-LCD-TV-Projekt mit WIZnets WIZ550io ermöglicht es, dass ein LCD-TV, welches mit USB ausgestattet ist, Bild und Ton über ein LAN streamt. Wenn der Anwender den Server mit einem



Power-Line-zu-Ethernet-Adapter kombiniert, kann er digitale Videos auf einem älteren TV überall in der Nähe sehen.

Dritter Preis

WIZ Security-Network

Claudiu Chiculita (Rumänien)

Das Projekt stellt ein Sicherheitssystem mit mehreren Knoten dar. Die Knoten können unabhängig Informationen sammeln und verarbeiten. Sie können Alarmer generieren sowie mit den anderen Knoten kommunizieren, um Sicherheitsbedrohungen aus verschiedenen Blickwinkeln zu untersuchen. Jeder Knoten besitzt einen W5500-Netzwerkchip von WIZnet, passives Power over Ethernet, einen passiven Infrarotsensor (PIR), einen Servomotor, Speichermöglichkeiten, eine Videokamera und Bildbearbeitungsfunktionen. Zur Überwachung und Konfiguration wird eine PC-Applikation eingesetzt.



Lobende Erwähnung

Sentry

David Penrose (USA)

Das Sentry-Projekt setzt eine Reihe von passiven IR-Sensoren in verschiedenen Räumen eines Seniorenheims ein, um Bewegungen im Haus zu verfolgen. Das System verwendet Bewegungssensoren, die (auch) an Stühlen/Betten angebracht sind, um zu ermitteln, ob der Bewohner den Stuhl/das Bett belegt. Diese Informationen werden über eine Funkverbindung an einen Prozessor mit dem angehängten WIZ550io geleitet, um die Aktivität der Bewohner unauffällig zu überwachen und zu entscheiden, ob eine normale Aktivität vorhanden ist oder nicht. Sollte die „Intelligenz“ des Systems entscheiden, dass die Bewegungsmuster außerhalb des normalen Verhaltens liegen, schlägt sie Alarm und benachrichtigt ein (entferntes) Familienmitglied oder das Pflegepersonal. Das Gerät gewährt älteren Bürgern die Privatsphäre in ihrer Wohnung und gleichzeitig die Sicherheit, dass jemand zur Hilfe eilt, wenn diese nötig ist.



Lobende Erwähnung

Tierfutter-Automat

Dean Boman (USA)

Der Tierfutter-Automat wurde entwickelt, um Kleintieren wie Ziegen und Schafen ihr täglich' Heu zuzuführen. Das WIZ-550io-Netzwerkmodul von WIZnet stellt die Netzwerkschnittstelle zur Verfügung. Das Design bringt das Internet of Things auf den Bauernhof; die Fütterung und Überwachung kann über das Internet ferngesteuert werden.



Lobende Erwähnung

WIZpix

Connected Pixel-Controller (CPC)

Robert Gasiorowski (USA)

Die WIZpix-Pixel-Controller verwenden einen WIZnet-W5500 zur Kontaktaufnahme mit dem Internet und einen Mikrocontroller zur Steuerung „intelligenter“ Pixel. Anlass für das Projekt war eine intelligente



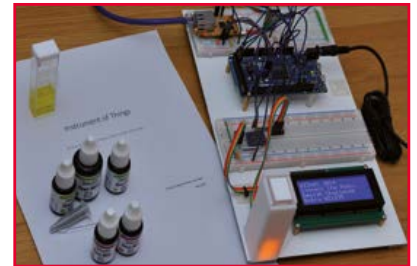
Weihnachtsbeleuchtung. CPC kann allerdings überall dort verwendet werden, wo man eine animierte Beleuchtung benötigt (Partys, Displays, Shows, Wohnungseinrichtung und so weiter). Ein teures und komplexes DMX-System ist nicht nötig. Dank der integrierten PoE-Funktion ist nur ein einziges Kabel erforderlich.

Lobende Erwähnung

Das Instrument of Things

Radko Bankras (Niederlande)

Das Instrument of Things (IoT) ermöglicht es, elektrische Geräte um eine industrielle Standard-Funktion für die Fernsteuerung über eine TCP/IP-Schnittstelle zu erweitern. Das WIZ-550io-Modul von WIZnet wird als Basis-Web-Server, für den Portmap-Dienst und als Server für die Fernsteuerung des Gerätes im VXI-11-Kommunikationsformat verwendet. Letztendliches Ziel des Instrument of Things ist es, das VXI-11-Kommunikationsprotokoll und die LAN eXtensions for Instruments-Technologie (LXI) jedem beliebigen elektrischen Gerät einfach hinzuzufügen zu können.



Lobende Erwähnung

Radioteleskop-Controller

Clayton Gumbrell (Australien)

Dieser Controller steuert ein Radioteleskop, das das Universum auf Wasserstoff-Emissionen bei einer Frequenz von 1.420 MHz (21 cm Wellenlänge) untersucht. Das Radioteleskop besteht aus einer 1,7-m-Parabolantenne an einer motorisierten Azimut/Elevation-Nachführung, lenkbar in alle Richtungen über dem Horizont. Ein WIZ550io-Ethernet-Modul von WIZnet verbindet die Antennensteuerung mit dem Internet, so dass der Astronom das Teleskop aus der Ferne einstellen kann.



(140309)

VariLab 402

Labornetzgerät 40 V mit zweistufiger Regelung



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

Das Labornetzgerät zählt zum wichtigsten Inventar jedes Elektronik-Labors. Das Gerät muss seine Aufgabe unter allen labortypischen Betriebsbedingungen zuverlässig erfüllen. Es darf weder der Ausgangsspannung Störsignale überlagern noch zu Instabilitäten bei Lastwechseln neigen. Unser neues Labornetzgerät arbeitet mit zweistufiger, schaltender und analoger Regelung. Die Spannung ist einstellbar im Bereich 0...40 V.

In jedem Elektronik-Labor sind mindestens drei Dinge unverzichtbar: Das Multimeter, das Lötgerät und das Labornetzgerät. An das Labornetzgerät werden hohe Anforderungen gestellt. Ihm werden nicht nur die labortypischen Spannungen und Ströme abverlangt, darüber hinaus hat die Zuverlässigkeit einen exponierten Stellenwert. So genannte Labornetzgeräte gibt es im Handel bereits für einige zehn Euro, doch es ist fraglich, ob sie diesen Namen verdienen. Die Preise qualitativ hochwertiger Produkte bewegen sich im Bereich mehrerer hundert oder sogar tausend Euro. Während der rund 50 Jahre seines Bestehens hat Elektor eine ganze Palette von Labornetzgeräten hervorgebracht. Das geschah nicht zuletzt, weil es hier um ein beliebtes und gefragtes Bauprojekt geht. In den zurückliegenden Monaten waren

Netzgeräte das Thema mehrerer Beiträge. Das hier vorgestellte neue Labornetzgerät ist eine vollständige Eigenentwicklung des Elektor-Labors, es bietet etwas mehr als gewohnt. Die meisten kostengünstigen Standard-Netzgeräte liefern Ausgangsspannungen bis 30 V. Da diese Obergrenze für bestimmte Anwendungen zu niedrig ist, haben wir die Ausgangsspannung des neuen Labornetzgeräts auf 40 V bei Strömen bis 2 A festgelegt. Ein Mikrocontroller übernimmt die Steuerung, er gibt Daten auf dem Display aus und kommuniziert mit einem PC.

Nachdem der Rahmen für unser Vorhaben abgesteckt war, sind wir auf die Suche nach einem geeigneten Konzept sowie nach den für die Realisierung nötigen Bauteilen gegangen. Wegen des weiten Spannungsbereichs 0...40 V wäre eine kon-

ventionelle lineare Regelung in hohem Maß unwirtschaftlich. Beim Ausgangsstrom 2 A müssten bis 80 W in Verlustwärme umgesetzt werden, der Wirkungsgrad wäre nicht akzeptabel. Ein überdimensionaler Kühlkörper oder ein starker Lüfter müssten die immense Verlustwärme abführen. Der Wirkungsgrad lässt sich in Grenzen durch einige Kunstgriffe steigern. Beispielsweise könnte das Netzgerät mit umschaltbaren Trafo-Spannungen arbeiten und mehrere feste Ausgangsspannungen liefern. Doch auch damit geht immer noch viel Energie als Wärme verloren. Aus diesem Grund haben wir uns für das Konzept des geschalteten Netzteils entschieden, kurz SMPS genannt (*Switched-Mode Power Supply*). Der hohe Wirkungsgrad eines SMPS muss allerdings durch Störspannungen auf der Ausgangsspannung erkauft werden, die von den Schaltvorgängen herrühren. Um die Störspannungen auf marginale Werte hinabzudrücken, ist hier dem SMPS eine analoge Regelung nachgeschaltet.

Blockschema

Die Funktionsbausteine unseres Labornetzgeräts „VariLab 402“ sind im Blockschema dargestellt, das **Bild 1** zeigt. Gewisse Parallelen zum „Professionellen Labornetzteil“ aus der Elektor-Ausgabe vom September 2014 sind unverkennbar, was nicht verwundert, da Teile jenes Projekts ebenfalls im Elektor-Labor entstanden sind.

Am Eingang liegt ein Schaltnetzteil aus industrieller Fertigung, das die feste Spannung 48 V und den Strom 3 A liefert. Der Schaltregler setzt diese Spannung abhängig von der Einstellung des Benutzers auf 3...43 V herab. Auf den Schaltregler folgt eine analoge Regelung, die Störspitzen und Restwelligkeiten nahezu vollständig beseitigt. Das Spannungsgefälle am analogen Regler wurde auf 3 V festgelegt, ein guter Kompromiss zwischen effizienter Störunterdrückung und tolerierbarer Verlustleistung (6 W bei 2 A). Letztes Glied der Kette ist ein Strommesser, er gibt den gemessenen Wert an den Mikrocontroller weiter. Der Mikrocontroller ist die Informations- und Steuerzentrale, hier laufen die Informationen über die eingestellten Spannungs- und Stromwerte sowie die tatsächlichen Werte zusammen. Alle eingestellten und gemessenen Werte werden auf einer vierzeiligen alphanumerischen Anzeige ausgegeben. Die Verbindung mit einem PC stellt der Mikrocontroller über den integrierten USB-Port her. Das Labornetzgerät wird über zwei Drehencoder und einen Drucktaster bedient.

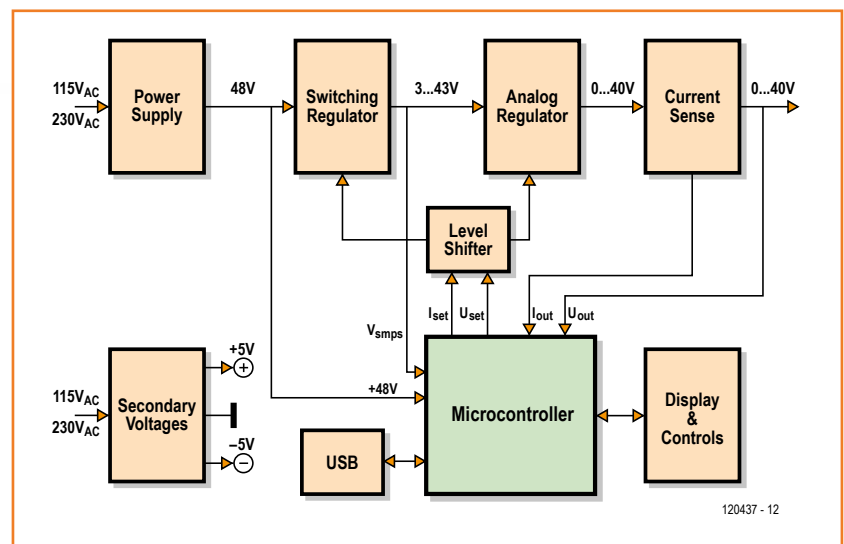
Eigenschaften

- Stabilisierung der Ausgangsspannung durch zweistufige Regelung
- Eingangsspannung von handelsüblichem Schaltnetzteil 48 V/150 W
- Ausgangsspannung 0...40 V, Ausgangsstrom 0...2 A
- Spannung und Strom wahlweise einstellbar über Potentiometer oder Mikrocontroller
- Mikrocontroller-Steuerung mit ATxmega128A4U-AU
- Bedienung mit zwei Dreh-Encodern und Drucktastern
- Vierzeilige Anzeige u. a. mit U_{set} , I_{set} , U_{uit} , I_{uit} , abgegebene Leistung und Crest-Faktor
- USB-Anschluss für Verbindung mit PC
- Ausgang abschaltbar mit Drucktaster
- Restwelligkeit < 15 mVSS bei 40 V/2 A
- Spannungsänderung bei Pulsbelastung 220 mV_{SS} (Last 100 mA/1 A, Frequenz 50 Hz, Duty-Cycle 50 %)
- Leistungsbedarf der Regelungsplatine ohne Last < 4 W
- Wirkungsgrad 92 % bei 40 V/2 A, 61 % bei 5 V/1,75 A (gemessen zwischen Ein- und Ausgang der Regelungsplatine)

Blockdiagramme sind schnell gezeichnet, der sicher schwierigere Schritt ist die Realisierung. Was die Beschreibung betrifft, beschränken wir uns in diesem Beitrag auf die Umsetzung des energietechnischen Teils. Die Steuerung mit dem Mikrocontroller und dem Display wird Thema eines zweiten Beitrags im nächsten Monat sein.

Dem Problem, einen leistungsstarken und ebenso teuren Netztransformator beschaffen zu müssen, sind wir aus dem Weg gegangen, indem wir als Gleichspannungslieferanten ein Schaltnetzteil des Herstellers Mean Well einsetzen. Das Modul leis-

Bild 1. Funktionsschema des „VariLab 402“. Hier arbeiten eine schaltende Regelung und eine analoge Regelung Hand in Hand.



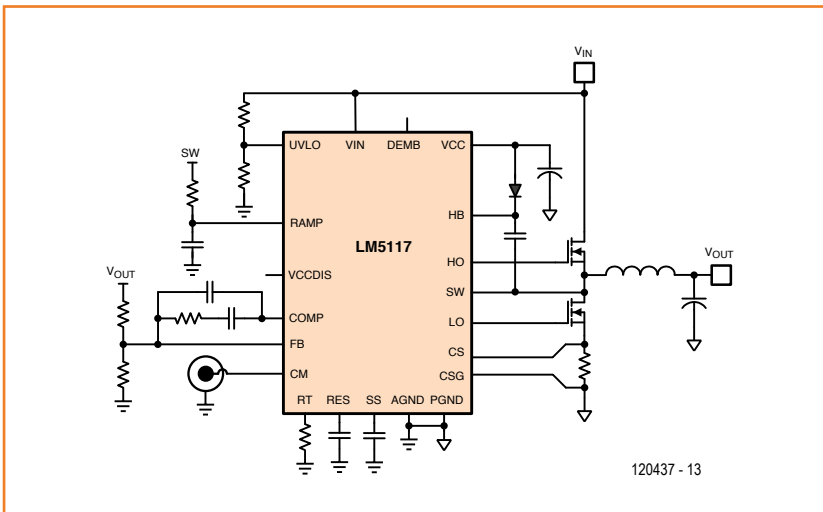


Bild 2.
Grundschriftung des
Abwärtsschaltreglers
LM5117 von Texas
Instruments.

tet 150 W, die Ausgangsspannung beträgt 48 V. Bei der Suche nach einem geeigneten Abwärtsregler-Baustein für die schaltende Stufe (SMPS) ist uns aufgefallen, dass nur wenige am Markt verfügbare Typen der Spannung 48 V gewachsen sind. Nach intensiven Recherchen haben wir uns für den LM5117 von Texas Instruments entschieden. Das Datenblatt [1] und die *Application Note* [2] können von der Hersteller-Website heruntergeladen werden. Die grundlegende Beschaltung dieses Bausteins ist in **Bild 2** wiedergegeben, sie ist in der detaillierten Schaltung leicht wiedererkennbar. Der LM5117 schaltet zwei externe Power-MOSFETs, die ihrerseits mit hoher Schaltfrequenz (etwa 100 kHz) eine LC-Kombination steuern. Der durch die MOSFETs fließende Strom wird mit einem Widerstand gemessen, der in der Source-Leitung eines MOSFETs liegt. Die übrigen Bauelemente dienen entweder zum Einstellen der Ausgangsspannung, oder sie steuern diverse zeitliche Abläufe.

Realisierung

Wie wir das Blockscheema real umgesetzt haben, ist aus **Bild 3** ersichtlich. Der Teil mit dem Mikrocontroller und dem Display ist für die nächste Elektor-Ausgabe reserviert.

Schaltregler

Die Eingangsspannung 48 V liegt an Klemme K1. Der schaltende Abwärtsregler IC1 zerhackt die Gleichspannung, sie gelangt in Form eines pulsformigen Signals über die Power-MOSFETs T1 und T2 zum LC-Kreis L1/C13...C16. Die Pulsbreite hängt von dem rückgekoppelten Signal ab, das an Pin 8 des LM5117 liegt. Das Konzept folgt weitgehend

dem von TI veröffentlichten Vorschlag [2]. Abweichend vom Vorschlag haben wir die niedrigere Schaltfrequenz 100 kHz gewählt, weil sie auf einer doppelseitigen Platine mit weniger Rauschen und niedrigeren Schaltverlusten verbunden ist. Noch wichtiger ist der maximale Duty-Cycle, er beträgt bei 100 kHz rund 96 %. Die Schaltfrequenz hängt von R4 ab, während C4 die Softstart-Zeit beim Einschalten bestimmt. Mit dem relativ kleinen Wert 470 nF liegt diese Zeit bei 38 ms. Spannungsteiler R2/R1 ist für die Mindestspannung zuständig, bei der IC1 seinen Betrieb aufnimmt. Die Schwelle ist auf ungefähr 44 V eingestellt. Die an R9 abfallende Spannung, die zum MOSFET-Strom proportional ist, wird von R7, R8 und C8 geglättet.

Kondensator C7 und Widerstand R6 sind für die Sägezahnspannung der Pulsweitenregelung zuständig. Der Wert 1 nF entspricht der Hälfte des maximalen Werts. Das Datenblatt empfiehlt für R6 den Wert 820 k Ω , doch mit dem Wert 1,2 M Ω ließ sich eine noch etwas höhere Regelungsstabilität bei unterschiedlichen Ausgangsspannungen erreichen. Die Entkoppelkondensatoren C9 (V_{CC}) und C10 (HB) haben die im Datenblatt empfohlenen Werte. Bootstrap-Diode D1 ist ein Typ mit niedriger Durchlassspannung, sie beträgt 0,57 V beim Strom 1 A.

Die in der *Application Note* für die Schalttransistoren T1 und T2 empfohlenen MOSFET-Typen von NXP sind für diese Anwendung wie maßgeschneidert. R11 und R10 beugen parasitären Schwingungen vor, sie verkürzen allerdings auch die Totzeit. Doch da sich die von den Ausgängen HO und LO kommenden Ströme in ihren Werten unterscheiden, abhängig davon, ob T1 und T2 Stromquelle oder Stromsenke sind, werden T1 und T2 schneller ausgeschaltet als eingeschaltet. Das RC-Glied R13/C12 dämpft Spannungsspitzen, die an den MOSFET-Ausgängen infolge der Induktivität L1 auftreten können.

Die Induktivität L1 haben wir nach der Formel aus dem Datenblatt berechnet, wobei die feste Spannung 48 V und als zulässige Restwelligkeit 40 % des maximalen Ausgangsstroms, also 0,8 A vorgegeben waren. Der rechnerische Wert beträgt 83,33 μ H, er liegt nahe dem Wert 82 μ H aus der Normreihe E12. Der verwendete Hochstrom-Typ von Würth Elektronik ist zwar überdimensioniert, dafür sind die Leistungsverluste gering. Durch diese Induktivität darf mehr als der doppelte Ausgangsstrom fließen, ohne dass die Sättigung erreicht wird. Im Prinzip kann der schaltende Teil des Labornetzgeräts wesentlich mehr leis-

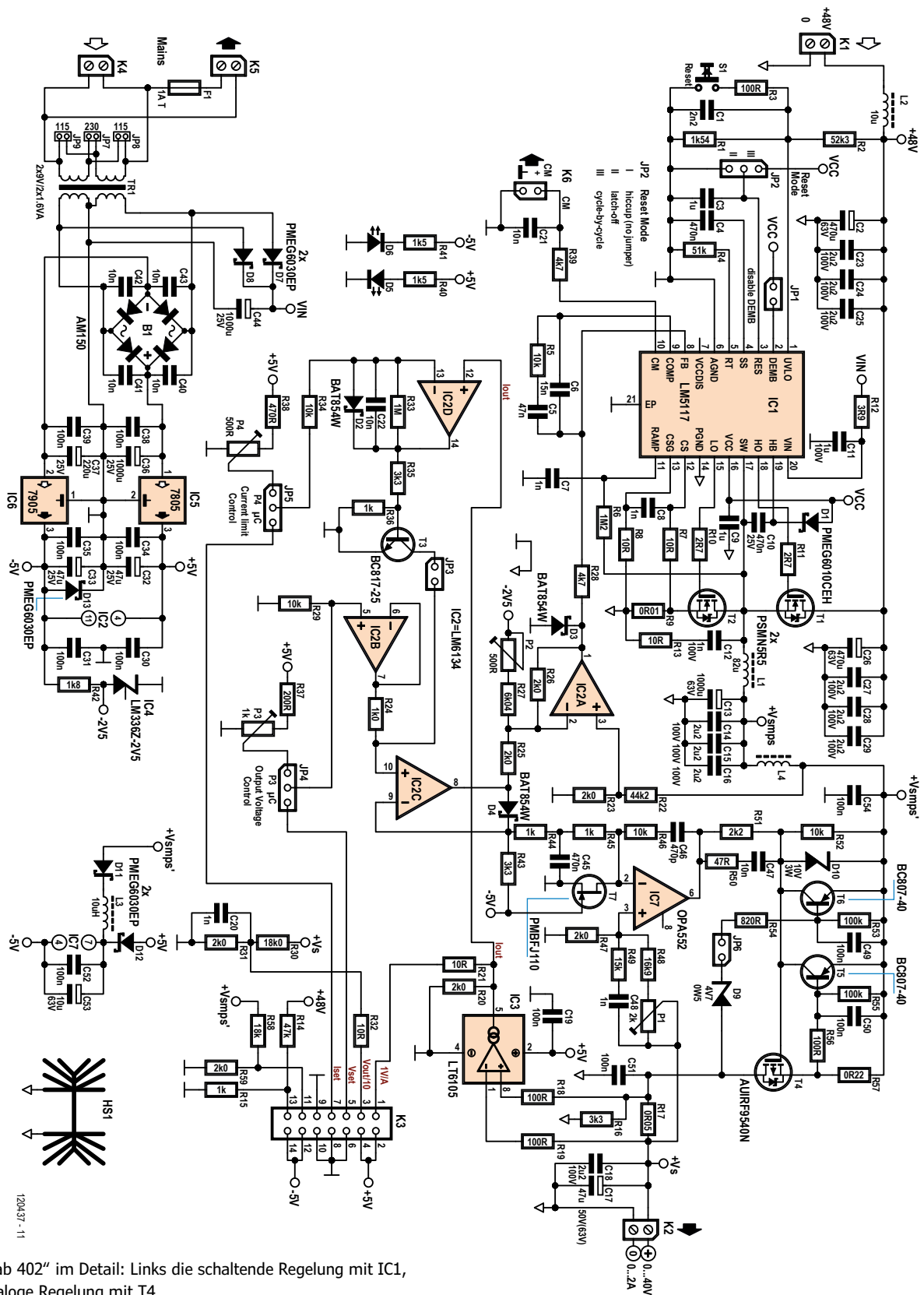


Bild 3. „VariLab 402“ im Detail: Links die schaltende Regelung mit IC1, rechts die analoge Regelung mit T4.

ten als der analoge Teil, was durchaus Absicht ist. Damit wird der Einfluss des schaltenden Teils auf die Gesamtregelung gemindert, insbesondere wenn der Stromregelmodus aktiv ist. Wir haben die Obergrenze auf 12 A festgelegt, was viel zu hoch erscheinen mag, doch Tests mit niedrigeren Werten resultierten in geringerer Stabilität. Kondensator C13 ist ein Typ mit niedrigem ESR, hoher Impulsfestigkeit (2,7 A bei 100 kHz) und langer Lebensdauer (3000 h bei 105 °C). Die hohe Impulsfestigkeit ist notwendig, um den Anforderungen des Schaltbetriebs standzuhalten. Die Kondensatoren C14...C16 liegen parallel, um den ESR zu minimieren. Trotz aller Maßnahmen beträgt die Restwelligkeit bei 20 V Gleichspannung am Ausgang noch etwa 40 mV. Diese Restwelligkeit wird von der nachgeschalteten analogen Regelung nahezu vollständig beseitigt. Der interne Spannungsregler des LM5117 (typische Spannung 7,6 V) liegt an dem separaten, mit Trafo TR1 aufgebauten Netzteil. Das Netzteil liefert auch die Spannung ± 5 V für die Opamps. Da diese Spannung wesentlich niedriger als die Spannung 48 V ist, muss der LM5117 nur wenig Leistung in Wärme umsetzen. Das RC-Glied R12/C11 dämpft hochfrequente Störungen, die beim Schalten entstehen. Das Signal CM des LM5117 (Pin 10) ist ein Maß für den mittleren Ausgangsstrom, doch diese Information gilt nur für kontinuierliche Ströme. Für Testzwecke ist das Signal an Steckverbinder K6 zugänglich.

Der LM5117 bietet die Reset-Modi *Hiccup*, *Latch-off* und *Cycle-by-cycle* an. Normalerweise ist *Hiccup* die zu bevorzugende Einstellung, weitere Informationen können dem Datenblatt entnommen werden. Mit Jumper JP2 lässt sich der Reset-Modus

umschalten. Der Regler startet im umgeschalteten Modus nach Drücken des Reset-Tasters S1 oder nach Ausschalten und Wiedereinschalten. Wenn im Modus *Hiccup* die Strombegrenzung wirksam war, hängt die Softstartzeit von C3 ab.

Analoger Regler

Die analoge Stufe besteht im Wesentlichen aus einem Opamp (IC7), der einen Power-MOSFET (T4) steuert. MOSFETs haben den Vorteil, dass Ströme ohne nachteilige Folgen auch rückwärts fließen dürfen, solange die Höhe der Ströme in Grenzen bleibt. Wir haben einen p-Kanal-MOSFET als Source-Folger geschaltet, um Spannungsänderungen am MOSFET infolge von Stromänderungen zu minimieren. Opamp IC7 steuert MOSFET T4 über den Spannungsteiler R52/R51. Der Spannungsteiler stellt sicher, dass der Opamp den MOSFET vollständig sperrt. Die niedrige Spannungsdifferenz zwischen dem Ausgang und der Betriebsspannung von IC7 (ungefähr 2,7 V) kann im ungünstigen Fall nicht ausreichen ($U_{GS} \geq -2V$). Das RC-Glied R50/C47 parallel zu R51 erhöht die Regelgeschwindigkeit. C47 hat eine Kapazität, die ein Vielfaches der Eingangskapazität des MOSFET beträgt (1300 pF).

Als Opamp haben wir einen speziellen Typ gewählt, den OPA552. Dieser Opamp ist für Betriebsspannungen im Bereich ± 4 V... ± 30 V geeignet, der Ausgangsstrom kann bis 200 mA betragen. Da der Ausgangsspannungsbereich des Reglers bei 0 V beginnen soll und dazu am Gate

des MOSFETs eine negative Spannung liegen muss, arbeitet der Opamp an der negativen Betriebsspannung -5 V.

Die RC-Kette R45/R46/C46 trägt zur Stabilität der Regelung bei, insbesondere im Stromregelmodus.

R49 und C48 im Rückkoppelzweig vom Schaltungsausgang zum nicht invertierenden Eingang des Opamp IC7 stabilisieren ebenfalls das Regelverhalten, insbesondere bei Transient-Lasten. Anders betrachtet ist dieser Eingang der invertierende Eingang der linearen Ausgangsstufe IC7 + T4, denn T4 invertiert das Signal. Das



Eingangssignal des Opamp wird von C45 geglättet, außerdem ist C45 Teil der Regelschleife im Stromregelmodus. C51 dämpft eventuelles hochfrequentes Rauschen.

D11, L3 und D12 sind daran beteiligt, die Betriebsspannung für Opamp IC7 aufzubereiten. Damit wird verhindert, dass beim Einstellen niedriger Ausgangsspannungen nahe 0 V die Betriebsspannung von IC7 so weit sinkt, dass T4 nicht mehr sicher gesteuert werden kann. Normalerweise erhält IC7 die Betriebsspannung von V_{smps} auf der Eingangsseite von T4. Über D12 übernimmt die Spannung +5 V die Funktion der Betriebsspannungsquelle. Beim Wegfall der Betriebsspannung ± 5 V und Ausgangsspannungen unter etwa 3 V kann die Ausgangsspannung von IC7 durch IC2C nicht mehr sicher gesteuert werden. Um dies zu verhindern, wurde FET T7 zum invertierenden Eingang des Opamp IC7 hinzugefügt. Bei Wegfall der Spannung -5 V leitet der FET, so dass T4 sperrt und die Ausgangsspannung sinkt.

Mit Shunt-Widerstand R17 vor dem Ausgang wird der Ausgangsstrom gemessen. Bei der gebräuchlichen Konfiguration liegt der Messwiderstand in der Rückleitung. Die hier angewandte Lösung hat den Vorteil, dass sämtliche Masseverbindungen unmittelbar miteinander verbunden sind. IC3 misst die Spannung an R17, sie ist proportional zum hindurch fließenden Strom. C17 und C18 puffern und glätten die Ausgangsspannung.

Die höhere Ausgangsimpedanz infolge R17 wird kompensiert, indem das Rückkopplungssignal für Opamp IC7 (über R47, R48 und P1) unmittelbar an Ausgangsklemme K2 abgegriffen wird. Damit werden alle Impedanzen zwischen dem MOSFET und K2 kompensiert, einschließlich sämtlicher Lötstellen. Die Ausgangsspannung wird mit P1 eingestellt, oder der Mikrocontroller übernimmt diese Aufgabe.

Strommessverstärker IC3 ist ein *Current-sense*-Opamp des Typs LT6105. Dieser Typ zeichnet sich durch einen weiten Eingangsspannungsbereich bis mindestens 44 V aus, während die Betriebsspannung nur 5 V betragen kann. Die Details dieses Bausteins sind dem Datenblatt [3] zu entnehmen. Die Verstärkung wird durch das Verhältnis R20 zu R18 und R19 eingestellt, sie ist hier auf 20 festgelegt. Daraus resultiert eine Messspannung an R20 von 1 V/A. Die Genauigkeit des Messverstärkers hängt in geringem Maß von der Ausgangsspannung ab. Dieser Effekt ist vernachlässigbar, solange die Anzeige lediglich drei Digits umfasst. Eine Beschreibung auf unse-

rer Website *elektor-labs* [4] geht darauf ausführlich ein.

Eine Sicherheitsschaltung mit Stromfühler-Widerstand R57 ($0,22 \Omega$) schützt Power-MOSFET T4 gegen Kurzschluss. Wenn der Strom durch R57 mehr als etwa 2,7 A beträgt, öffnet die an R57 abfallende Spannung den Transistor T5, so dass die Gate-Source-Spannung sinkt. Dies bedeutet, dass der Strom begrenzt wird. Jetzt muss der Mikrocontroller die Steuerung übernehmen und die Spannung am MOSFET wie auch den Wärmehaushalt des Kühlkörpers in erlaubten Grenzen halten. MOSFET T4 wird auch gegen Überspannung geschützt. Normalerweise beträgt die Spannung etwa 2,6 V. Wenn sie steigt, gerät T6 über D9 und R54 in den Leitzustand, so dass MOSFET T4 abgeriegelt wird. Beim Labornetzgerät in Vollausstattung übernimmt der Mikrocontroller die Überwachung. Falls das Labornetzgerät in der Minimalkonfiguration betrieben wird (ohne Mikrocontroller- und Anzeigeplatine), lässt sich diese Sicherung durch Setzen des Jumpers JP6 aktivieren.

Stellen und Regeln

Die Ausgangsspannung und die Strombegrenzung können mit den Mehrgang-Trimmpotis P3 und P4 auf der Hauptplatine eingestellt werden, oder der Mikrocontroller auf der noch erscheinenden Anzeigeplatine übernimmt diese Aufgabe. Die Einstellart ist mit den Jumpers JP4 und JP5 wählbar. Opamp IC2B puffert das von JP4 kommende Signal, das die Spannung einstellt. Widerstand R24 hinter dem Ausgang begrenzt den Strom, falls T3 durchschaltet. T3 zieht das Signal nach Masse, sobald der eingestellte Strom überschritten wird. Ist JP3 nicht vorhanden, wird der Strom ausschließlich vom Mikrocontroller, nicht von Hardware-Komponenten begrenzt.

Opamp IC2D vergleicht das Signal der Strombegrenzung, das an JP5 liegt, mit dem von IC3 gemessenen Ausgangsstrom. Ein stabiles Verhalten ist gewährleistet, weil R33, R34 und C22 die Verstärkung und die Bandbreite begrenzen. Schottky-Diode D2 schränkt den negativen Bereich der Ausgangsspannung von IC2D ein. Überschreitet der Geräte-Ausgangsstrom den eingestellten Wert, steuert IC2D den Transistor T3 in den Leitzustand, so dass der nicht invertierende Eingang von IC2C nach Masse gezogen wird.

Wie bereits beschrieben, ist das Spannungsgefälle an der mit MOSFET T4 aufgebauten analogen Regelstufe fest eingestellt. Der im Blockschema (Bild 1) gezeichnete *Level Shifter*, der für

das Spannungsgefälle verantwortlich ist, besteht in der Schaltung (Bild 3) aus Opamp IC2C und Schottky-Diode D4. Die Stellschaltung für den analogen Regler am Verbindungspunkt R43/R44 ist gleich der Spannung an JP4. Die Stellschaltung für den schaltenden Regler (R25) liegt wegen des Spannungsabfalls an D4 etwas höher. Der Wert von R43 ist maßgebend für den Strom durch D4 und daher auch für die Spannung an D4. Bei der Ausgangsspannung 0 V (5 V an R43) wurde an D4 die Spannung 0,252 V gemessen, bei 40 V (9 V an R43) lag der Messwert bei 0,268 V. Die Spannung an D4 kann als annähernd konstant

betrachtet werden. Da sowohl die Regelung um IC1 und IC2A als auch die Regelung um T4 und IC7 mit dem Faktor 10 verstärkt, resultiert daraus ein Spannungsgefälle zwischen den beiden Stufen von 2,52...2,68 V.

Opamp IC2A liefert das Stellsignal für den Rückkopplungseingang des Schaltreglers IC1. R27 und P2 sind hinzugefügt, um die interne Referenzspannung (0,8 V) zu kompensieren. Dazu liegt P2 an der stabilen Referenzspannung 2,5 V, die Spannungsnormal IC4 bereitstellt. Die Ausgangsspannung der schaltenden Stufe lässt sich auf 0 V stellen, wegen der nachfolgenden analogen Stufe

Stückliste

Widerstände (SMD 0805, wenn nicht anders angegeben):

R1 = 1k54, 1 %, 125 mW
 R2 = 52k3, 1 %, 125 mW
 R3,R18,R19,R56 = 100 Ω, 1 %, 125 mW
 R4 = 51 k, 1 %, 125 mW
 R5,R29,R34,R46,R52 = 10 k, 1 %, 125 mW
 R6 = 1M2, 1 %, 125 mW
 R7,R8,R21,R32 = 10 Ω, 1 %, 125 mW
 R9 = 0Ω01, 5 %, 1 W (TE Connectivity/CGS, CGSSL1R01J)
 R10,R11 = 2Ω7, 1 %, 125 mW
 R12 = 3Ω9, 5 %, 125 mW
 R13 = 10 Ω, 1 %, 0,75 W (Vishay Draloric, CRCW201010R0FKEF, 2010)
 R14 = 47 k, 1 %, 125 mW
 R15,R24,R36,R44,R45 = 1 k, 1 %, 125 mW
 R16 = 3k3, 1 %, 1 W (Multicomp, RCL1218 3K3 1% 100 PPM/K E3, 1218)
 R17 = 0Ω05, 1 %, 1 W (Bourns, CRM2010-FZ-R050ELF, 2010)
 R20,R23,R25,R26,R31,R47,R59 = 2k00, 1 %, 125 mW
 R22 = 44k2, 1 %, 125 mW
 R27 = 6k04, 1 %, 125 mW
 R28,R39 = 4k7, 1 %, 125 mW
 R30,R58 = 18k0, 1 %, 125 mW
 R33 = 1 M, 1 %, 125 mW
 R35,R43 = 3k3, 5 %, 125 mW
 R37 = 200 Ω, 1 %, 125 mW
 R38 = 470 Ω, 1 %, 125 mW
 R40,R41 = 1k5, 5 %, 125 mW
 R42 = 1k8, 5 %, 125 mW
 R48 = 16k9, 1 %, 125 mW
 R49 = 15 k, 5 %, 125 mW
 R50 = 47 Ω, 5 %, 125 mW
 R51 = 2k2, 5 %, 125 mW
 R53,R55 = 100 k, 5 %, 125 mW
 R54 = 820 Ω, 5 %, 125 mW
 R57 = 0Ω22, 5 %, 5 W (TE Connectivity/CGS, SMW5R22JT, SMW5R22JT)
 P1 = 2 k, 10 %, 0,5 W, Trimpoti 25 Gänge, Stellschraube oben (Bourns, 3296Y-1-202LF)
 P2,P4 = 500 Ω, 10 %, 0,5 W, Trimpoti 25 Gänge, Stellschraube oben (Bourns, 3296Y-1-501LF)
 P3 = 1 k, 10 %, 0,5 W, Trimpoti 25 Gänge, Stellschraube oben (Bourns, 3296Y-1-102LF)

Kondensatoren (SMD 0805, wenn nicht anders angegeben):

C1 = 2n2/50 V, 5 %, COG/NP0
 C2,C26 = 470 μ/63 V, 20 %, ESR 0,027 Ω, I_{AC} 2 A (Panasonic, EEUF11471)

C3,C9 = 1 μ/16 V, 10 %, X7R
 C4,C10,C45 = 470 n/25 V, 10 %, X7R
 C5 = 47 n/50 V, 10 %, X7R
 C6 = 15 n/50 V, 10 %, COG/NP0
 C7,C8,C12,C20,C48 = 1 n/100 V, 5 %, COG/NP0
 C11 = 1 μ/100 V, 10 %, X7R (Murata, GRM32CR72A105KA35L, 1210)
 C13 = 1000 μ/63 V, 20 %, D 16 mm, Raster 7,5 mm, I_{AC} 2,77 A (Panasonic, EEUF11J102)
 C14..C16,C18,C23..C25,C27..C29 = 2μ2/100 V, 10 %, X7R (TDK, C3225X7R2A225K, 1210)
 C17 = 47 μ/50 V, 20 %, D 10 mm, Raster 5 mm, ESR 29 mΩ (Nichicon, PLX1H470MDL1TD)
 C19,C30,C31,C49,C50 = 100 n/25 V, 10 %, X7R
 C21,C22,C47 = 10 n/50 V, 10 %, X7R
 C32,C33 = 47 μ/35 V, 20 %, Raster 2,5 mm
 C34,C35,C38,C39 = 100 n/63 V, 5 %, Raster 5/7,5 mm
 C36,C44 = 1000 μ/25 V, 20 %, D 10 mm, Raster 5 mm, ESR 0,02 Ω (Panasonic, EEUF1E102)
 C37 = 220 μ/25 V, 20 %, D 8 mm, Raster 3,5 mm, I_{AC} 950 mA (Panasonic, ECA1CM221)
 C40..C43 = 10 n/100 V, 10 %, Raster 5 mm
 C46 = 470 p/100 V, 10 %, X7R
 C51,C52,C54 = 100 n/100 V, 10 %, X7R
 C53 = 10 μ/63 V, 20 % D 6,3 mm, Raster 2,5 mm (Rubycon, 100YXF10MEFC6.3X11)

Induktivitäten:

L1 = 82 μH, 7 A, SMD, High-current, R_{DC} 30,4 mΩ (Würth Elektronik, 74435588200)
 L2 = 10 μH, 7,2 A, SMD, High-current, R_{DC} 16,3 mΩ (Würth Elektronik, 7443251000)
 L3 = 10 μH, 950 mA, stehend, Raster 2 mm, R_{DC} 0,14 Ω (Murata Power Solutions, 11R103C)
 L4 = 70 Ω @ 100 MHz, 3,5 A, 0,022 Ω, 0603 (Murata, BLM18KG700TN1D)

Halbleiter:

D1 = PMEG6010CEH (SOD-123F)
 D2,D3,D4 = BAT854W (SOT-123)
 D5 = LED rot 3 mm, mit Drahtanschlüssen
 D6 = LED grün 3 mm, mit Drahtanschlüssen
 D7,D8,D11,D12,D13 = PMEG6030EP (SOD-128)
 D9 = Zenerdiode 4V7/500 mW (SOD-123F)
 D10 = Zenerdiode 10 V/3 W (SMB)
 B1 = Brückengleichrichter 50 V/1,5 A (Multicomp, AM150)
 T1,T2 = PSMN5R5-60YS (LPAK/SOT669)
 T3 = BC817-25 (SOT-23)

ist dies jedoch nicht notwendig. Die Schottky-Diode D3 am Ausgang von IC2A verhindert, dass die Ausgangsspannung negative Werte annimmt. Beim Schaltregler LM5117 (IC1) sind negative Eingangsspannungen nicht erlaubt. Eine weitere Abhandlung zur Dimensionierung dieses Schaltungsteils haben wir auf unsere Website *elektor-labs* [4] gestellt.

Verbindungen

Die relevanten Signale und Spannungen sind auf Steckverbinder K3 zusammengeführt. K3 wird mit der Mikrocontroller- und Anzeige-Platine verbunden

den, die in einem nachfolgenden Beitrag veröffentlicht wird. An Pin 1 und Pin 3 von K1 liegen Informationen über den gemessenen Ausgangsstrom (1 V/A) und die gemessene Ausgangsspannung ($V_{out}/10$ über Spannungsteiler R30/R31). Pin 5 und Pin 7 sind Eingänge der vom Mikrocontroller kommenden Stellspannungen für die Ausgangsspannung und die Strombegrenzung. An Pin 11 ist die Spannung verfügbar, die am Ausgang der schaltenden Stufe vorhanden ist ($V_{smps}/10$ über R58/R59), während an Pin 13 die auf +1 V herabgeteilte Eingangsspannung liegt (+48 V über R14/R15).

T4 = AU1RF9540N (TO-220)
T5,T6 = BC807-40 (SOT-23)
T7 = PMBFJ110 (SOT-23)
IC1 = LM5117 (MXA20A)
IC2 = LM6134 (SO-14)
IC3 = LT6105 (MSOP-8)
IC4 = LM336Z-2.5 (TO-92)
IC5 = 7805 (TO-220)
IC6 = 7905 (TO-220)
IC7 = OPA552UA (SO-8)

Außerdem:

K1,K2 = Kabelschraubklemme 2-polig, Raster 5 mm
K3 = Wannenstecker 14-polig, Raster 2,54 mm
K4,K5 = Kabelschraubklemme 2-polig, Raster 7,5 mm

K6,JP1,JP3,JP6 = Stiftkontaktleiste 2-polig, Raster 2,54 mm
JP2,JP4,JP5 = Stiftkontaktleiste 3-polig, Raster 2,54 mm
JP1..JP6 = Jumper, Raster 2,54 mm
S1 = Drucktaster 6 · 6 mm SPST-NO (TE Connectivity/Alcoswitch, FSM4JH)
F1 = Sicherungshalter für Platinenmontage, mit Sicherung 1 AT und Schutzkappe
TR1 = Netztrafo für Platinenmontage, prim. 2 · 115 V, sek. 2 · 9 V, 3,2 VA (Block, AVB3.2/2/9)
HS1 = Kühlkörper für Platinenmontage, z. B. Fischer SK 129 38,1 STS, 6,5 K/W
Schaltnetzteil 48 V/3 A, z. B. Mean Well S-150-48
Platine 120437-1

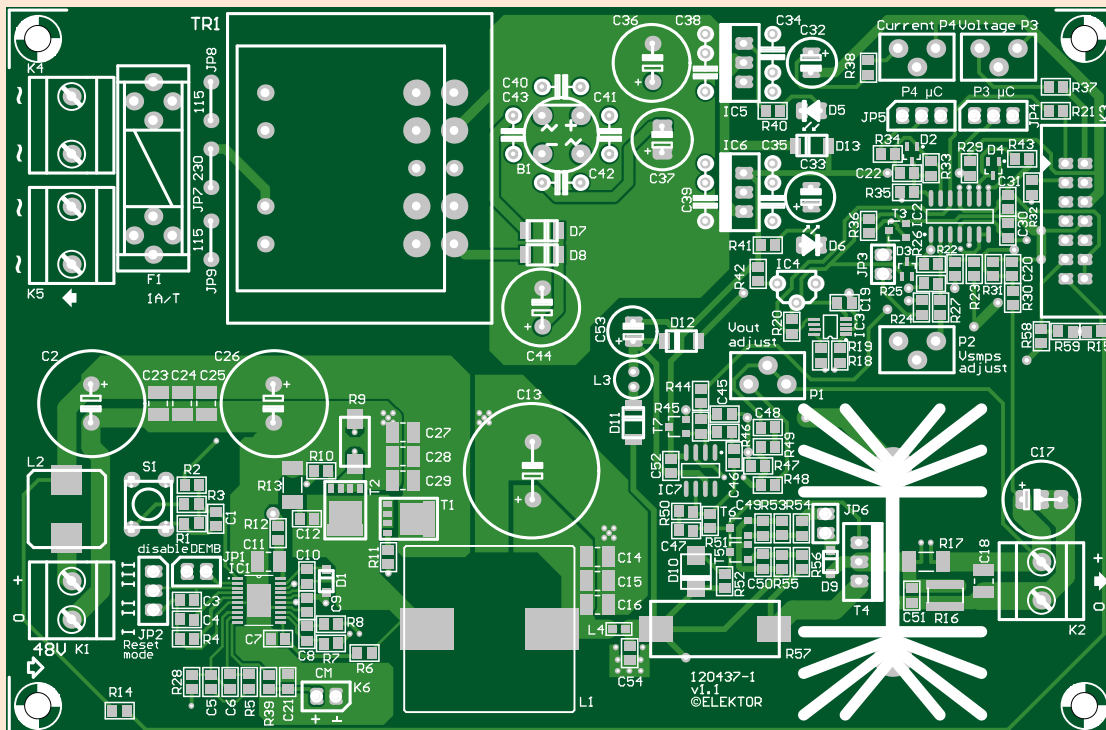


Bild 4.
Auf der Platine mischen sich SMDs mit bedrahteten Komponenten, die Montage erfordert Lötverfahren.

Stromversorgung

Ein industriell gefertigtes Schaltnetzteil des Stromversorgungsspezialisten Mean Well liefert die Eingangsgleichspannung 48 V, der Strom beträgt 3 A. Natürlich sind auch andere Schaltnetzteile mit vergleichbaren Eigenschaften einsetzbar.

Für die Betriebsspannungen der Opamps ist ein separates analoges Netzteil vorhanden, so dass Störspitzen und Rauschen aus dem Schaltnetzteil umgangen werden. Das analoge Netzteil ist mit Platinentrafo TR1 (115 V/230 V, umschaltbar mit JP7...JP9), Brückengleichrichter B1 und zwei Spannungsreglern (IC5, IC6) aufgebaut. Wie erwähnt, liefert der Trafo über D7 und D8 auch die Betriebsspannung für den Schaltregler IC1. Der Ausgang des negativen Spannungsreglers IC6 wird von Schottky-Diode D13 gegen positive Spannungsspitzen, beispielsweise beim Ausschalten geschützt. Die Ausgangsspannung von IC6 kann +0,2 V nicht übersteigen. Die Spannungen +5 V und -5 V sind auf Steckverbinder K3 für die Mikrocontroller- und Anzeigeplatine herausgeführt. Die LEDs D5 und D6 leuchten auf, wenn die Spannungen vorhanden sind. Mit Klemme K5 kann der Netzanschluss des Schaltnetzteils verbunden werden, das die Eingangsspannung 48 V liefert. Sicherung F1 erhöht die Sicherheit des Schaltnetzteils.

Alle Leitungen, die Störspannungen und Störspitzen transportieren können, sind durch wirksame Maßnahmen entkoppelt. Zu den Maßnahmen zählen Drossel L2 ($10\ \mu\text{H}$, $R_{\text{DC}} = 16,3\ \text{m}\Omega$) sowie die nachfolgenden, parallel geschalteten Kondensatoren C2, C23...C25. Diese Kombination hat ebenso wie C26...C29 einen ESR von nur $27\ \text{m}\Omega$, die Ströme können bis 2 A bei 100 kHz betragen. Die übrigen Kondensatoren sind SMDs, diese Bauform hat sich bei hohen Frequenzen bewährt.

Aufbau

Die vorstehende Beschreibung ist lang, wir haben jedoch nahezu jedes Bauelement einzeln betrachtet. **Bild 4** zeigt die doppelseitige Platine, die wir für unser Labornetzgerät „VariLab 402“ entworfen haben. Der Aufbau ist ohne Besonderheiten, allerdings ist Erfahrung im Löten von SMD-Bauelementen Voraussetzung für den Erfolg. Die Angaben in der Stückliste sind konsequent zu befolgen, insbesondere dürfen die vorgeschriebenen Elkos nicht durch minderwertige Typen ersetzt werden. MOSFET T4 wird isoliert auf einen klei-

nen Kühlkörper montiert. Die isolierte Montage ist notwendig, da der Kühlkörper über die Haltestifte an Masse liegt.

Stellen Sie mit einer beziehungsweise zwei Drahtbrücken die Netzspannung ein. Bei 230 V muss eine Drahtbrücke auf JP7, bei 115 V eine auf JP8, die zweite auf JP9.

Im Prinzip können Sie bereits jetzt ein Netzteil anschließen, das 48 V Gleichspannung liefert, und die montierte Platine auf Funktion testen. Zuvor müssen Sie nur noch Jumper auf JP3, JP4 (Verbindung P3), JP5 (Verbindung P4) und JP6 stecken. Bringen Sie vorläufig P1 und P2 in Mittelstellung und schalten Sie die Netzspannung ein. Schließen Sie eine niedrige Last zusammen mit einem Spannungs- und Strommesser an K2 an. Prüfen Sie, ob die Spannung mit P3 einstellbar ist. Mit P4 muss sich der Strom begrenzen lassen, dabei muss die Spannung sinken.



Wenn Ihre Mühe bis hier erfolgreich war, sind Sie gerüstet für den nächsten Beitrag zum „VariLab 402“, in dem es um die Mikrocontroller- und Anzeigeplatine einschließlich zugehöriger Software geht. Auch der Einbau der Gerätekomponten in ein Gehäuse wird ein Thema sein.

(120437)gd

Weblinks

- [1] www.ti.com/lit/ds/symlink/lm5117.pdf
- [2] www.ti.com/lit/ug/snva466b/snva466b.pdf
- [3] <http://cds.linear.com/docs/en/datasheet/6105fa.pdf>
- [4] www.elektor-labs.com/120437

**Don't miss
the future!**

- ✓ **Täglich Live-Demos** aus den Bereichen IoT, Messtechnik, Bauelemente und Entwicklungskits
- ✓ Reisen Sie in das Jahr **2025**
- ✓ Holen Sie sich Ihren **3D Mini-Zwilling**
- ✓ Ziehen Sie das „**Golden Ticket**“!

**Explore
the
Future**
Hall A5.462

Mehr Infos und kostenfreie Tickets unter:
electronica.conrad.de

CONRAD
Business Supplies

Android Open Accessory

Elektronik steuern mit Android und Arduino



Android Open Accessory (kurz AOA) ist ein einfaches und sicheres Protokoll zur Verbindung von Mikrocontroller-gesteuerten Geräten mit einem Android-Smartphone oder -Tablet. Dieses Buch zeigt anhand von leicht nachbaubaren Schaltungen und den dazu gehörenden Programmbeispielen, wie man AOA in Verbindung mit der Mikrocontroller-Plattform Arduino verwendet, um täglich anfallende Aufgaben im Haus zu automatisieren: Beleuchtung, Belüftung, Klimatisierung und Musik-Entertainment-Systeme – bequem und komfortabel mit dem Smartphone, wohlgemerkt!

Die Grundkenntnisse des Arduino-Frameworks voraussetzend, versorgt das visionäre Autorenduo Göransson/Cuartielles Ruiz den Leser mit den Werkzeugen (Tools), die er braucht, um nützliche und anspruchsvolle Projekte realisieren zu können.

Die Programmbeispiele aus diesem Buch stehen auf der Elektor-Website zum Gratis-Download bereit.

380 Seiten (kart.) • Format 18,8 x 23,5 cm
ISBN 978-3-89576-279-6

€ 42,00
CHF 52,95

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

USB-Hub/ UART-Konverter mit RS232 und RS422/RS485

Ohne Mikrocontroller, ohne Software!

Von **Sébastien
Guerreiro de Brito**
(F)

Die serielle RS232-Schnittstelle, die früher an jedem PC als neunpoliger D-Sub-Steckverbinder existent war, gehört der Vergangenheit an. Ältere periphere Geräte mit ausschließlich dieser Schnittstelle sind zwar über Adapter weiterhin nutzbar, der Anschluss mehrerer Geräte jener Generation führt jedoch zu Konflikten. Unser USB-Hub/UART-Konverter mit zwei RS232- und zwei RS422-Ports löst das Problem.

Als die serielle RS232-Schnittstelle verschwand, kam die Zeit der USB/RS232-Konverter. Das ist meistens eine praktikable Lösung, doch wenn mehrere Geräte anzuschließen sind, braucht jedes Gerät einen eigenen Konverter. Um außerdem Geräte nach RS422 oder RS485 anzuschließen, sind spezielle Konverter nötig, sie sind leider nicht weit verbreitet.

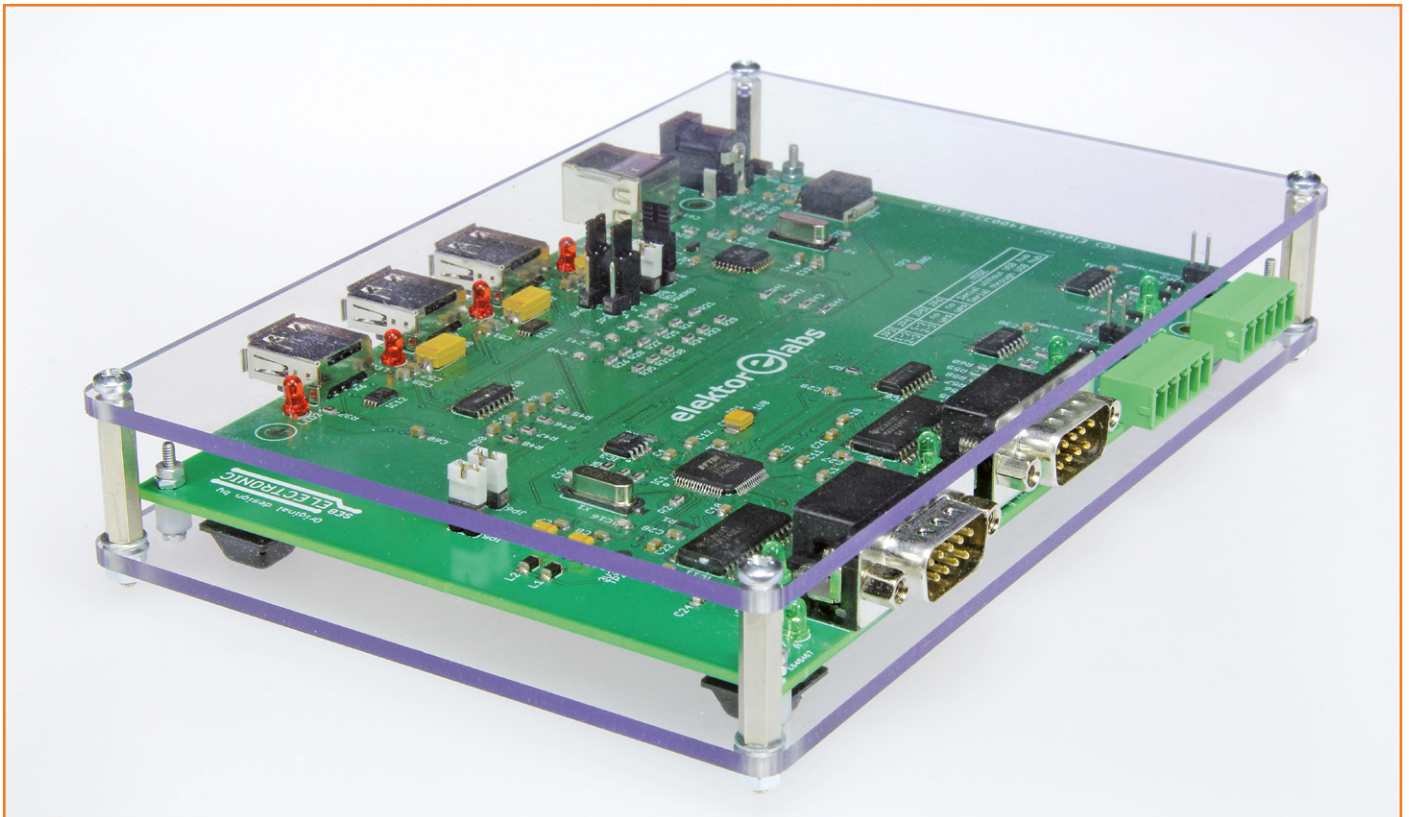
Diese Handicaps brachten mich auf die Idee, einen Konverter zu entwickeln, der gleichzeitig mehrere RS232- und RS422/RS485-Verbindungen herstellen kann. Ich begann mit der Suche nach einem geeigneten Gehäuse, was nicht einfach war, denn in dieses Gehäuse muss eine Platine mit überdurchschnittlich vielen Anschlussbuchsen passen. Oder aus anderer Sicht betrachtet: Die Gehäusebauform beeinflusst die Abmessungen der Platine und damit auch das Konzept der Schaltung. Das Gehäuse, das ich nach einiger Suche fand, bietet nicht nur Platz für die Anschlussbuchsen, meine erste Platine füllte nur die Hälfte des Gehäusevolumens aus. Den freien Raum habe ich genutzt, um einen USB-Hub zu integrieren.

Zwei Etappen

Mein Projekt habe ich in zwei Etappen realisiert: Den Anfang machte der USB/UART-Konverter, dann folgte der USB-Hub. Auch aus diesem Grund ist das Funktionsschema in **Bild 1** deutlich zweigeteilt. Die detaillierte Schaltung ist in **Bild 2** dargestellt.

Zuerst sollen die Wege der Betriebsspannung betrachtet werden. Die Betriebsspannung wird entweder über die USB-Schnittstelle K1 dem Host-PC entnommen, oder die Gleichspannung +5 V eines externen Netzadapters liegt an K5. Jumper JP8 muss offen gelassen werden, wenn die Schaltung über USB versorgt werden soll. Wird ein Netzadapter verwendet, schützt Schottky-Diode D9 die übrigen Bauteile gegen Verpolung. Spannungsregler IC2 setzt die Spannung +5 V nach +3,3 V um, +3,3 V ist hier überwiegend die Betriebsspannung der aktiven Bausteine.

Die meisten Funktionen des USB-Hubs stecken in IC9, einem TUSB2046BVF von Texas Instruments. Dieser Baustein steuert auch IC10, einen *Quad Power Distribution Switch* mit der Typenbezeichnung TPS2044. Er verteilt die Betriebsspannung



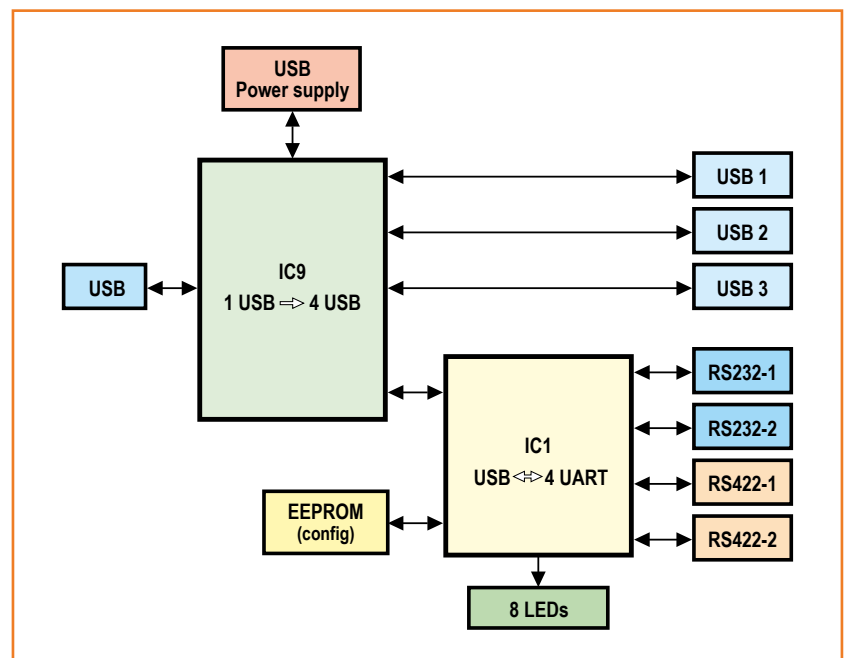
und den Betriebsstrom von einem Eingangsport (K1) auf vier Ausgangsports. Intern arbeitet eine so genannte *State Machine*, die ohne Steuerung durch Software auskommt. Ein Mikrocontroller ist hier nicht vorhanden!

Der Betriebsstrom wird von IC10 nicht nur auf die USB-Ports verteilt, sondern auch auf den UART-Teil der Schaltung. Für jeden Kanal sind maximal 500 mA verfügbar. Der Baustein sorgt dafür, dass diese Grenze nicht überschritten wird, gleichzeitig schützt er vor thermischer Überlastung. Falls der Strom die Grenze von 500 mA überschreitet, werden ein oder mehrere *Overcurrent*-Ausgänge (OC) aktiv. Diese Open-Drain-Ausgänge werden von den Pullup-Widerständen R45...R48 auf die Betriebsspannung gezogen. Der Betriebsstrom wird schaltend verteilt, der Grenzwert kann dabei kurzzeitig überschritten werden. Die OC-Ausgänge würden auf diese Spitzen reagieren, wenn sie nicht von den angeschalteten RC-Gliedern abgeflacht würden.

LED1...LED5 signalisieren die Belegung der Anschlüsse. Wenn der USB-Hub aktiv ist, leuchtet LED1, während die Leuchtdioden LED2...LED4 zu den Ports des Hubs gehören.

Die Eingänge und Ausgänge können mit Störsignalen behaftet sein. Wenn die Störsignale hohe Intensität haben und von langer Dauer

Bild 1.
Funktionsschema des
USB-Hub/UART-Konverters.



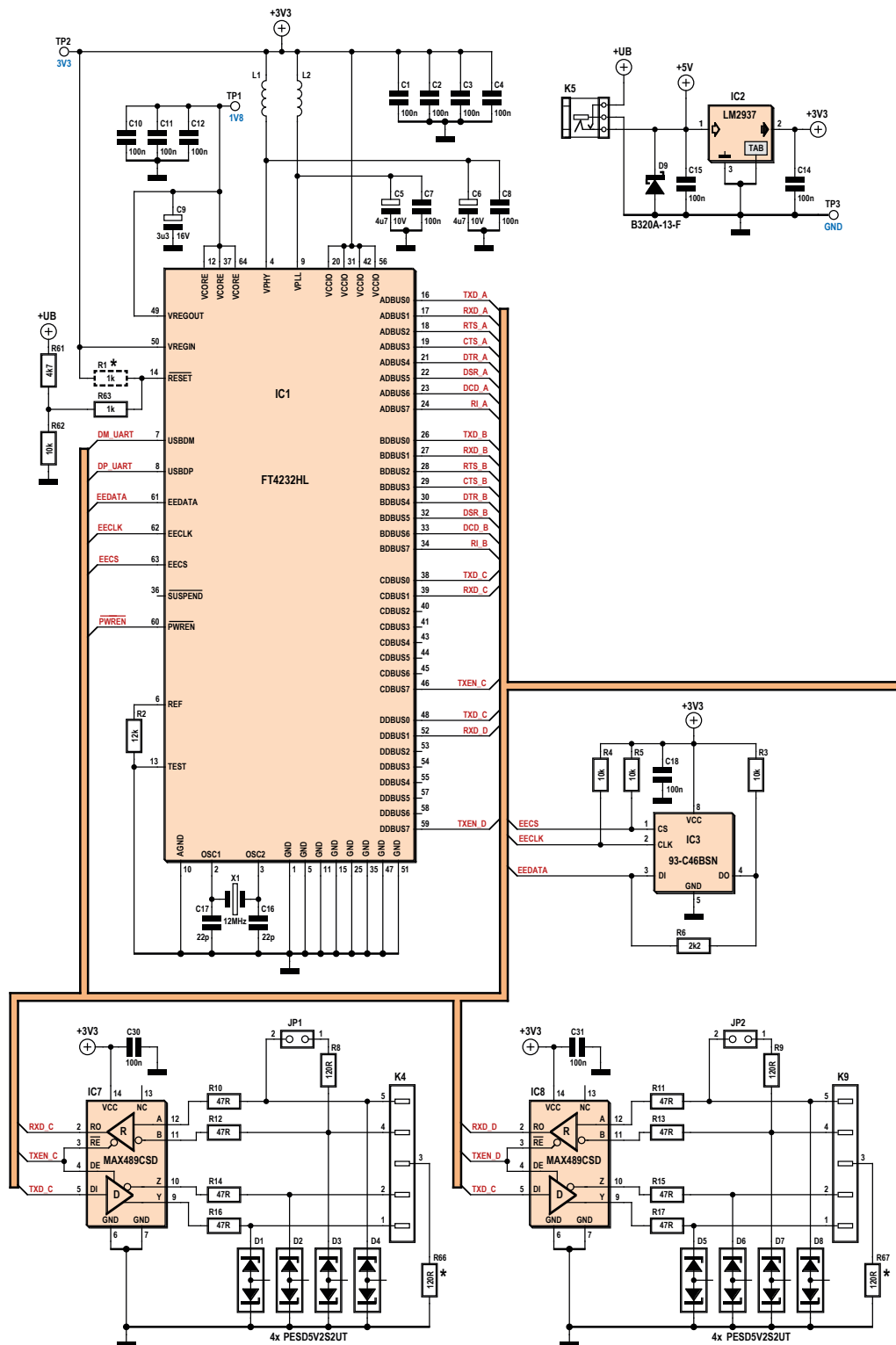
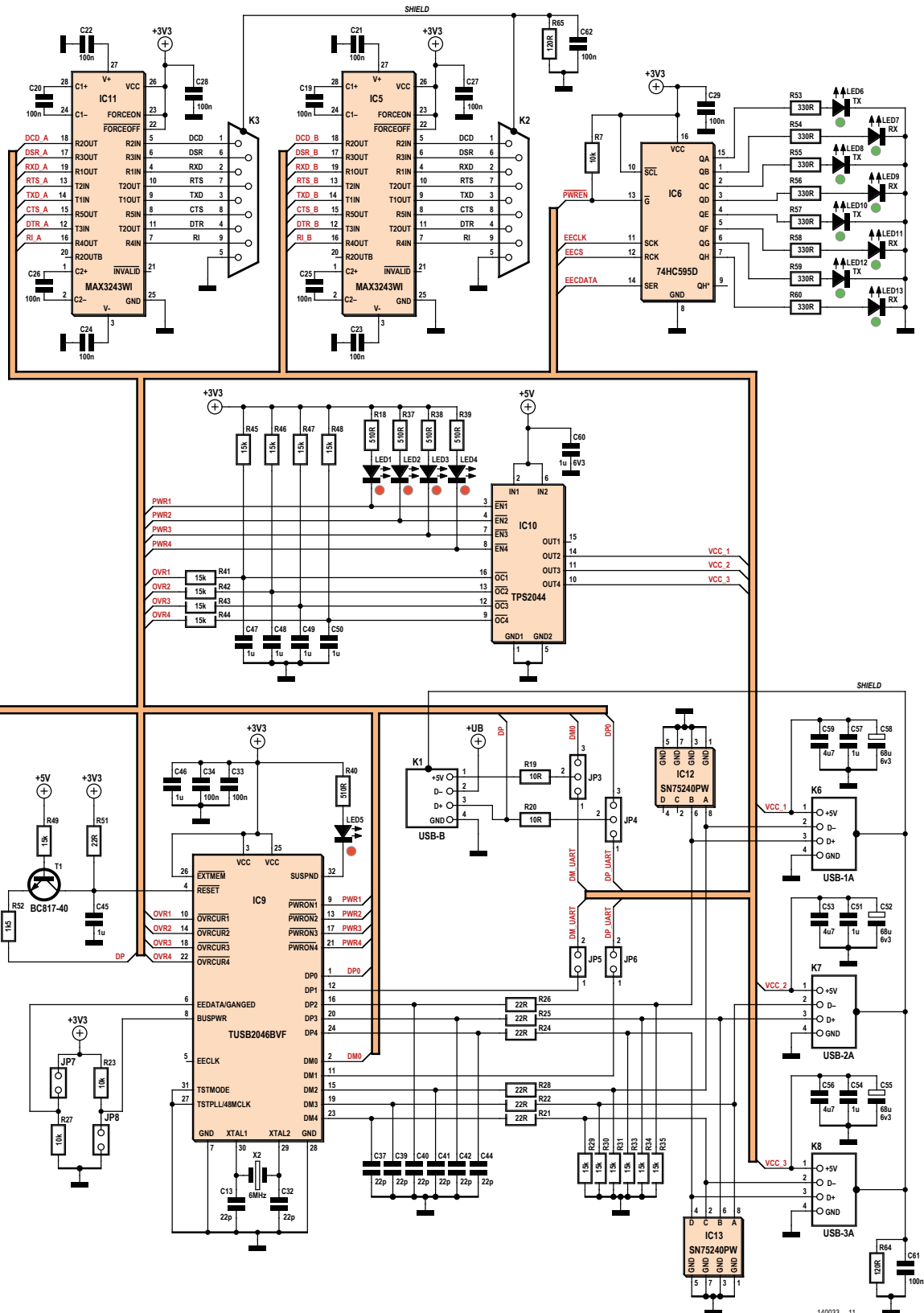


Bild 2.
Die Schaltung kommt ohne
Mikrocontroller aus!



sind, besteht die Gefahr, dass der USB-Hub und der USB/UART-Konverter Schaden nehmen. Um dem vorzubeugen, sichern IC12 und IC13 die Schaltung gegen schädigende Einflüsse. Der SN74240PW, ebenfalls von Texas Instruments, ist ein zweifacher *Transient Voltage Suppressor* (ESD-Schutz), der allerdings die USB-Datenleitungen kapazitiv belastet. Aus diesem Grund ist der Konverter für hohe Übertragungsgeschwindigkeiten, wie sie beim USB-Standard 2.0 auftreten, nur bedingt geeignet.

UART

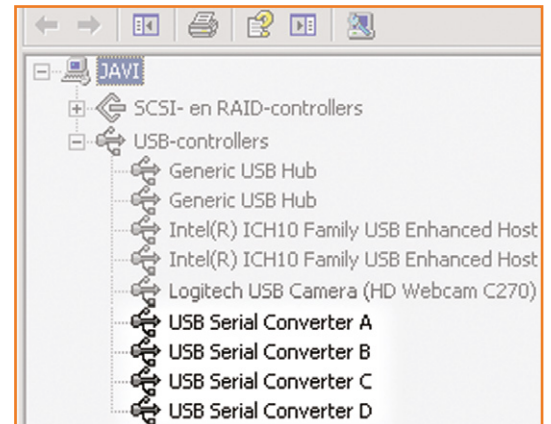
Der Funktionsblock „UART“ in Bild 1 wird in der Schaltung (Bild 2) von IC1 verkörpert, ein FT4232HL von FTDI. Dieser Baustein ist das Bindeglied zwischen der USB-2.0-Schnittstelle (480 Mbit/s) und den vier UARTs (*Universal Asynchronous Receiver Transmitter*). Vorhanden sind zwei UARTs für RS232 sowie zwei weitere UARTs für RS422/485. Ich komme später darauf zurück. Die Konfiguration wird in IC3, einem EEPROM, nicht flüchtig gespeichert.

Wenn Daten über die seriellen Kanäle ausgetauscht werden, leuchten LED6...LED13 auf. Diese LEDs werden von IC6 gesteuert, einem 8-bit-Seriell-In/Parallel-Out-Schieberegister, es ist mit Konverter IC1 verbunden. Ausgang $\overline{PWREN}$ steuert den Eingang *Output Enable* ($\overline{OE}$) von IC6. Damit wird verhindert, dass die LEDs willkürlich aufleuchten, solange der USB-Wandler noch nicht vom PC erkannt wurde.

Die Signale an den Ausgängen von IC1 haben TTL-Pegel, sie werden von IC11 und IC5, zwei *Line Drive Receiver* MAX3243EIDW, vollduplex nach RS232 umgesetzt. Die zugehörigen D-Sub-Steckverbinder sind K2 und K3 (rechts oben in Bild 2). Der MAX3243EIDW ist gegen statische Elektrizität bis ± 15 kV (*Human Body Model*) und ± 8 kV bei Berührung geschützt.

Das RC-Glied R65/C62 verbindet die Abschirmung der Steckverbinder mit Masse und dämpft gemäß EMC-Richtlinien die Abstrahlung nach außen. Eine identische Funktion hat das RC-Glied R64/C61 für die USB-Buchsen.

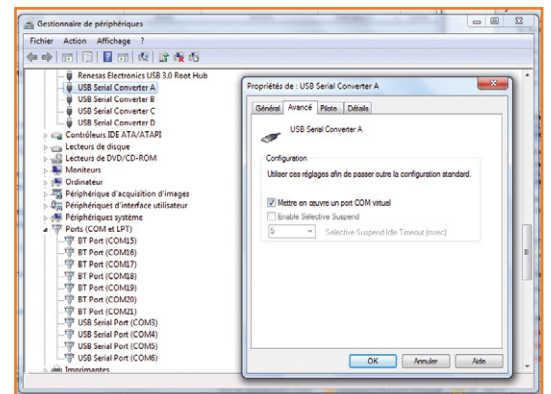
IC7 und IC8, zwei Transceiver vom Typ MAX489CSD+, sind für die RS422/485-Schnittstellen zuständig. Die Ausgänge dieser Bausteine werden von D1...D8 gegen ESD-Schäden geschützt. Was noch fehlt, sind die Abschlusswiderstände an den Eingängen. Über die Jumper JP1 und JP2 können Abschlusswiderstände



120 Ω zugeschaltet werden (siehe Kasten „Abschlusswiderstände“).

Wenn die Schaltung erstmalig mit dem PC verbunden wird, fordert das Betriebssystem die Installation von USB-Treibern an [1]. Nachdem die Treiber installiert sind, verfügt der PC über vier virtuelle COM-Ports, zwei RS232- und zwei RS422/485-Ports.

Der USB-Hub/UART-Konverter wurde unter Windows und Linux getestet. Inzwischen wird er im Elektor-Labor häufig und vielfältig genutzt. Noch eine Anmerkung: Für das Betriebssystem ist ein USB/Seriell-Konverter nicht identisch mit einem seriellen Port. Um ihn als seriellen Port zu konfigurieren, müssen im „Geräte-Manager“ der Systemsteuerung die „Erweiterten Eigenschaften“ der betreffenden Schnittstelle geöffnet werden. Dort muss die Option „VCP“ durch einen Haken aktiviert sein. Diese Prozedur ist für jeden USB/Seriell-Konverter einzeln vorzunehmen. Nach Unterbrechen der USB-Verbindung und dem Wiederanschießen dauert es kurze Zeit, bis vier serielle Ports verfügbar sind (vergleiche Bild 6).



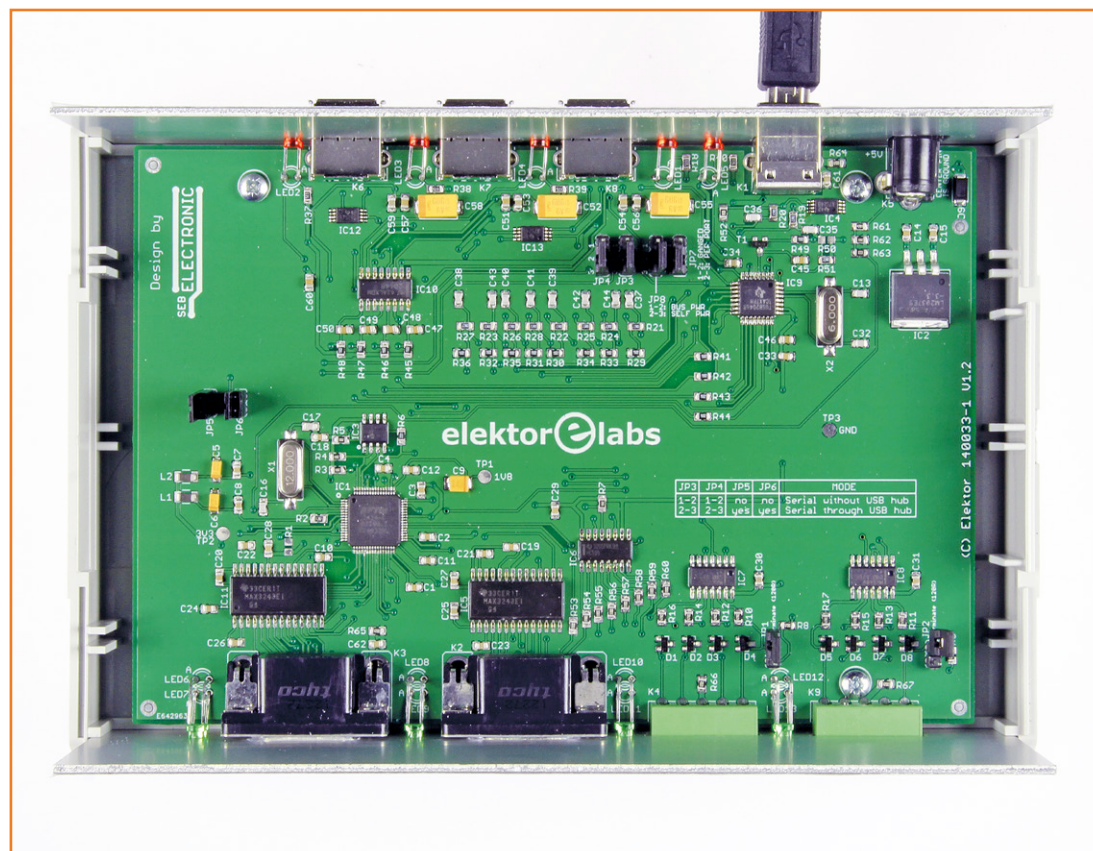
Das getestete, einsatzfähige Modul ist im Elektor-Shop erhältlich: www.elektor.de/USB-serial-hub

Mechanisches

Die Fotos vom Prototypen geben Hilfestellung, die Bauelemente der Stückliste auf der Platine (**Bild 3**) erfolgreich zu montieren. Voraussetzung ist allerdings, dass ein Lötgerät für SMDs verfügbar ist, und natürlich kann auch etwas Übung im Umgang mit diesen Bauelementen nicht schaden. Bei IC1, IC9, IC12 und IC13 liegen die Pins sehr eng beieinander, Geduld und Sorgfalt sind hier oberstes Gebot. Die Anschlüsse gepolter Bauelemente, beispielsweise der Tantalelkos C5, C6, C9, C52, C55 und C58, dürfen nicht vertauscht werden. Widerstand R1 entfällt, sein Platz auf der Platine ist ein Überbleibsel der ersten Version. Die grünen LEDs werden paarweise übereinander montiert. Vor dem Löten sind die Anschlussdrähte so abzuwinkeln, dass die LEDs nach vorn leuchten.

ten. Damit die LEDs in gleicher vertikaler Ebene liegen, werden sie am Platinenrand ausgerichtet. In vielen Schränken und Regalen fristen noch voll funktionsfähige Geräte ihr Dasein, die den einzigen Nachteil haben, dass nur ein RS232- oder RS422/485-Anschluss vorhanden ist. Weil nicht jeder Besitzer solcher Geräte über die zum Löten von SMDs nötige Ausstattung verfügt, bietet der Elektor-Shop den USB-Hub/UART-Konverter auch als getestetes, einsatzfähiges Modul an. Die unbestückte Platine ist selbstverständlich ebenfalls lieferbar, weitere Informationen können wie immer der Projektseite [2] entnommen werden.

Eine passende Unterkunft für die aufgebaute Platine ist das in der Stückliste genannte Gehäuse von Vero. Die Bohrschablonen für die Front- und Rückseite stehen auf der Projektseite [2]



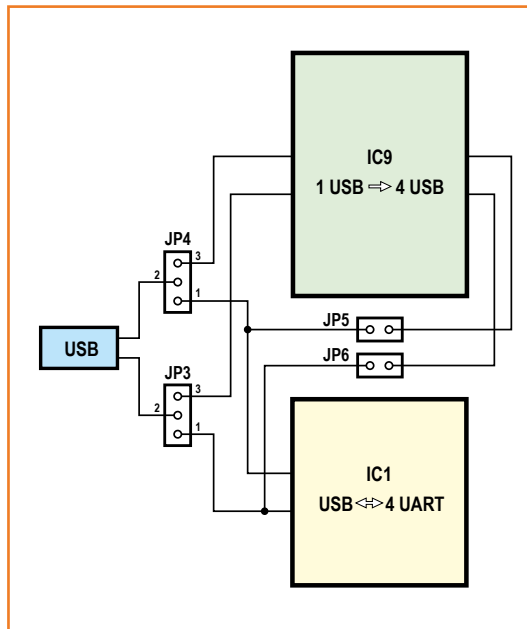


Bild 4.
Konfiguration der Jumper JP3, JP4, JP5 und JP6.

im DXF- und FPD-Format zum Download bereit. Das FPD-Format ist ein Format des Frontplatten- und Gehäuse-Spezialisten Schaeffer [3]. Zu Stromversorgungsbuchse K5 ist anzumerken, dass bei der ersten Version der Bohrschablone der Durchbruch für K5 fehlte. Noch kurz vor Redaktionsschluss wurde die Buchse sorgfältig vermessen und der zugehörige Durchbruch nachgetragen. Aus Zeitmangel konnte jedoch nicht sichergestellt werden, dass alles exakt passt. Deshalb ist es ratsam, sich vor dem Bohren und Sägen über die Stimmigkeit der Maßangaben zu vergewissern. Achtung: Der zentrale Stift der Stromversorgungsbuchse K5 ist der Masse-Anschluss! Bei dem in der Stückliste angegebenen Gehäuse von Vero fehlt die Befestigungsbohrung neben dem D-Sub-Steckverbinder K3. Um die Platine trotzdem sicher zu befestigen, können die D-Sub-Steckverbinder mit der Frontplatte verschraubt werden.

Stückliste

Widerstände: (SMD 0805)

R1 = entfällt
R2 = 12 k, 1 %
R3,R4,R5,R7,R62 = 10 k
R6 = 2k2
R8,R9,R64,R65,R66,R67 = 120 Ω
R10,R11,R12,R13,R14,R15,R16,R17 = 47 Ω
R18,R37,R38,R39,R40 = 510 Ω
R19,R20 = 10 Ω
R21,R22,R24,R25,R26,R28,R51 = 22 Ω
R23,R27 = 10 k
R29,R30,R31,R33,R34,R35,R41,R42,R43,R44,R45,R46,
R47,R48,R49 = 15 k
R52 = 1k5
R53,R54,R55,R56,R57,R58,R59,R60 = 330 Ω
R61 = 4k7
R63 = 1 k

Kondensatoren:

(SMD 0805, wenn nicht anders angegeben)
C1...C4,C7,C8,C10...C12,C14,C15,C18...
C31,C33,C34,C61,C62 = 100 n
C5,C6 = 4,7 μ /10 V, Tantal (A)
C9 = 3,3 μ /16 V, Tantal (E)
C13,C16,C17,C32,C37,C39,C40,C41,C42,C44 = 22 p
C45...C51,C54,C57,C60 = 1 μ
C52, C55, C58 = 68 μ /6,3 V, Tantal (D)
C53,C56,C59 = 4 μ 7

Induktivitäten:

L1,L2 = Drossel mit Ferritkern, 70 Ω @ 100 MHz,
SMD 1206

Halbleiter:

D1...D8 = PESD5V2S2UT
D9 = B320A-13-F
LED1,LED2,LED3,LED4,LED5 = LED rot, 3 mm

LED6,LED7,LED8,LED9,LED10,LED11,LED12,LED13 =
LED grün, 3 mm
T1 = BC 817-40
IC1 = FT4232HL
IC2 = LM2937ES-3.3
IC3 = 93LC46B-I/SN
IC12,IC13 = SN75240PWR
IC5,IC11 = MAX3243EIDW
IC6 = 74HC595D
IC7,IC8 = MAX489CSD
IC9 = TUSB2046BVF
IC10 = TPS2044BD

Außerdem:

JP1,JP2,JP5,JP6,JP7,JP8 = Stiftleiste 2-polig
(für JP1,JP2 ggf. Abschlusswiderstand 120 Ω bei
RS485, siehe Kasten)
JP3,JP4 = Stiftleiste 3-polig
K1 = USB-Buchse Typ B abgewinkelt für
Platinenmontage
K2,K3 = D-Sub-Steckverbinder 9-polig für
Platinenmontage
K4,K9 = Stiftleiste 5-polig abgewinkelt mit Führungs-
kragen, Raster 3,81 mm
K5 = Stromversorgungsbuchse für Platinenmontage
12 V/3 A mit Schaltkontakt, Mittelstift 1,95 mm
K6,K7,K8 = USB-Buchse Typ A abgewinkelt für
Platinenmontage
X1 = Quarz 12 MHz, SMD
X2 = Quarz 6 MHz, SMD
Gehäuse 65 · 180 · 120 mm, z. B. Vero 75-265742
Platine 140033-1
Aufgebautes und getestetes Modul: 140033-91

Anmerkung: Einige Bauelemente wie IC4 und R50 erscheinen nicht in der Stückliste. Sie wurden weggelassen, weil sie in der endgültigen Schaltung nicht mehr vorkommen. Die verbleibenden Bauelemente wurden nicht neu nummeriert.

Bild 3.
Platine für den USB-Hub/
UART-Konverter.

Konfiguration, Einstellungen, Treiber

Mit den Jumpern JP3...JP6 (siehe **Bild 4**) kann die UART-Schaltung vom USB-Hub getrennt werden. Die Trennung ist notwendig, um die UART-Schaltung für eine RS485-Schnittstelle zu konfigurieren.

Die Brücken werden dann wie folgt gesetzt:

- Brücke auf Pin 1 und 2 von JP4
- Brücke auf Pin 1 und 2 von JP3
- JP5 und JP6 bleiben offen

Im normalen Betrieb müssen die Brücken wie folgt gesetzt sein:

- Brücke auf Pin 2 und 3 von JP4
- Brücke auf Pin 2 und 3 von JP3
- Brücke auf JP5
- Brücke auf JP6

Die Funktionen von IC9 werden mit zwei weiteren Jumpern festgelegt:

Jumper				
JP	Funktion			
1	R _t 120 Ω*			
2	R _t 120 Ω*			
3	1-2 :	serial without hub	2-3 :	serial through hub
4	1-2 :		2-3 :	
5	0 :		1 :	
6	0 :		1 :	
7	1 : all outputs switched		0 : single output switched	
8	1 : external power		0 : bus powered	
1 = présent 0 = absent * (option, cf texte)				

- Von JP7 hängt die Reaktion ab, wenn versucht wird, einem Port einen zu hohen Strom zu entnehmen. Liegt Eingang GANGED (Pin 6) an +3,3 V (Brücke auf JP7), werden

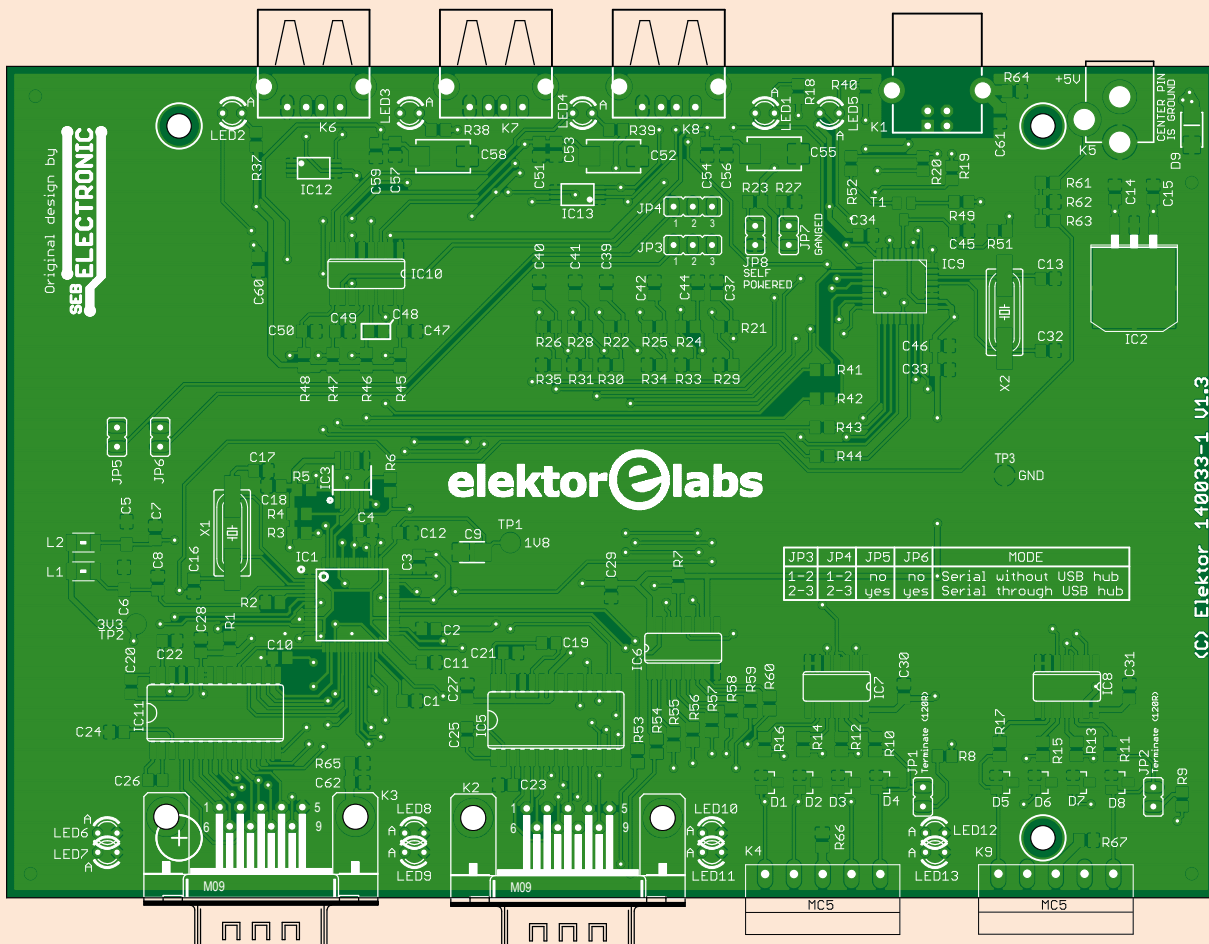




Bild 5.
Platine aufgebaut im
Gehäuse, mit dem
Frontplatten-Design des
Autors.

sämtliche Ausgänge abgeschaltet. Bei Eingang GANGED an 0 V (JP7 offen), wird nur der betroffene Ausgang abgeschaltet.

- Die Einstellung von JP8 signalisiert die Art der Stromquelle. Ist JP8 offen, liegt das Signal BUSPWR (Pin 8) auf +3,3 V, jetzt ist der

USB-Anschluss die Stromquelle. Eine Brücke auf JP8 signalisiert, dass ein Netzadapter den Strom liefert.

Achtung: Auf JP8 muss eine Brücke gesteckt sein, wenn der USB-Hub/UART-Konverter zum

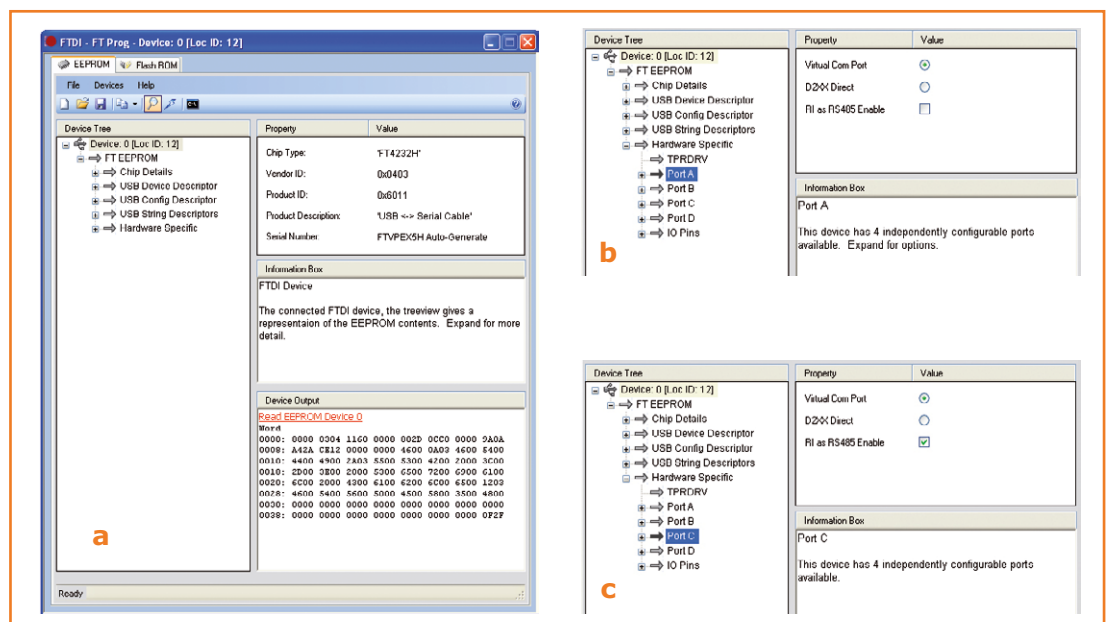


Bild 6.
Konfiguration mit dem Tool
FTPROG von FTDI.

RS422/RS485: Seriell und differentiell

Der RS422-Bus ist im Standard EIA-422-B-1994 als *Simplex*-Bus definiert. Dies bedeutet, dass gleichzeitig nur ein Bus-Teilnehmer senden darf, er darf bis zu 10 UL steuern. Eine UL ist eine *Unit Load*, eine Einheitslast. Die Signale werden über differentielle Verbindungen übertragen, was eine sichere und zuverlässige Datenübertragung über lange Distanzen gewährleistet, auch in störungsreicher Umgebung.

Baudrate gegen Reichweite

Der Sender eines RS422-Busses hat eine Reichweite von ungefähr 1200 m (4000 Fuß), die Übertragungsgeschwindigkeit kann jedoch bei weiten Distanzen sinken. Nach EIA-Standard muss sie über kurze Distanzen 10 Mbit/s erreichen (**Bild 7**).

Abschlusswiderstand

Wenn die Impedanzen von Empfänger und Kabel nicht übereinstimmen, entstehen auf der Leitung störende Reflexionen. Sie treten insbesondere bei hohen Datengeschwindigkeiten oder weiten Distanzen auf. Die Reflexionen werden gemindert, wenn das Kabel auf der Seite des Empfängers mit dem Wellenwiderstand, meistens 120 Ω , abgeschlossen wird (auf dem Board über die Jumper JP1/JP2). Reflexionen können auch mit anderen Methoden auf ein verträgliches Maß herabgedrückt werden.

RS485-Bus

Der RS485-Bus, skizziert in **Bild 8**, ist mit der RS422-Schnittstelle eng verwandt, bietet jedoch mehr. Die Anzahl der zulässigen UL beträgt 32, außerdem werden *Multi-Point*-Verbindungen unterstützt, so dass ein Sender mit mehreren Empfängern gleichzeitig kommunizieren kann.

In Bild 8 ist die Grundkonfiguration des RS485-Busses dargestellt, bei der die Daten halbduplex übertragen werden (zwei Adern für Daten). Im Halbduplex-Betrieb müssen die Empfänger erkennen, welche Daten an sie adressiert sind. Jedem Empfänger muss eine einmalig vergebene Adresse zugeordnet sein. Es ist auch möglich, Daten gleichzeitig an alle Empfänger zu versenden (*Broadcast*). Für diese Übertragungsart wird eine spezielle Adresse reserviert, beispielsweise die Adresse 255. Normalerweise ist die Empfängeradresse Bestandteil des Kopfes eines Datenpakets. Die Empfänger dürfen nicht antworten, solange sie nicht eine ausdrückliche Erlaubnis erhalten. Das Verfahren, mit dem ein solches Datennetz arbeitet, heißt *Master-Slave*-Betrieb. Der Master sendet, die Slaves müssen gewissermaßen zuhören. Nur wenn der Master einen Slave auffordert, darf nur dieser Slave antworten. Sendet der Master Daten gleichzeitig an alle Slaves, darf kein Slave antworten. Ein verbreiteter Irrtum ist die Annahme, dass der RS485-Bus mit nur zwei Adern auskommt. Es sind drei: Data+, Data- und Masse!

Unit Load

Nach EIA-Standard kann beim RS485-Bus ein Master (Sender) mit 32 Empfängern verbunden sein. Der Begriff *Unit Load* bezieht sich auf den Eingangswiderstand der Empfänger, er beträgt normgemäß 12 k Ω . Inzwischen sind Konverter auf dem Markt, die den Bus mit nur 1/8 UL belasten. Damit können theoretisch bis zu $8 \cdot 32 = 256$ Teilnehmer am gleichen Bus liegen.

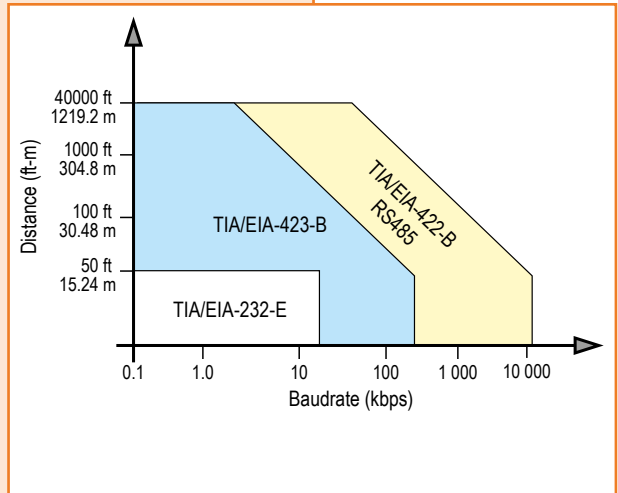


Bild 7.
Baudrate gegen Reichweite
bei RS422/RS485-
Verbindungen.

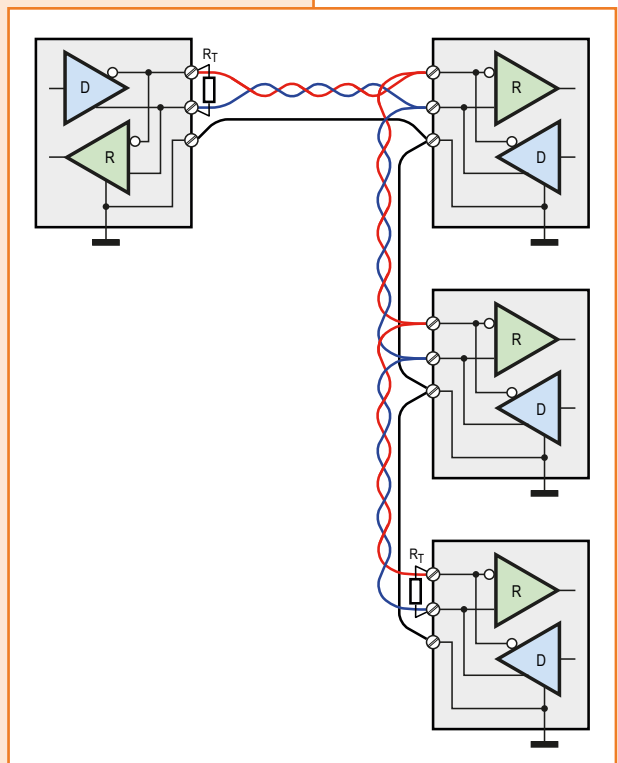
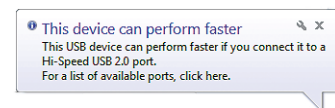
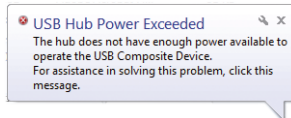


Bild 8.
Charakteristische Topologie
eines RS485-Busses.



Eine Leidenschaft

Es war vor rund 25 Jahren in den Sommerferien. Der Regen wollte nicht aufhören, und um die Zeit zu überbrücken, kaufte ich an einem nahegelegenen Kiosk mehr oder weniger zufällig die Sommer-Doppelausgabe von Elektor. Seitdem hat mich die Elektronik nicht mehr losgelassen. In meinem Berufsleben bin ich Mitarbeiter eines renommierten Ingenieurbüros, meine Freizeit gehört fast ausschließlich der praxisorientierten Elektronik und Informatik. Kaum ein Tag vergeht, an dem ich nicht Projekte entwickle oder mit ihrer Realisierung beschäftigt bin. Meiner Frau und meinen Töchtern sage ich Dank, dass sie mich hier gewähren lassen. Danke sage ich auch Pascal aus Tour für seine wertvolle, unermüdliche Unterstützung.



ersten Mal in Betrieb genommen wird. Dadurch wird verhindert, dass eine Meldung über eine zu hohe Stromentnahme erscheint. Die Meldung würde das Installieren der Treiber vorzeitig beenden. Wenn die Treiber installiert sind, kann die Brücke von JP8 entfernt werden (oder sie verbleibt, falls ein Netzadapter den Strom liefert). Mit dem Tool FTPROG lässt sich der USB-Hub/UART-Konverter für beide serielle Protokolle konfigurieren. Dieses Tool sowie die USB-Treiber für diverse Betriebssysteme können von der Website des Herstellers FTDI [1] heruntergeladen werden. Starten Sie nach der Treiber-Installation das Tool FTPROG. Nach einem Mausklick auf das Symbol

„Lupe“ sucht FTPROG den Konverter. Auf dem Bildschirm erscheint das in **Bild 6a** dargestellte Fenster. Wenn FTPROG den Konverter gefunden hat, klicken Sie, wie in **Bild 6b** gezeigt, unter *Device Tree* auf *Hardware Specific*. Port A und Port B bleiben unverändert. Bei Port C und Port D müssen Sie, wie in **Bild 6c** gezeigt, das Kästchen bei *RI as RS485 Enable* anhängen (kontrollieren Sie bitte auch, ob die Checkbox *TX enable* angeklickt ist). Anschließend speichern Sie die Konfiguration im EEPROM, indem Sie auf das Symbol „Blitz“ klicken.

(140033)gd

FTDI und Android

Unmittelbar vor Redaktionsschluss wurde der USB-Hub/UART-Konverter mit einigen Android-Tablet-Computern getestet. An den Tablets, deren USB-Ports als Hosts konfiguriert waren, ist der USB-Hub zusammen mit einem USB-Stick funktionsfähig. Noch wichtiger wäre ein Test der RS232/RS485-Ports gewesen, doch dazu müsste eine Kommunikation zwischen dem FT4232 von FTDI und dem Betriebssystem Android stattfinden, wie ein Videoclip auf Youtube [6] zeigt. Dieser Link verweist auf eine Dokumentation, die für die weitere Entwicklung des Konverters nützlich sein kann.

Weblinks

- [1] FTDI Treiber und Tool FTPROG: www.ftdichip.com
- [2] www.elektor-magazine.de/140033
- [3] Frontplatten-Hersteller: www.schaeffer-ag.de
- [4] Application Note AN-1031 TIA/EIA-422-B Overview: www.ti.com/lit/an/snla044b/snla044b.pdf

- [5] RS422- und RS485-Anwendungen (eBook): www.bb-elec.com/Learning-Center/All-White-Papers/Serial/RS-422-and-RS-485-Applications-eBook.aspx
- [6] FTDI und Android: www.youtube.com/watch?v=QSR7IAAWL1c



eSTORE
Beta LAYOUT

Entwickeln, Bestücken, Löten



€ 444,00*

**Jubiläums-
Reflow-Kit V3**

** gegenüber Kauf der Einzelkomponenten

Raspberry Pi B+



€ 39,90*

**ERSA Lötstation
i-CON NANO**



€ 198,00*

**i-Solder-Pot
Selektiv Löttiegel**



€ 203,00*

**30 MHz, 2 CH Digital-
Speicheroszilloskop**



€ 470,00*

* inkl. MwSt. und zzgl. Versandkosten

**über 50 €
sparen! ****

www.beta-eSTORE.com

**Besuchen
Sie uns:
electronica
2014**

**Halle A2, Stand 478
Messe München**

25 Jahre Beta

LAYOUT
create : electronics

eSTORE® ist eine eingetragene Marke der Beta LAYOUT GmbH



2-tägiges Seminar USB-Treiber für Mikrocontroller

Jedes Gerät – unabhängig ob es für die Industrie oder den Endverbraucher ist – benötigt meist eine Schnittstelle zum Konfigurieren oder Bedienen der Anwendung. Nach vielen Jahren hat USB endgültig die RS232-Schnittstelle abgelöst.

Lange konnte man um USB einen großen Bogen machen, in dem man eine USB-RS232-Bridge für den PC verwendete. Die virtuelle COM-Schnittstelle bietet aber oft nicht die notwendigen Eigenschaften, die von der Anwendung aus gewünscht sind (Problem 1 Millisekunden Frame, eigene Benutzererkennung usw.).

Moderne Mikrocontroller bieten seit einiger Zeit USB-Schnittstellen direkt an. Diese kann mit dem notwendigen Know-how einfach in die eigene Anwendung integriert werden. Viele Beispiele existieren, die man mit einfachen Handgriffen anpassen kann.

In diesem Seminar wird gezeigt, wie man sich basierend auf den integrierten USB-Schnittstellen der Mikrocontroller eine Kommunikation mit Betriebssystemen (Windows und Linux) über Treiber und Bibliotheken aufbauen kann. Das Seminar ist mit vielen praktischen Beispielen abgerundet, um verschiedene Möglichkeiten für eine Anbindung zu sehen. Kennt man erst einmal diese und hat dazu ein paar passende Beispiele, kann jeder USB so verwendet werden, wie es die Anwendung erfordert.

Veranstaltungsort/-termin: Augsburg, 10. + 11. Dezember 2014

Referent: Dipl.-Inf. (FH) Benedikt Sauter

Teilnahmegebühr: 749,00 Euro (inkl. MwSt.)

Im Preis sind Mittagessen, Seminarunterlagen, Dokumentation und Teilnahmezertifikat inbegriffen.

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

Elektor-Abomitglieder erhalten 5% Rabatt auf den Workshop-Preis!

Elektor-Verlag GmbH | Süsterfeldstr. 25 | 52072 Aachen

Tel. +49 241 88 909-16

www.elektor.de

j.grotenrath@elektor.de

C-Module

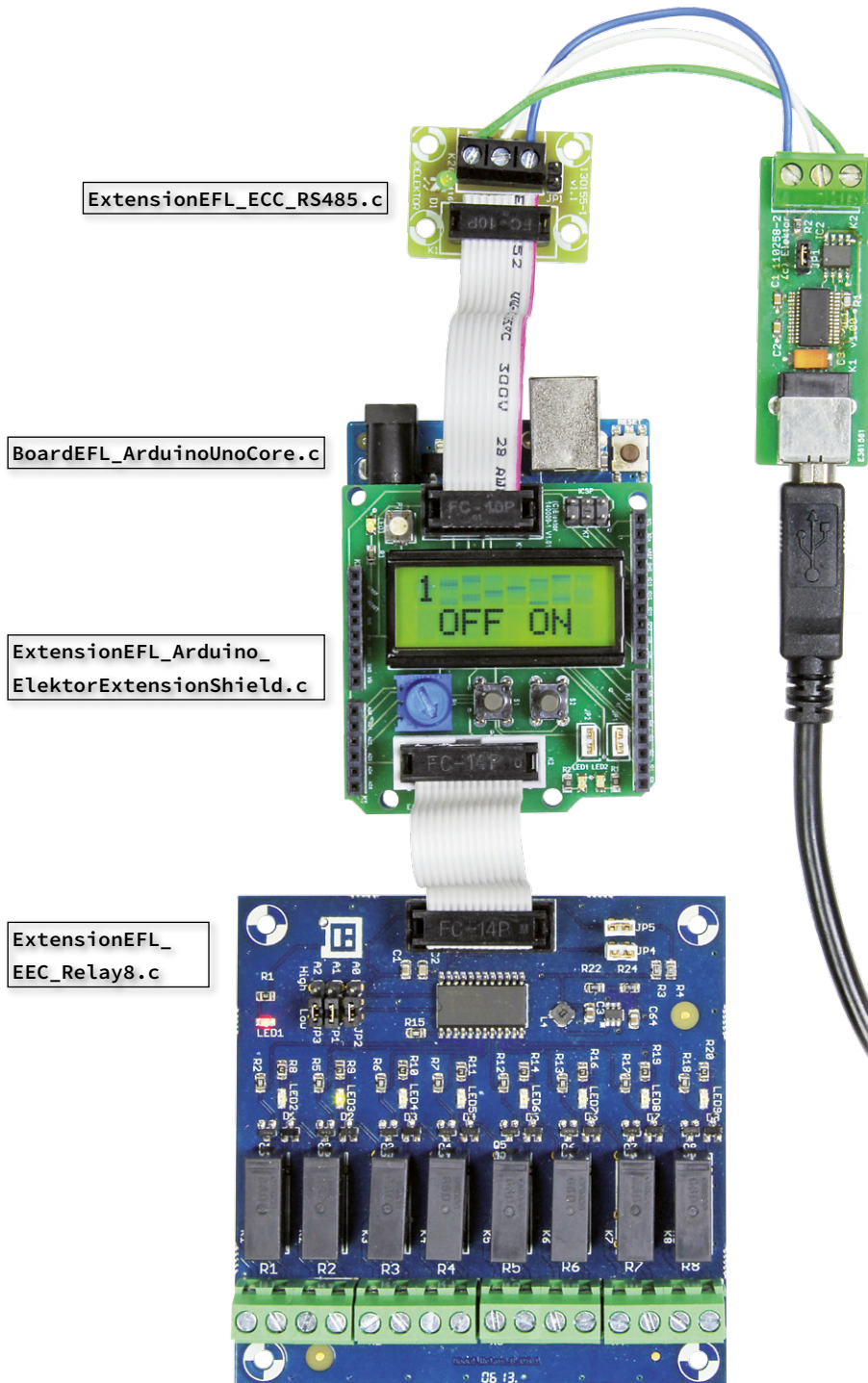
Software für Elektor Extension-Shield, Relais-Board und mehr

Von **Jens Nickel**

Mit modularer Hardware kommt man schnell zu einem passenden Prototypen, weil man fertig entwickelte Elemente nur noch zusammenfügen muss. Die Programmiersprache C bietet sich dafür an, dieses Prinzip auf Software zu übertragen. Für unser Elektor Extension-Shield und drei Erweiterungsboards stellen wir hier passende Softwaremodule und Demo-Anwendungen vor.

In der Sommer-Doppelausgabe haben wir ein kompaktes Aufsteckboard für den Arduino Uno vorgestellt, das ein Display, zwei User-LEDs, zwei Taster und ein Poti enthält [1]. Einsteiger können damit gleich loslegen und erste Schritte in der Programmierung von Mikrocontrollern unternehmen, zum Beispiel digitale Ausgänge setzen (LEDs) und digitale Eingänge einlesen (Taster). Fortgeschrittene finden auf dem Board noch zwei Erweiterungsstecker. Auf den 10-poligen Embedded Communication Connector (ECC) sind die UART-Signale TX/RX und zwei GPIO-Pins herausgeführt. Über ein Flachkabel lässt sich zum Beispiel ein RS485-Modul anschließen, so dass man Bytes auch über lange Leitungen zuverlässig versenden und empfangen kann. Gerade entwickelt wird ein NFC-Gateway, das mit einfachen ASCII-Kommandos das Auslesen und Beschreiben von NFC-Karten und eine Kommunikation mit einem NFC-fähigen Smartphone erlaubt. Weitere ECC-Module sind in Planung.

Der 14-polige Embedded Extension Connector (EEC) ist auch unter dem Namen GnuBlin-Connector bekannt, weil sich hier die GnuBlin-Module von Benedikt Sauter und seinem Team anschließen lassen. Von Benedikt stammt auch die Spezifikation



des Steckers; beim Elektor Extension-Shield werden allerdings nur vier Pins genutzt, nämlich 3,3 V, GND und die beiden I2C-Leitungen. Die meisten der GnuBlin-Boards [2] verwenden aber nur die I2C-Pins, zum Beispiel das Portexpander-Board, das Temperatursensor-Board, ein Board mit acht Relais [3] und die vom Elektor-Labor entwickelte ADC-Platine aus der Septemberausgabe [4].

Software-Module

Im in dieser Ausgabe beendeten Bascom-Kurs hat Burkhard Kainka einige interessante Anwendungen für das Elektor Extension-Shield entwickelt; weitere sollen in lockerer Folge in Elektor vorgestellt werden.

Was bisher fehlte, war eine Software-Unterstützung in C. C eignet sich besonders gut für modulare Softwareprojekte. Einmal entwickelte und getestete Software-Module (entweder selbst oder von Dritten erstellt) lassen sich immer wieder in eigenen Projekten verwenden. Das verkürzt die Entwicklungszeit enorm, im Extremfall kommt man innerhalb von mehreren Minuten zu einer größeren, funktionierenden Anwendung. Und wenn sich die Anforderungen an das Projekt ändern, also beispielsweise zusätzlich eine RS485-Schnittstelle benötigt wird, dann lässt sich ein solches Softwareprojekt ebenso schnell passend erweitern.

Auf Seiten der Hardware verfügen wir nun schon über ein funktionierendes System aus Modulen, die man nur noch zusammenstecken muss. Auf ein Arduino-Uno-Board wird das Elektor Extension-Shield gesteckt, an das wir beispielsweise wiederum ein RS485-Modul, eine Relais-Karte oder ein ADC-Board anstecken können. Dieses Prinzip wollen wir nun auch auf die Software übertragen; für eine Anwendung mit bestimmten Anforderungen werden einfach die entsprechenden Code-Module in einem Software-Projekt zusammengefügt. Idealerweise gehört zu jedem Hardware-Modul ein Software-Modul, das unverändert verwendet werden kann, unabhängig davon, in welcher Konstellation die Hardware zusammenspielt. Ein Beispiel ist der Code für das Relais-Erweiterungsboard; er bleibt immer gleich, egal ob wir die Relais an das Elektor Extension-Shield, an das Xmega-Board oder an das Elektor Linux-Board anschließen.

Nah am Ideal

Mit der Programmiersprache C und der in Elektor [5] vorgestellten Embedded Firmware Library

Neue File-Struktur

Die EFL eignet sich nun auch für größere modulare Projekte, bei denen an ein Controllerboard mehrere Erweiterungsboards angeschlossen werden (auch verkettet). Hierfür musste die Bibliothek gegenüber dem bisherigen Stand [5] etwas erweitert und umstrukturiert werden.

- Das hierarchische System der Includes wurde entwirrt. Es werden jetzt überall nur noch Header-Files aus dem Ordner „Common“ eingebunden. Diese Files sind bei allen EFL-Projekten gleich, egal welche Boards und Controller verwendet werden.
- Hierzu wurde ein neues Headerfile im Common-Ordner angelegt (ControllerDefinesEFL.h). Es enthält nun alle Funktionsdefinitionen der Controller-API, die ja immer gleich sind. Das controllerspezifische Headerfile (ehemals ControllerEFL.h) enthält jetzt nur noch controllerspezifische Definitionen, etwa für die Register.
- Statt ControllerEFL.h/.c bzw. BoardEFL.h/.c vergeben wir nun aussagekräftige Namen für die controller- und boardspezifischen Codedateien. Das File, das die Controller-API für den ATmega328 implementiert, heißt nun ControllerEFL_ATmega328. Das File, das von der Verdrahtung auf dem Arduino-Uno-Board abstrahiert, heißt BoardEFL_ArduinoUnoCore.
- Wir können nun mehrere Extensionboards in unser Projekt einbinden, da die zugehörigen Codedateien statt „ExtensionEFL.c/.h“ alle unterschiedliche Namen tragen. Beispiel:
 - ExtensionEFL_Arduino_ElektorExtensionShield.c/.h
 - ExtensionEFL_ECC_RS485.c/.h
 - ExtensionEFL_EEC_Relay8.c/.h

Zwischen den Unterstrichen steht hier immer der Typ des Erweiterungssteckverbinders.

- Die Board-Init-Funktionen, welche die Verdrahtung auf den Boards in die Tabellen eintragen und die Peripherie-Einheiten vorbereiten, wurden ebenfalls spezifischer benannt, zum Beispiel ExtensionEFL_Arduino_ElektorExtensionShield_Init(0) statt ExtensionEFL_Init(). Um das einheitlich zu machen, heißt auch die Controller-Init-Funktion jetzt zum Beispiel ControllerEFL_ATmega328_Init() statt ControllerEFL_Init().

Eine Initialisierung des Controllers und aller Boards kann nun zum Beispiel so aussehen:

```
ControllerEFL_ATmega328_Init();
BoardEFL_ArduinoUnoCore_Init();
ExtensionEFL_Arduino_ElektorExtensionShield_Init(0);
ExtensionEFL_ECC_RS485_Init(2);
ExtensionEFL_EEC_Relay8_Init(3);
```

Den Init-Funktionen der Erweiterungsboards wird die fortlaufende Blocknummer des Erweiterungssteckers mitgegeben, an die sie angeschlossen sind. So wird die Verdrahtung (von den Controllerpins bis zur äußersten Peripherie) korrekt in den EFL-internen Tabellen abgebildet. Es ist nun auch das Verketteten von Erweiterungsboards möglich wie im vorliegenden Fall (siehe hierzu auch den Fließtext und Bild 1).

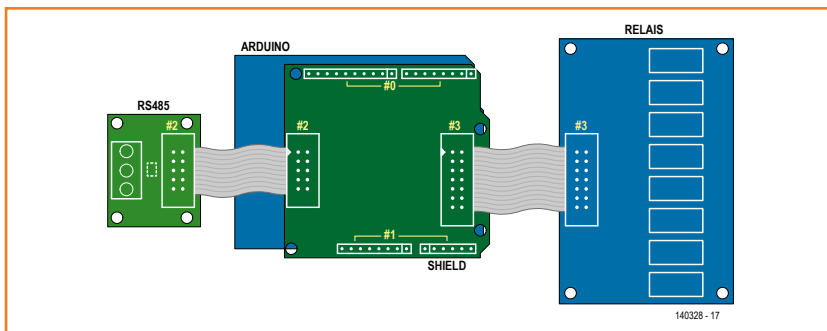


Bild 1.
Verkettung von Erweiterungsböden: Das Extension-Shield wird auf zwei Buchsenleisten des Arduino Uno gesetzt (#0 = Digital-Pins, #1 = Analog-Pins). Das Shield stellt wiederum die Steckverbinder #2 (ECC) und #3 (EEC) zur Verfügung.

(EFL) können wir diesem Ideal schon sehr, sehr nahe kommen, wie wir im Folgenden zeigen werden. Bisher konnte man die EFL nur für ein Duo aus Controllerboard und Extensionboard nutzen. Damit wir in einem Projekt nun mehrere Erweiterungsboards verwenden können, musste die Bibliothek etwas erweitert und umstrukturiert werden. Wir wollen die Praktiker unter unseren Lesern aber nicht mit Details langweilen, und haben die Erklärung in Textkästen ausgelagert. Hier nur die wichtigsten Änderungen in Kürze:

- Bisher enthielt ein Filepaar namens ControllerEFL.h/.c den controllerspezifischen Quellcode, dabei werden zur Ansteuerung der Ein- und Ausgänge jedes Controllers einheitliche Funktionen zur Verfügung gestellt wie zum Beispiel `IO_SetPinLevel(...)`. Ein Filepaar, das immer `BoardEFL.h/.c` hieß, bildete die Verdrahtung auf dem Board ab (die der Anwendungsentwickler dann nicht mehr kennen muss). Statt der uniformen Bezeichnungen tragen diese Files nun aussagekräftige Namen. Das File, das für den ATmega328 zuständig ist, heißt

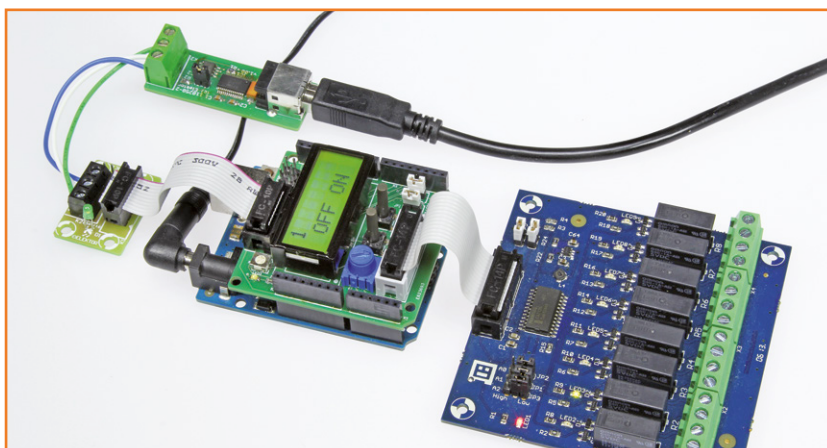
nun `ControllerEFL_ATmega328`. Das File, das zum Arduino-Uno-Board gehört, heißt `BoardEFL_ArduinoUnoCore`.

- Wir können nun mehrere Extensionboards in unserem Projekt nutzen, da auch die entsprechenden Codefiles alle verschiedene Namen tragen, zum Beispiel:
 - `ExtensionEFL_Arduino_ElektorExtensionShield.c/.h`
 - `ExtensionEFL_ECC_RS485.c/.h`
 - `ExtensionEFL_EEC_Relay8.c/.h`
- Die Init-Funktionen, die man beim Start aufrufen muss, wurden ebenfalls spezifischer benannt. Hier die Initialisierung des Controllers und aller Boards für das nachfolgend beschriebene erste Anwendungsbeispiel:

```
ControllerEFL_ATmega328_Init();
BoardEFL_ArduinoUnoCore_Init();
ExtensionEFL_Arduino_ElektorExtensionShield_Init(0);
ExtensionEFL_ECC_RS485_Init(2);
ExtensionEFL_EEC_Relay8_Init(3);
```

Den Init-Funktionen der Erweiterungsboards wird die Nummer des ersten Erweiterungssteckers mitgegeben, an den sie angeschlossen sind; die Nummer wird während der Initialisierung der Boards hochgezählt (**Bild 1**). Das Extension-Shield wird auf den Arduino Uno gesetzt, der zwei Erweiterungsstecker definiert, diese bekommen die Nummer #0 (Digital-Pins) und #1 (Analog-Pins). Das Extension-Shield ist an #0 und #1 angeschlossen und besitzt wieder zwei Erweiterungssteckverbinder, nämlich #2 (ECC) und #3 (EEC). An den ersteren wird das RS485-Modul angeschlossen, an den zweiten das Relais-Modul.

Bild 2.
Anwendung 1: Acht Relais können lokal über ein User-Interface geschaltet und über RS485 ferngesteuert werden.



(Fern-)Steuerung von Relais

Wir haben drei Demo-Projekte für Atmel Studio 6 vorbereitet, diese findet man im Download zu diesem Artikel [6]. Bereits in der Entwicklung ist übrigens ein Konfigurator für den PC, der automatisch die nötigen Files für ein EFL-Projekt in Atmel Studio zusammenstellt und einbindet. Man muss diesem nur noch mitteilen, welche Boards man im Projekt verwenden möchte.

Beginnen wir mit dem ersten Projekt, einer kleinen Steuerungsanwendung. Acht Relais können lokal über ein User-Interface geschaltet und auch von einem PC aus per Terminalprogramm ferngesteuert werden. Auf einen Arduino Uno

wird ein Elektor Extension-Shield gesteckt; die GnuBlin-Platine mit acht Relais schließen wir dort über ein Flachkabel an (**Bild 2**). Optional ist das RS485-ECC-Modul, das ebenfalls am Shield angeschlossen wird; über RS485 (zwei Leitungen plus Masse) geht es dann weiter zu einem RS485/USB-Konverter und den PC. Die Anwendung funktioniert aber auch, wenn man den Arduino Uno direkt über USB mit einem PC verbindet.

Die Software findet man im Download-Paket im Ordner ElektorShieldRelay. Nach einem Klick auf ElektorShieldRelay.atsIn öffnet sich das Projekt in Atmel Studio (**Bild 3**). Im Solution Explorer rechts sieht man die eingebundenen Files.

Unbedingt ansehen sollte man sich einmal den Ordner Hardware. Neben den erwähnten Dateien für den Controller, das Controllerboard und die Erweiterungsboards findet man hier auch die neuen BlockEFL-Files mit den Low-Level-Funktionen für den RS485-Treiber, die digitalen Aus- und Eingänge und das Display (siehe Kasten „BlockEFL-Files“). Fortgeschrittene sollten sich einmal den Code im File ExtensionEFL_Arduino_ElektorExtensionShield.c ansehen (**Bild 3**). Der Code trägt die Peripherieblöcke, die sich auf dem Shield befinden, in die EFL-interne Block-Tabelle ein (**Bild 4**). Die Buttons auf dem Board sind ab jetzt anzusprechen über die Blocknummer #0, die LEDs über die Blocknummer #1, denn es befindet sich bereits eine LED direkt auf dem Arduino-Board, die über die Blocknummer #0 angesteuert wird. Für das Poti wird ein ADC-Block #0 angelegt; danach folgt das Display mit der Nummer #0. Zum Schluss wird die Verdrahtung zu den ECC- und EEC-Steckverbindern abgebildet, indem zwei neue Blöcke für Steckverbinder angelegt werden (#2, #3). Die Einträge werden dann wiederum vom Code der ECC- und EEC-Module genutzt. Letztendlich kann jede Peripherieeinheit in den Tabellen „nachschaun“, mit welchem Controllerpin sie verdrahtet ist.

Sehen wir uns nun das Hauptprogramm im File ElektorShieldRelay.c an. Die Routine ApplicationSetup() enthält alle Initialisierungen, vom Controller über die Boards bis zu den Bibliotheken. Das alles kann man unverändert lassen, so lange man mit dem gleichen Hardware-Setup arbeitet.

Kurzer Code

Die eigentliche Anwendung sieht man in **Listing 1**. In der Funktion ApplicationLoop() ste-

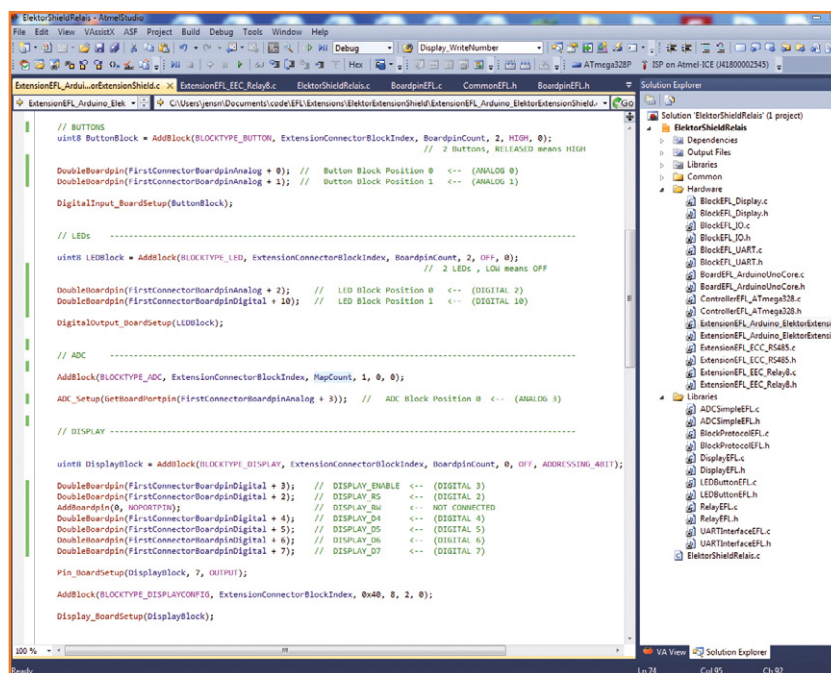
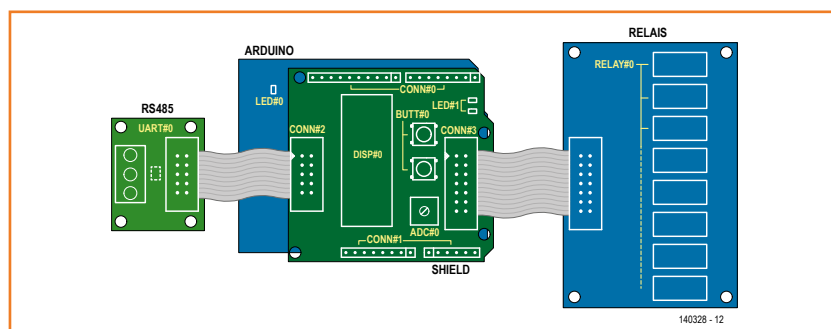


Bild 3. Die Relais-Steuerung in Atmel Studio 6. Der Code zeigt die Initialisierung des Extension-Shields.

hen alle Befehle, die ständig wiederholt werden sollen. Zuerst werden die Taster abgefragt, dann wird auf Zeichenketten im BlockProtocol-Format [7] gelauscht, die über den UART empfangen wurden. Die Funktion ADCSimple_GetRawValue(0, 0) gibt den Wert des ADC-Eingangs 0 im ADC-Block #0 zurück, hier ist ja das Poti angeschlossen. Der Wert kann 0..1023 betragen (10 bit); wir reduzieren diesen per Rechtsschift um 7 bit, um eine Zahl zwischen 0 und 7 zu erhalten. So können wir eines der acht Relais über die Stellung des Potis auswählen. Der ausgewählte Wert wird in der Zeile 0 des Displays #0 dargestellt.

In der Funktion ButtonEventCallback(...) steht Code, der beim Drücken auf einen Taster ausgeführt wird. Welcher der beiden Taster gedrückt worden ist, kann man anhand der Variablen But-

Bild 4. Die gesamte Peripherie lässt sich über die Blocknummern bequem und einheitlich ansprechen.



BlockEFL-Files

Die BoardEFL- und ExtensionEFL-Files enthielten bisher noch die sogenannten Low-Level/Block-Funktionen, zum Beispiel die Funktion

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

welche von der Verdrahtung zwischen Controller und Display abstrahiert (4 bit beziehungsweise SPI). Eine Display-Library wie DisplayEFL muss dann nur dafür sorgen, dass die richtigen Bytes zum Display geschickt werden, der Code dort ist unabhängig davon, auf welchem Weg die Bytes zum Display kommen. Diese Block-Funktion muss genau ein Mal im Projekt definiert sein, wenn sich auf irgendeinem Board ein Display befindet. Sinnvollerweise legen wir daher für die Low-Level-/Block-Funktionen, die ein Display benötigt, ein eigenes Filepaar im Hardware-Ordner an. Benutzt man irgendwo ein Display in einem Projekt, muss man BlockEFL_Display.h/.c zusätzlich einbinden. Ein anderes BlockEFL-File ist für die RS485-Funktionen zuständig. Die Block-Funktionen für digitale Aus- und Eingänge, wie zum Beispiel SwitchDigitalOutput(uint8 BlockIndex, uint8 Position, uint8 ON_OFF) wurden ebenfalls aus dem BoardEFL-File gezogen und in ein neues File BlockEFL_IO verlagert.

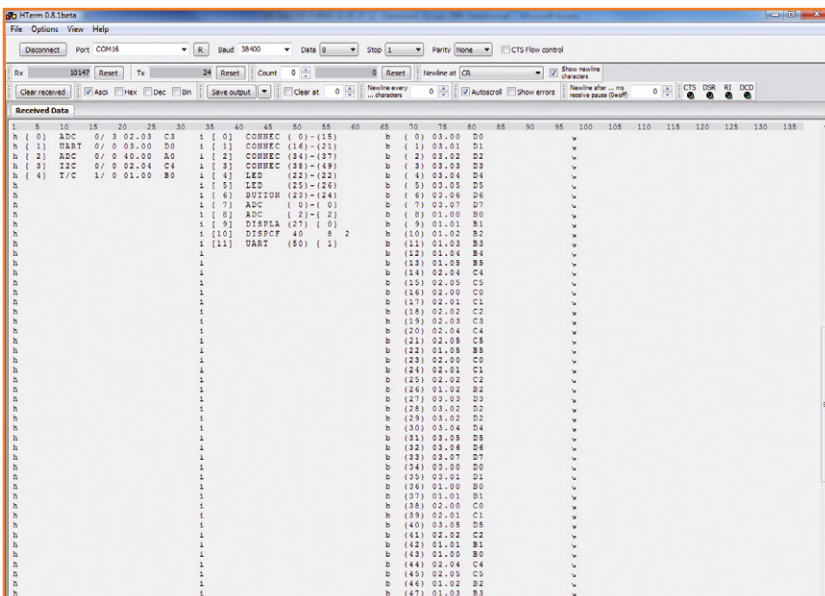
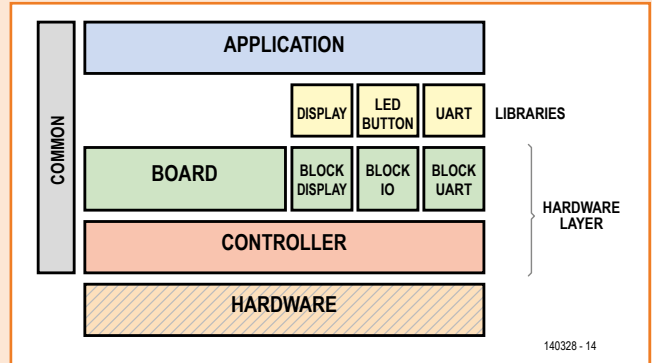
Die Boardfiles enthalten nun im Normalfall nur noch die Init-Funktionen, welche die Verdrahtung festlegen und die integrierten Peripherie-Blocks wie zum Beispiel das Display bereit machen. Hierfür wird dort die Funktion

```
void Display_BoardSetup(uint8 DisplayBlockIndex);
```

aufgerufen, die sich jetzt in BlockEFL_Display.h/.c befindet.

Insgesamt hat man es nun noch mit etwas mehr Files zu tun, die Modularität der EFL wird aber nochmals verbessert.

Bereits in der Entwicklung ist ein Projekt-Konfigurator, der automatisch die nötigen Files für ein Projekt zusammenstellt und einbindet.



tonPosition unterscheiden (0 oder 1). Falls wir den linken Button drücken (0), dann schalten wir das ausgewählte Relais aus, falls wir den rechten Button (1) drücken, dann schalten wir das Relais an. Da die Funktion SwitchRelay(...) gerade eine 0 = OFF oder eine 1 = ON als dritten Parameter verlangt, fällt der Code schön kurz und knackig aus. Infos zu dieser und den anderen EFL-Funktionen findet man in der Doxygen-Dokumentation (Klick auf Index.htm im Download). Über das BlockProtocol [7] lassen sich die Relais von einem Terminalprogramm aus fernsteuern. Das Kommando `R 0 2 + <CR>`

Bild 5. Die EFL-Tabellen kann man sich mit einem Terminalprogramm anzeigen lassen. In der Mitte sieht man alle angelegten Blocks.

schaltet im Relaisblock #0 das dritte Relais an (es wird wie immer nullbasiert gezählt 0, 1, 2...). R 0 2 - <CR> schaltet es wieder aus. Mit L 0 0 + <CR> können wir die LED anschalten, die sich direkt auf dem Arduino befindet. L 1 0 + <CR> und L 1 1 + <CR> sind die entsprechenden Befehle für die LEDs auf dem Extension-Shield. Probieren Sie auch einmal das Kommando x <CR>, Sie bekommen dann den Inhalt der EFL-Tabellen auf den Schirm. In der Mitte sieht man alle angelegten Blocks (**Bild 5**).

Genaue Messungen

Für unsere zweite Anwendung stecken wir das Relais-Board ab und stattdessen die 16-bit-ADC-Platine an, die wir in der Septemberausgabe vorgestellt haben [4]. Wie auch im dortigen Artikel zu lesen, verbinden wir den Ausgang A3 der unteren Arduino-Buchsenleiste (auf dem Shield gedoppelt) über einen fliegenden Draht mit dem Eingang AIN0 des ADC-Boards (**Bild 6**). Nun können wir die Stellung des Potis auch mit dem genauen externen ADC digitalisieren.

Ein Doppelklick auf `ElektorShieldADC.atsln` fördert den Quellcode zutage. Wie der Solution Explorer zeigt, befinden sich statt der zwei Files `ExtensionEFL_EEC_Relay8.c/.h` nun die zwei Files `ExtensionEFL_EEC_ADCModule16bit.c/.h` im Projekt. Außerdem sind noch die Files `ADC_ADS1x1xEFL.c` und `BlockEFL_DeviceRegister16.c` hinzugekommen. `BlockEFL_DeviceRegister16.c` im Ordner `Hardware` enthält Low-Level-Funktionen zum Ansprechen eines I2C-Chips, der 16-bit-Register aufweist. `ADC_ADS1x1xEFL.c` sorgt für die richtige Zusammenstellung der Bytes, welche in diese Register geschrieben werden (die Bibliothek meines Kollegen Clemens Valens [4] wurde hier an die EFL angepasst).

Der Anwendungsentwickler muss sich für all das nicht interessieren. Er muss nur wissen, dass nach der Initialisierung der Boards ein weiterer ADC-Block mit der Nummer #1 angelegt wurde (**Bild 7**). Auf den externen AD-Wandler kann man nun genauso zugreifen wie auf einen internen ADC des Controllers.

Die Anwendungs-Bibliothek `ADCSimpleEFL.c` stellt zum Beispiel die Funktion `ADCSimple_GetMillivoltValue(..)` bereit, mit der man die Spannung in Millivolt an einem Analogeingang eines bestimmten ADC-Blocks ermitteln kann.

Listing 1. Relais (fern)schalten.

```
uint8 RelayPosition;

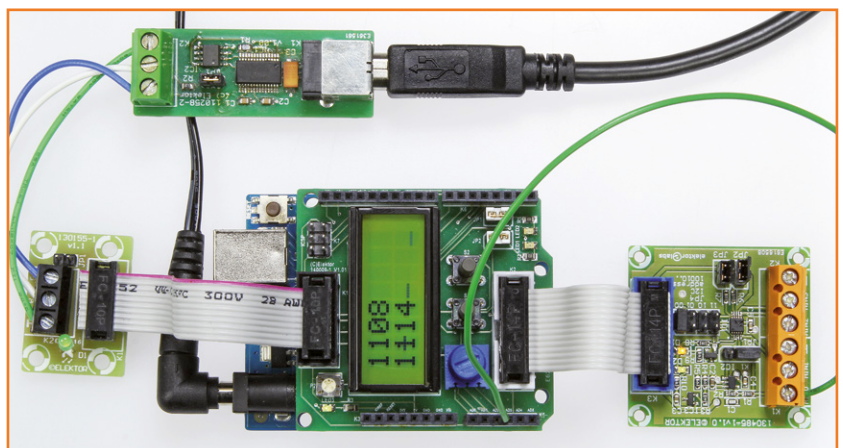
void ApplicationLoop()
{
    ButtonPoll(0);
    BlockProtocol_Engine();

    RelayPosition = ADCSimple_GetRawValue(0, 0) >> 7;
    Display_WriteNumber(0, 0, RelayPosition);
}

void ButtonEventCallback(uint8 BlockType, uint8
ButtonBlockNumber, uint8 ButtonPosition, uint8 Event)
{
    if (Event == EVENT_BUTTON_PRESSED)
    {
        ToggleLED(1, 0);
        SwitchRelay(0, RelayPosition, ButtonPosition);
    }
}
```

Der eigentliche Anwendungscode steht im File `ElektorShieldADC.c`, die wesentliche Funktion haben wir in **Listing 2** abgedruckt. Man müsste nun eigentlich schon erkennen können, was die Anwendung macht: Die Spannung am Arduino-Pin A3 wird sowohl vom internen als auch vom externen ADC digitalisiert und auf dem Display dargestellt. Prüfen Sie es ruhig auch mal mit einem guten Multimeter nach: der externe ADC arbeitet wirklich genau.

Bild 6.
Anwendung 2: Externer 16-bit-ADC am Arduino Uno. Über die Leitung wird die Spannung am Poti weitergegeben, so dass sich die Ergebnisse des internen und externen ADCs vergleichen lassen.



Virtualisierung

Portexpander und externe AD-Wandler sind zwei Beispiele für Chips, welche die Fähigkeiten eines Controllers erweitern. Häufig werden diese Chips über einen I2C-Bus oder eine andere serielle Schnittstelle angesprochen. Eine modulare Prototyping-Bibliothek sollte soweit es geht von der verwendeten Hardware abstrahieren. Für den Entwickler einer Anwendung (oder einer Bibliothek, welche Peripherie ansteuert) sollte es keine Rolle spielen, wie die Leitungen auf dem Board laufen, also an welche Controllerpins die Peripherie angeschlossen ist. Idealerweise sollte er auch nicht wissen müssen, ob ein digitaler Ein- oder Ausgang an einen echten Controllerpin oder an einen Porterweiterungschip angeschlossen ist. Wir können das Problem lösen, indem wir dem Controller zusätzlich zu seinen Ports 0, 1, ... usw. „virtuelle“ Ports zuweisen, welche mit der Nummer 0x40 = 64 beginnen, um sie von echten Ports unterscheiden zu können.

Falls man beispielsweise ein Relais anschalten möchte, das an einer Porterweiterung hängt, dann wird der Zugriff auf das Relais von der RelayEFL-Library ganz normal an die Funktion `SwitchDigitalOutput(...)` weitergegeben, die sich jetzt in der Datei `BlockEFL_IO.c` befindet. Diese Funktion schaut in den EFL-Tabellen nach, zu welchem Controllerport und -pin das Relais gehört und ruft die Funktion `IO_SetPinLevel(...)` im Controllerfile auf. In den Tabellen wurde für das Relais ein Port 0x50 hinterlegt. Jetzt wären wir normalerweise am Ende, denn einen Port 0x50 kennt der Controller nicht. Doch wurde dem Controller-Code für solche Fälle bei der Initialisierung des Relais-Boards eine spezielle Funktion des Relais-Board-Codes übermittelt, die er in diesen Fällen anrufen soll. Genau diese Funktion sendet dann die entsprechenden I2C-Befehle

an die auf dem Board befindliche Porterweiterung, um den Ausgangs-Pin zu setzen. Das Konzept der „Virtualisierung“ wurde nun auf analoge Eingänge ausgedehnt. Der externe ADC befindet sich in unserem Fall auf dem EEC/Gnublin-ADC-Erweiterungsboard, also muss der Code im File `ExtensionEFL_EEC_ADCModule16bit.c` die Ansteuerung seines ADCs übernehmen. Bei der Initialisierung des Boards mit der Funktion `ExtensionEFL_EEC_ADCModule16bit_Init(...)` wird der externe ADC wie ein interner ADC in die EFL-Tabellen eingetragen (in unserem Fall mit der Blocknummer #1), dort wird aber auf einen virtuellen Port 0x40 verwiesen. Darüber hinaus wird das I2C-Interface vorbereitet. Der Controller-Code bekommt außerdem die Funktion „`Virtual_ADC_GetValue(...)`“ mitgeteilt.

Über die Funktion `ADCSimple_GetRawValue(uint8 ADCBlockNumber, uint8 ADCPosition)` hat der Anwender nun Zugriff auf einen ADC-Pin in einem bestimmten ADC-Block, unabhängig von der Verbindung mit dem Controller. Die Funktion ruft direkt die Controllerfunktion `ADC_GetValue(...)` auf. Der Controller schaut in den EFL-Tabellen nach und erkennt anhand der hohen Portnummer, dass kein interner ADC gemeint ist, sondern stattdessen die Funktion `Virtual_ADC_GetValue(...)` aufzurufen ist, welche sich im Codefile des Erweiterungsboards befindet. Hier erfolgt dann der Zugriff auf den externen ADC über I2C. Neben der Funktion `ADC_GetValue(...)` wird in unserer Anwendung auch die Funktion `ADC_GetParameter(...)` virtualisiert, mit der die Auflösung und der Spannungsbereich eines ADCs abgefragt werden kann. Damit lassen sich dann ADC-Werte in Millivolt berechnen.

Auch bei dieser Anwendung erhalten wir wieder Fernzugriff über das BlockProtocol. Das Kommando `A 0 0 # <CR>` bewirkt, dass uns der Arduino den gerade gesampelten Wert des inter-

nen ADCs in Hex-Ziffern zurücksendet. `A 1 0 # <CR>` gibt den letzten Wert des externen ADCs zurück. Die Angaben sind ganz verschieden, weil es sich um Rohwerte und nicht um Angaben in Millivolt handelt. Um die BlockProtocol-Library nicht über Gebühr aufzublasen, haben wir dort auf eine Funktion zur Umrechnung von ADC-Werten in Millivolt verzichtet.

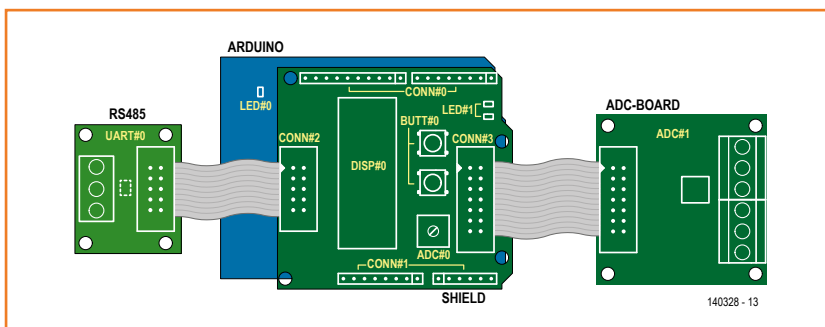


Bild 7. Interner und externer ADC lassen sich über die gleichen Funktionen ansprechen, zur Unterscheidung genügen die Blocknummern #0 und #1.

Mini-Protokoll

Doch da wir die Millivolt-Angaben in der Anwendung ja bereits ausrechnen, können wir sie nicht einfach über den UART abrufen?

Doch – wir basteln uns einfach ein eigenes Mini-Protokoll. Falls wir 0 <CR> übermitteln, dann soll der Wert des internen ADCs in Millivolt zurückkommen. Eine 1 <CR> soll bewirken, dass uns der Arduino den Wert des externen ADCs zurückschickt. Jetzt sind beide Werte direkt im Terminalprogramm vergleichbar.

Listing 3 zeigt den zugehörigen Code (Projekt ElektorShieldADCMiniProtocol.atsn im Download). ReceiveRingbuffer enthält die Adresse des Empfangs-Ringpuffers, in den die Zeichen geschrieben werden, die den Arduino über den UART (UART-Block #0) erreichen.

In der Anwendungs-Routine SendADCValueOverUART(...) wird der Millivolt-Wert in Hexziffern umgewandelt und über den UART #0 versandt.

In einer der nächsten Ausgaben werden wir den Konfigurator vorstellen, mit dem Sie selbst ein EFL-Projekt generieren können. Außerdem werden wir eine kleine Anleitung zum Schreiben eines eigenen Boardfiles geben. Bleiben Sie dran!

(140328)

Weblinks

- [1] www.elektor-magazine.de/140009
- [2] www.elektor.de/tools/gnublin
- [3] www.elektor-magazine.de/130157
- [4] www.elektor-magazine.de/130485
- [5] www.elektor-magazine.de/120668
- [6] www.elektor-magazine.de/140328
- [7] www.elektor-magazine.de/130154

Listing 2. Messung mit internem und externem ADC.

```
void ApplicationLoop()
{
    ButtonPoll(0);

    BlockProtocol_Engine();

    uint16 ADCValue1 = ADCSimple_GetMillivoltValue(0, 0);
    Display_WriteNumber(0, 0, ADCValue1);

    uint16 ADCValue2 = ADCSimple_GetMillivoltValue(1, 0);
    Display_WriteNumber(0, 1, ADCValue2);
}
```

Listing 3. Fernablesung der Werte mit einem Mini-Protokoll.

```
void ApplicationLoop()
{
    ButtonPoll(0);
    //BlockProtocol_Engine();

    uint16 ADCValue1 = ADCSimple_GetMillivoltValue(0, 0);
    Display_WriteNumber(0, 0, ADCValue1);

    uint16 ADCValue2 = ADCSimple_GetMillivoltValue(1, 0);
    Display_WriteNumber(0, 1, ADCValue2);

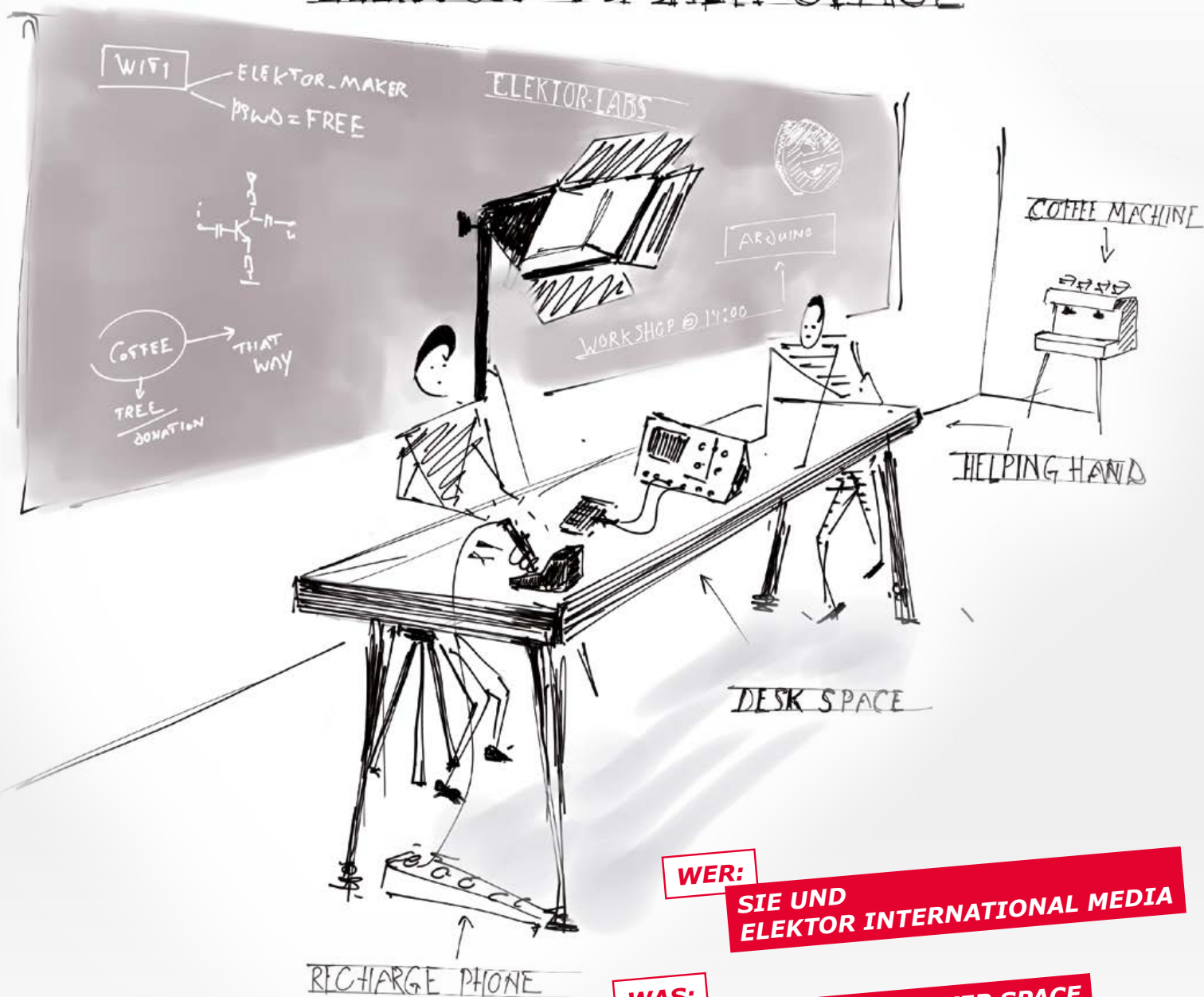
    while (Ringbuffer_IsEmpty(ReceiveRingbuffer) == FALSE)
    {
        uint8 ReceivedChar = Ringbuffer_GetByte(ReceiveRingbuffer);
        if (ReceivedChar == '1')
        {
            SendADCValueOverUART(ADCValue2);
        }
        if (ReceivedChar == '0')
        {
            SendADCValueOverUART(ADCValue1);
        }
    }
}

void SendADCValueOverUART(uint16 ADCValue)
{
    uint8 sd[3];

    sd[0] = (ADCValue & 0xFF00) >> 8;
    sd[1] = ADCValue & 0x00FF;
    sd[2] = 13;

    UARTInterface_Send(0, sd, 3);
}
```

ELEKTOR-MAKER-SPACE



WER:

**SIE UND
ELEKTOR INTERNATIONAL MEDIA**

WAS:

ELEKTOR LABS MAKER SPACE

WANN:

11.-14. NOVEMBER 2014

WO:

**ELECTRONICA, MÜNCHEN,
HALLE A6 / STAND 380**

Make It in München

Treffen Sie die Elektor-Macher auf der *electronica* 2014

Von Wisse Hettinga

Was erwarten Sie von einer solch gradiosen Show wie der *electronica* in München? Makellos getrimmte Stände und Pavillons, geschniiegelte Menschen mit attraktiv gestalteten Produkten hinter Glas, exquisit gereinigte Hallen mit Kellnern in den Restaurants?

Verstehen Sie mich nicht falsch, ich bin nicht gegen dem Anlass angemessene Kleidung - aber dann und wann muss man auch mal runterkommen, sich ein wenig entspannen, einfach gehen lassen, rumhängen, chillen, sich mit etwas „richtiger“ Arbeit die Hände schmutzig machen.

Dies fasst zusammen, was das Elektor-Labor auf der Messe präsentiert: echte Arbeit in unserer eigenen kleinen Ecke, ein *Spielraum*, nicht ein Messestand.

Wir nennen es **Elektor Labs Maker Space** und bei uns geht es bewusst leger zu. Alles beginnt bei den großen Tischen, von einem einfachen „Möchten Sie gerne eine Tasse Kaffee?“ über „Hier ist Energie, um Ihr Handy aufzuladen“ bis hin zu „Sie können gerne mit unseren Werkzeugen etwas löten oder messen“. Die großen Tische sind auch das Zentrum der kostenlosen Mini-Workshops, an denen Sie teilnehmen können. Das Team des Elektor-Labors ist da, Sie können plaudern, Ihr Lieblings-Projekt durchkauen oder Antworten auf Ihre technischen Fragen erhalten.



Was gibt es sonst noch im Elektor Labs Maker Space zu entdecken?

Auf unseren Tischen gibt es alle Arten von Equipment. Perfektionieren Sie Ihre Lötkünste mit LötKolben von Conrad, fangen Sie flüchtige Signale mit dem Hameg-Oszilloskop ein oder machen Sie sich schlau mit der Messgeräteplattform Virtual Bench von National Instruments. Wir halten T-Boards bereit und viele andere Embedded-Boards, damit Sie damit experimentieren können. Wir bringen auch einige Kartons mit Elektor Preferred Parts (ELPP) mit, damit Sie selbst Schaltungen aufbauen können.

Wir werden jeden Tag eine Reihe von Mini-Workshops durchführen. Einige machen wir selbst über Themen, die wir interessant finden. Andere werden von Partner-Unternehmen veranstaltet. Lernen Sie den Redakteur Jan Buiting kennen, der normalerweise das ganze Retronik-Zeug herumschleppt und Experte für die Verbindung alter und neuer Technologien ist. Jan hat Alkertümchen im Gepäck und wird auch ein paar Sätze dazu verlieren. Luc Lemmens ist auch da und demonstriert die neuen T-Boards von Elektor.

Mittwochnachmittag um 16.00 Uhr gibt es ein Webinar live am Elektor-Stand, Jan Visser und Jaime werden die Fragen von Lesern beantworten. Besuchen Sie unseren Maker Space oder nehmen Sie online teil!

Für mehr als 2.000 Besucher gibt es schöne Goody-Bags, vollgestopft mit Zeitschriften und anderem Material - hinterlassen Sie Ihren Namen, Ihre E-Mail-Adresse oder Visitenkarte und erhalten Sie eine Tasche.

Also, wenn Sie müde vom Herumlaufen und Herumreden sind, Ihre Batterien etwas erschöpft sind, wenn Sie Ihre Krawatte neu binden müssen, kostenloses WLAN oder einen „Clooney-Kaffee“ möchten, wenn Sie sich nach praktischer Elektronik sehnen oder einfach nur die Elektor-Goody-Bag wollen ... im Namen des Elektor-Teams: Herzlich willkommen!

(140327)



DesignSpark Tipps & Tricks

Neunummerierung von Bauteilen

Von **Neil Gruending**
(Kanada)

Es ist Ihnen bestimmt schon aufgefallen, dass bei vielen Projekten die Bauteilnummern schön sauber aufeinander folgen, sodass man sich gut daran orientieren kann. In dieser Folge geht es um die Renummerierung der Bauteile einer Platine und das anschließende Übertragen dieser Änderungen auf die Schaltung mit Hilfe eines Backward Annotation genannten Verfahrens.

Neue Bauteilenummern

Die Tools zum Umbenennen von Bauteilen in DesignSpark funktionieren so, dass jede Seite einer Platine in Streifen aufgeteilt wird und die Bauteilnummern dann passend zu dieser räumlichen Orientierung angepasst werden. Man konfiguriert das Tool durch Einstellung der Ausrichtungen nach „Left to Right“, „Top to Bottom“, „Right to Left“ und „Bottom to Top“. Normalerweise ordnet man die Bauteile nach „Left to Right“ und „Top to Bottom“, ganz wie auf der linken Seite von **Bild 1** gezeigt. Man kann das natürlich auch anders machen. Die rechte Seite des Bilds zeigt, wie die Anordnung „Right to Left“ und „Bottom to Top“ aussieht. In DesignSpark legt man zuerst die Richtung der Streifen fest und dann die Richtung innerhalb der Streifen. Bei „Left to Right“ und „Top to Bottom“ wird die Streifenrichtung auf „Left to Right“

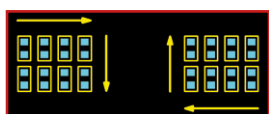


Bild 1.
Richtung der
Neunummerierung.

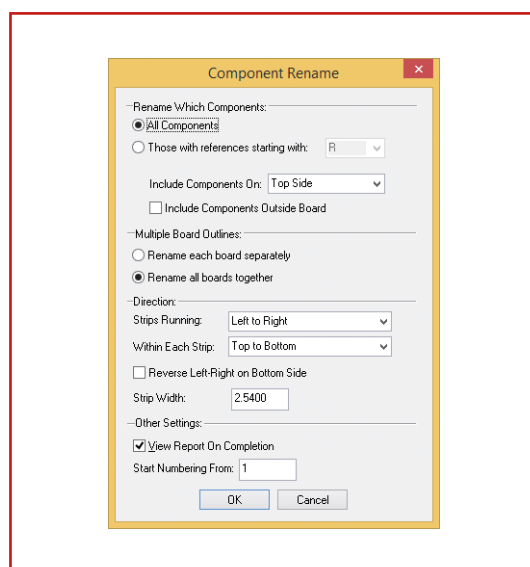


Bild 2.
Das Component-Rename-
Fenster.

gesetzt und die Richtung innerhalb der Streifen auf „Top to Bottom“. Die einzige Bedingung von DesignSpark ist, dass man eine horizontale mit einer vertikalen Richtung kombinieren muss. Die Kombination „Top to Bottom“ mit „Bottom to Top“ etwa funktioniert nicht.

Das Rename-Tool scannt dann jeden Streifen in der festgelegten Richtung und sucht nach Bauteilen. Das erste gefundene Exemplar wird mit der ersten freien Bauteilnummer (wenn möglich beginnend mit der Nummer 1) belegt. Eventuelle weitere Bauteile dieses Streifens werden dann in Suchrichtung sukzessive mit freien Nummern belegt. Im Beispiel von Bild 1 (links) wird jeder Streifen von links nach rechts und dann von oben nach unten geordnet.

Nun ein Blick auf das Component-Rename-Fenster von **Bild 2**, das erscheint, wenn man im Tools-Menü auf „Auto Rename Components“ klickt. Die Auswahl „Rename Which Components“ dient zur Spezifikation der Bauteile, die man umbenennen will. Normalerweise wählt man hier „All Components“, doch kann man auch Bauteile anhand ihrer Nummer und der Boardseite auswählen. Die Auswahl „Multiple Board Outlines“ ist sinnvoll, wenn mehrere Boards in einer Design-Datei enthalten sind, weshalb man diese Standardeinstellung unverändert lässt, damit in solchen Fällen die Bauteile aller Boards zusammen umbenannt werden. Bei der Richtungsangabe wurden die Richtungen ja schon festgelegt. Der Parameter „Strip Width“ legt die Streifenbreite beim Scannen des Boards fest. Ich empfehle, die Option „Reverse Left-Right“ zu aktivieren, damit DesignSpark die Richtungen auf der Rückseite der Platine umkehrt. Das ist sehr praktisch, denn es entspricht einem Umdrehen der Platine vor der Neunummerierung. Bei der Auswahl „Other Settings“ kann man festlegen, bei welcher Zahl die Nummerierung starten soll.

Bild 3 zeigt ein Beispiel, in dem das Board links durch Renaming in das Board rechts verwandelt wurde. Als Richtungen sind „Left to Right“ und „Top to Bottom“ festgelegt. Die Streifenbreite ist mit 5 mm sehr schmal gewählt, um ein Board mit mehreren Streifen demonstrieren zu können.

Die Nummern der Transistoren haben sich übrigens nicht verändert. Dies liegt daran, dass es für jeden Bauteil-Typ (R, C, Q etc.) einen eigenen Zähler gibt.

Update der Schaltung

Nachdem die Platine wunschgemäß neu nummeriert wurde, sollten die neuen Nummern auch in den Schaltplan übertragen werden. Hierzu wird ein Prozess namens Backwards Annotation verwendet. Man stößt diesen Prozess über das Menü *Tools* → *Backwards Annotation* an. Dadurch werden alle Bauteilenummern von der Platine in die Schaltung übertragen, womit beide wieder übereinstimmen. Andere Änderungen wie etwa bei Netzverbindungen oder Bauteileigenschaften werden nicht übertragen.

Das Annotation-Tool geht davon aus, dass eine oder mehrere Schaltungen mit der Platine verknüpft sind. Wenn man nicht mit einer Projekt-Datei arbeitet, dann geht das Tool davon aus, dass die Datei mit der Schaltung den gleichen Namen wie den der Platine trägt. In den Projekteinstellungen kann man aber festlegen, dass mehrere Schaltungsdateien mit unterschiedlichen Namen mit einer Platine verknüpft werden. Das kann man nicht übergehen und stattdessen etwa manuell die upzudatende Schaltung auswählen.

Bild 4 zeigt das Backward-Ground-Annotation-Fenster. Ein Klick auf „OK“ aktiviert das Update der Schaltung. Man kann sich aber schon vorher ansehen, was passiert, indem man „View Renames“ anklickt. Diese Überprüfungsmöglichkeit vor dem Update kann hilfreich sein. Bei „Delete Renames“ passiert etwas Anderes: Die Schaltung bleibt unverändert, aber die Änderungen werden markiert. Damit kann man sehr gut die Änderungen an der Schaltung manuell vornehmen. Man sollte damit vorsichtig sein, denn wenn man die Umbenennungen gelöscht hat, kann man sie nicht später wieder aktivieren. Die letzte Option ist „View Report on Completion“. Hierdurch wird eine Textdatei erzeugt, die detailliert die Änderungen enthält.

Es ist empfehlenswert, dass man vor einem Klick auf „OK“ alle Schaltplan-Dateien sichert, denn nach der Übernahme der Änderungen kann man dies nicht mehr ungeschehen machen.

Die linke Seite von **Bild 5** zeigt das Beispiel vor der Umbenennung - die rechte Seite den Zustand danach. Wie man sehen kann, wurden alle geänderten Bauteilenummern in die Schaltung übernommen, ohne die Schaltung selbst zu verändern.

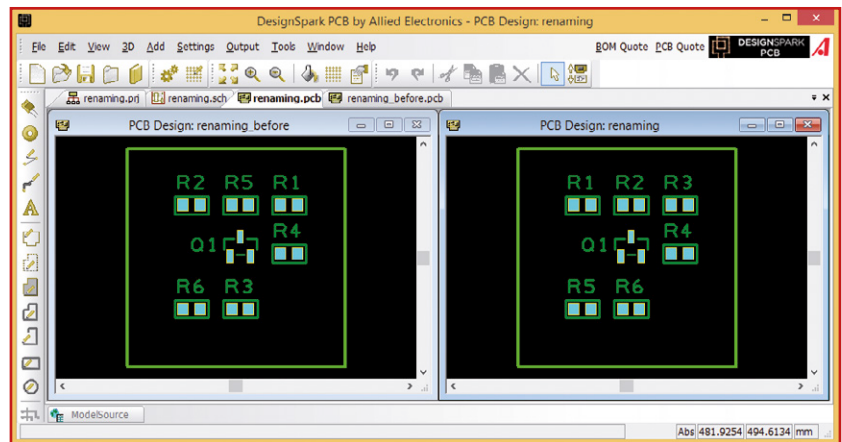


Bild 3.
Umbenannte Bauteile,
vorher und nachher.

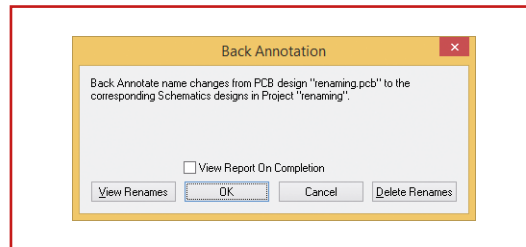


Bild 4.
Das Backwards-Annotation-
Tool.

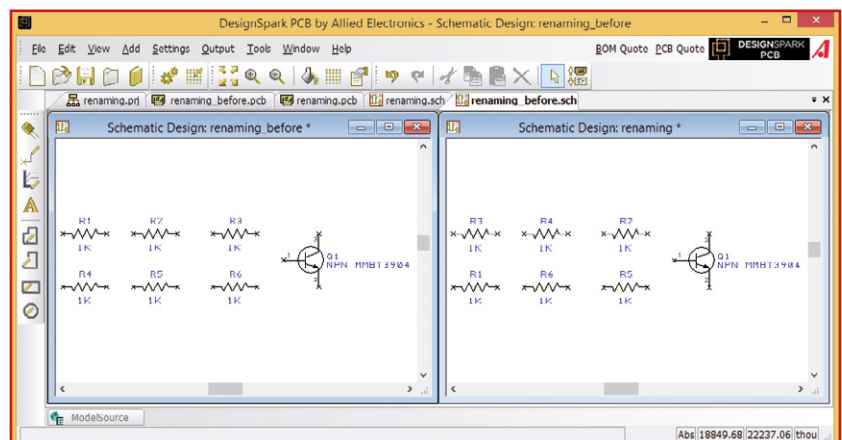
Fazit

In dieser Folge wurde das Component-Renaming-Tool vorgestellt. Dies ist der einzige Weg, automatisch Bauteile umzubenennen.

Gut gewählte Bauteilenummern erleichtern das Finden dieser Bauteile auf größeren Boards und damit eine Fehlersuche beträchtlich. In der nächsten Folge geht es um Bauteile an sich und wie man mit ihnen in der Schaltung und beim Platinen-Layout am besten umgeht.

(140292)

Bild 5.
Beispiel einer neu
nummerierten Platine.



Strommonitor-ICs

Ströme messen mit $m\Omega$ -Shunts

Von
Viacheslav Gromov
(D)

Shunt-Widerstände sollen klein sein und der resultierende Spannungsabfall am besten massebezogen, wenn man Mikrocontroller zur Messung von Strömen einsetzt. Für diese Zwecke gibt es kleine Spezial-ICs, die dank integrierter Verstärkung beiden Anforderungen gerecht werden.

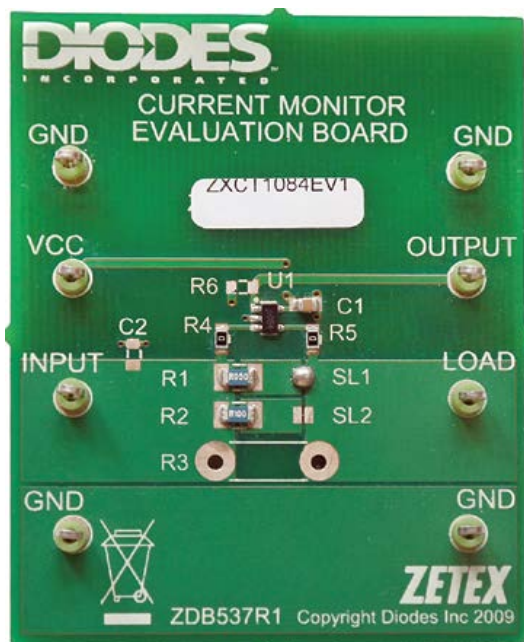


Bild 1.
Ein typisches Evaluation-Board. Der Typ ZXCT1084EV1 ist mit dem IC ZXCT1084 bestückt. Es wird ein ausführliches Manual mitgeliefert, das Versuchsaufbauten beschreibt.

Ströme mit einem Shunt-Widerstand zu messen ist einfach und bewährt. Je kleiner der Widerstand eines Shunts aber ausfällt, desto geringer der Spannungsabfall und desto weniger Leistung wird unnütz verheizt. Für kleine Widerstände braucht es daher extra Verstärkung, damit der Spannungsabfall z.B. zum Eingangsspannungsbereich des AD-Wandlers eines Mikrocontrollers passt. Hinzu kommt, dass das Einschleifen eines Shunts in die Masseleitung einer Stromversorgung häufig nicht opportun ist, ein Mikrocontroller aber eine zu messende Spannung erwartet, die entweder direkt massebezogen ist oder auch bei differentiellen Eingängen zumindest im begrenzten Gleichtaktbereich der analogen Eingangsstufe liegt. Man könnte diese Probleme zwar mit ein paar Opamps lösen, die man trickreich

verschaltet, doch besser sind spezielle ICs, die den Spannungsabfall am Shunt verstärken und deren Ausgangsspannung gleichzeitig auf Masse bezogen ist, sodass es auch bei höheren Versorgungsspannungen keine Probleme gibt.

Der Hersteller Diodes mit Sitz in Plano (Texas) hat sich auf analoge Bausteine und diskrete Bauteile spezialisiert. Seine Strommonitor-ICs aus der Zetex-Produktlinie sind sehr stromsparend. Diese ICs gibt es mit unterschiedlichen Eigenschaften. Die ICs sind mit ihrem fünfpoligen SOT23-Gehäuse sehr klein und brauchen nur wenig externe Beschaltung. Für einige Typen gibt es auch extra Evaluation-Boards (**Bild 1**). Die Anwendungsbereiche sind, nicht zuletzt wegen des großen Spannungsbereichs, sehr vielfältig. Sie reichen von einer einfachen Stromüberwachung über das Batterie-Management bis zum Automotive-Sektor.

Die Familie

Die ICs der Zetex-Familie unterscheiden sich zunächst anhand der Spannungsverstärkung: Die Typen ZXCT1084 und ZXCT1085 verstärken fix um den Faktor 25. Die beiden Varianten ZXCT1086 und ZXCT1087 kommen hingegen auf einen Verstärkungsfaktor von 50. Ihre am Pin *OUT* zur Verfügung stehende Ausgangsspannung kann direkt mit einem ADC-Eingang eines Mikrocontrollers verbunden werden. Die Ausgangsimpedanz ist mit 125 k Ω zwar relativ hoch, doch ADC-Eingänge sind in der Regel sehr hochohmig.

Der feste Verstärkungsfaktor erlaubt die Einsparung eines Spannungsteilers, mit denen man ansonsten die Verstärkung einstellen müsste. Doch auch wer diese Option benötigt, kann sich bedienen: Die beiden Typen ZXCT1082 und ZXCT1083 brauchen drei zusätzliche Widerstände,

mit denen die Verstärkung individuell eingestellt werden kann.

Gerade für den Batteriebetrieb ist die niedrige Stromaufnahme von nur 0,6 µA ideal. Über den Shunt fließt bei einer Messung ein Messstrom von maximal 1,7 µA (hängt vom zu messenden Strom ab). Mit einer typischen Reaktionszeit von 5...7 µs können diese ICs auch schnellere Stromänderungen problemlos erfassen (**Bild 2**). Die Zetex-Familie kennt noch einen weiteren Unterschied: Die Typen ZXCT1082/84/86 können mit Spannungen von 2,7...60 V versorgt werden, während die drei anderen Varianten ZXCT1083/85/87 nur für Spannungen von 2,7...40 V geeignet sind.

Die Beschaltung

Bild 3 zeigt die relativ simple Beschaltung der Typen mit fester Verstärkung, während **Bild 4** den minimal höheren Aufwand an externen Bauteilen (die drei Widerstände R_{GT} sowie R_G) der Exemplare mit einstellbarer Verstärkung abbildet. Die Verstärkung der beiden Typen ZXCT1082/83 ergibt sich nach folgender Formel:

$$U_{OUT} = U_{SENSE} * R_G / R_{GT}$$

Diese beiden IC-Typen verfügen im Gegensatz zu den anderen über einen Stromausgang. Dank der Verstärkung können die Shunts recht niederohmig ausfallen, was gerade bei höheren Strömen den Vorteil einer geringen abzuführenden Verlustleistung mit sich bringt. Das ist für SMD-Widerstände als Shunts nicht zu unterschätzen. Benötigt ein AD-Wandler eines Mikrocontrollers beispielsweise 1 V für „Vollausschlag“, ergibt sich bei 50-facher Verstärkung für einen Messbereich von 1 A demnach nur ein Shunt von 20 mΩ statt 1 Ω, was beim Strom von 1 A zu einer maximalen Verlustleistung von nur 20 mW statt 1 W führt. Die Verstärkung ist daher direkt der Leistungsreduktion proportional.

Weitere Daten und Beschaltungsmöglichkeiten sowie Datenblätter und andere Strommonitor-ICs finden sich unter [1]. In den Datenblättern sind auch viele Informationen und Diagramme über das Zeit- und Frequenzverhalten sowie über die Temperaturstabilität enthalten.

(140179)

Weblink

[1] http://diodes.com/catalog/voltage_output_80

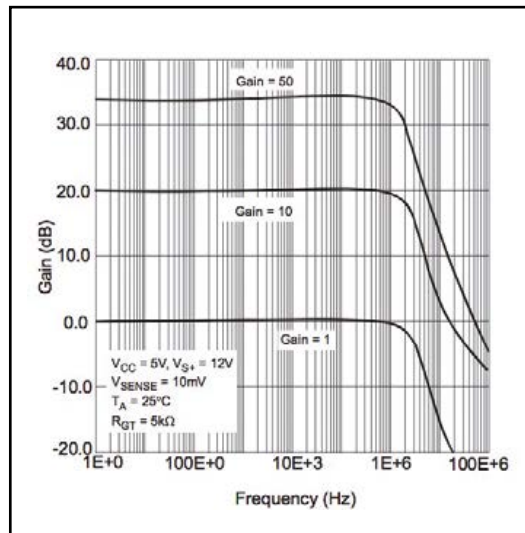


Bild 2.
Der Frequenzgang bei unterschiedlichen Verstärkungen.

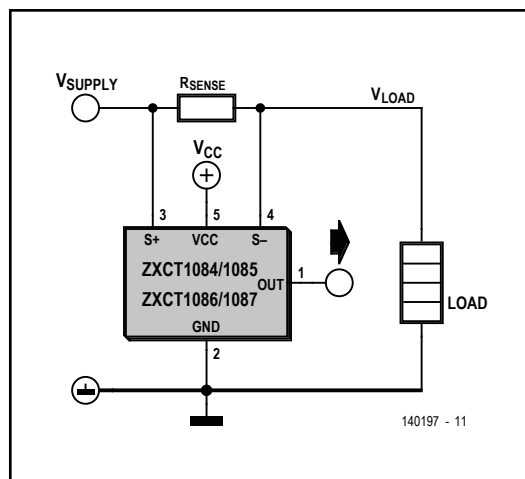


Bild 3.
Die Beschaltung von ZXCT1084/85/86/87 ist minimalistisch.

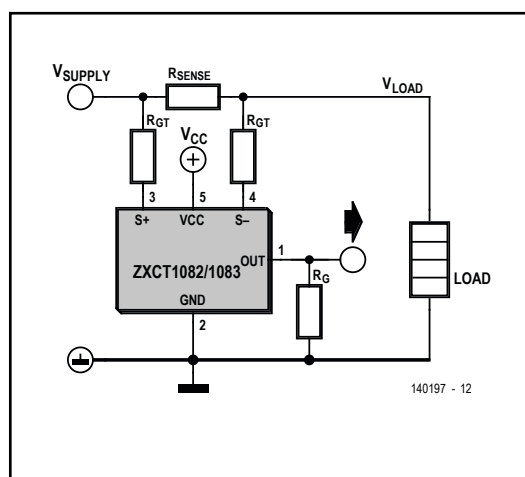


Bild 4.
Bei den Typen ZXCT1082/83 kommen für eine variable Verstärkung noch drei Widerstände hinzu.



Glimmlampen

Seltsame Bauteile (10)

Von **Neil Gruending**
(Kanada)



Bild 1.
Glimmlampen.

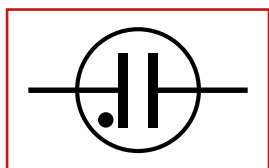


Bild 2.
Schaltsymbol der
Glimmlampe.

Eine Glimmlampe in einer alten Tektronix-Stromversorgung veranlasste mich, etwas über dieses klassische Bauteil zu forschen. Erst dachte ich, es wäre nur eine Power-Anzeige, aber bald zeigte sich überraschenderweise, dass die Lampe als Schutzschaltung fungierte. Wie kann das funktionieren?

Glimmlampen (nicht zu verwechseln mit Leuchtstofflampen) sind ziemlich einfache Geräte. Ein Glasröhrchen, an dessen Ende(n) zwei Elektroden eingelassen sind, wird unter niedrigem Druck mit einem Neongas-Gemisch gefüllt (**Bild 1**). Wenn man eine Gleichspannung an die Elektroden legt, glimmt nur die negative Elektrode, legt man dagegen eine Wechselspannung an, so leuchten beide. Das Schaltzeichen der Glimmlampe ist in **Bild 2** dargestellt.

Eine Glimmlampe verhält sich wie ein offener Stromkreis, wenn sie ausgeschaltet ist. Legt man eine Spannung an, die ausreichend hoch ist und die Gasdurchbruchspannung (Zündspannung) erreicht, wird der Stromkreis geschlossen und die Lampe leuchtet.

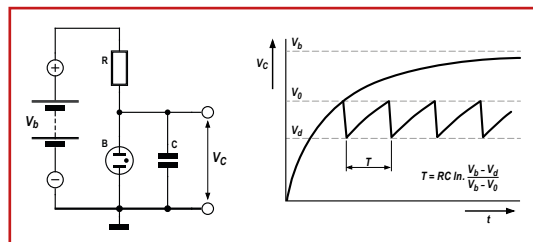


Bild 3.
Ein Kipposzillator mit
Glimmlampe und sein
typisches Ausgangssignal.

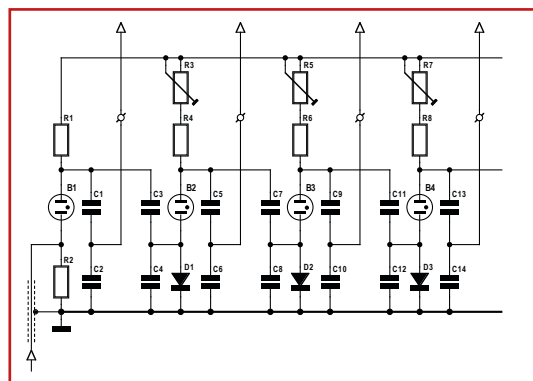


Bild 4.
Die kaskadierten
Glimmlampen arbeiten
als Frequenzteiler in einer
elektrischen Orgel.

Sobald die Lampe leuchtet, bildet sich Plasma in der Röhre, so dass die Betriebsspannung deutlich sinkt. Sie bleibt dann über einen ziemlich weiten Strombereich konstant. Dieses Verhalten ist auch der Grund für einen großen ($>100\text{-k}\Omega$ -)Widerstand in Reihe mit der Lampe. Die Zündspannung liegt bei 90 V für eine gewöhnliche NE-2-Lampe, die Haltespannung ist mit 60 V deutlich geringer. Da die Spannungsschwellen durch Umgebungslicht beeinflusst werden, ist der Röhre eine kleine Menge radioaktives Argon oder Krypton hinzugefügt, um sicherzustellen, dass die Lampe wie gewünscht leuchtet.

Aber wie werden Glimmlampen in einer Stromversorgung verwendet? Zünd- und Haltespannung stellen quasi eine Hysterese dar, die als Überspannungsschutz eingesetzt werden kann. Besonders gut ist, dass die Lampe einen Fehlerzustand auch optisch anzeigt. Heutzutage gibt es mit MOVs oder TVS-Dioden bessere Schutzrichtungen, aber Glimmlampen stellen immer noch einen guten Hochspannungsindikator für Stromversorgungen dar.

Dank ihrer negativen Widerstandscharakteristik eignen sich Glimmlampen auch für andere Anwendungen, etwa als aktives Element in einem Oszillator mit Widerstand und Kondensator (**Bild 3**). Zwei Glimmlampen arbeiten als astabiler Multivibrator und wie in **Bild 4** kaskadiert als Ringzähler, Teiler oder andere einfache Logikschaltung.

Ich habe Glimmlampen tatsächlich in vielen Anwendungen und an unerwarteten Orten wie in der besagten Stromversorgung oder in einer elektrischen Orgel aus den 1960ern entdeckt. Das Beste an ihnen ist aber, dass sie immer noch in einer Vielfalt von Ausführungen leicht und preiswert erhältlich sind und sich dadurch ideal für Experimente eignen. Bei neueren Modellen liegt die Zünd- näher an der Haltespannung als früher, so dass man einige Schaltungen eventuell anpassen muss, damit sie auch richtig funktionieren.

(140291)

Weblink

[1] <https://de.wikipedia.org/wiki/Glimmlampe>

KEYSTONE SYSTEM

Ideal für individuelle Anwendungen in den Bereichen Home Entertainment, Netzwerk, Industrie und Hobby.

DELOCK
we move the world

- ✓ große Modulauswahl für individuelle Konfigurationen
- ✓ viele Einbauvarianten – mit und ohne Halterung
- ✓ einfache Snap-in-Montage

SNAP-IN-MONTAGE:



• MODULE • ABDECKUNGEN • GEHÄUSE • HALTERUNGEN • PATCHPANELS •

COAXIAL



F-Buchse > F-Buchse
DELOCK 86300 3,95



BNC-Buchse > BNC-Buchse
DELOCK 86301 3,95



IEC > F-Buchse
DELOCK 86302 3,95

MULTIMEDIA



Cinch-Buchse > Cinch-Buchse
DELOCK 86305 ● 3,95
DELOCK 86306 ● 3,95
DELOCK 86307 ○ 3,95



Cinch-Buchse > Cinch-Buchse
DELOCK 86308 ● 3,95
DELOCK 86309 ● 3,95
DELOCK 86310 ○ 3,95



Cinch-Buchse > F-Buchse
DELOCK 86311 ● 3,95
DELOCK 86312 ● 3,95
DELOCK 86313 ○ 3,95



Polklemme > Gewindebuchse
DELOCK 86304 ● 3,95
DELOCK 86303 ● 3,95



Toslink Buchse > Toslink Buchse
DELOCK 86337 3,95



Terminalblock > Terminalblock
DELOCK 86323 2,95

PC-MODULE



HDMI-Buchse > HDMI-Buchse
DELOCK 86316 7,95



USB 2.0-A-Buchse > USB 2.0-A-Buchse
DELOCK 86317 5,95



USB 2.0-B-Buchse > USB 2.0-B-Buchse
DELOCK 86318 5,95



USB 3.0-A-Buchse > USB 3.0-A-Buchse
DELOCK 86319 8,95



USB 2.0-A-Buchse > USB 2.0-B-Buchse
DELOCK 86320 5,95



FireWire-A-Buchse > FireWire-A-Bu., 4-pin.
DELOCK 86325 2,95

ABDECKUNGEN

Staub- und Insektenschutz im 4er-Pack



weiß, ohne Kabel-durchführung
DELOCK 86314 1,95



schwarz, ohne Kabel-durchführung
DELOCK 86321 1,95



weiß, mit Kabel-durchführung
DELOCK 86315 1,95

EINBAUGEHÄUSE

Einzelhalter für den Geräteeinbau



DELOCK 86213 1,95

NETZWERK



RJ45-Buchse > LSA Cat.6A
DELOCK 86205 4,95

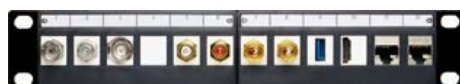
GEHÄUSE



Leergehäuse zur individuellen Bestückung

DELOCK 86226 Leergehäuse, 1 Port 1,50
DELOCK 86207 Leergehäuse, 2 Ports 1,95
DELOCK 86208 Leergehäuse, 4 Ports 2,95
DELOCK 86227 Leergehäuse, 6 Ports 3,50
DELOCK 86209 Leergehäuse, 8 Ports 3,95

PATCHPANEL



Beispiel: 10"-Patchpanel (12 Ports) mit verschiedenen Keystone-Modulen

Patchpanel für 19"-Racks oder 10"-Wandschränke

- standardisierte Maße
- einfache Snap-In-Montage
- für 12 bzw. 24 Keystone-Module
- integriertes Beschriftungsfeld

DELOCK 43259 10", 12 Ports 6,95
DELOCK 43277 19", 24 Ports 14,95

DIN-HUTSCHIENEN-HALTER

DIN-Hutschienenhalterung zur einfachen Snap-in-Montage eines Keystone-Moduls.

- für 35-mm-Hutschienen
- mit Beschriftungsfeld
- anreihbar
- Maße: 65 x 18 x 78,5 mm

DELOCK 86212 1 Halter 1,95
DELOCK 86272 1 Abdeckung 1,50

Entdecken Sie unser komplettes **KEYSTONE PROGRAMM**

Einfach QR-Code scannen oder Kurzlink aufrufen:



<http://rch.lt/gS>

Jetzt bestellen: **www.reichelt.de**

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

Katalog 06.1/2014!

Kostenlos – Jetzt anfordern!

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

Preisstand: 30. 9. 2014
Tagesaktuelle Preise:
www.reichelt.de

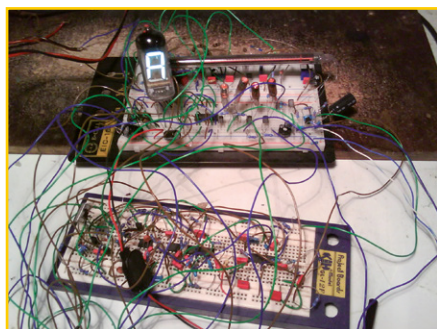
Elektor.Labs, die Jobmaschine

Von **Clemens Valens**
(Elektor-Labor)



Get Elektorized!

Erinnern Sie sich daran? Sie sagen ja, aber erinnern Sie sich tatsächlich? Ich schon, und ich weiß jetzt, zumindest eine weitere Person auch. Diese Person veröffentlichte einige Projekte auf Elektor.



Karriere machen mit
Spaghetti-Breadboard.

Labs und bekam daraufhin einen Job. *Wie bitte?* Ja, Sie hören richtig! Man wird nämlich nach der Veröffentlichung interessanter Projekte auf unserer Website eingestellt. Genau dies passierte Labs-User *lux36*. Und was noch besser ist, er erzählte mir davon. Herzlichen Glückwunsch, *lux36*! Sicherlich gibt es noch andere Menschen, die in irgendeiner Weise „elektorisert“ wurden, aber leider hat uns

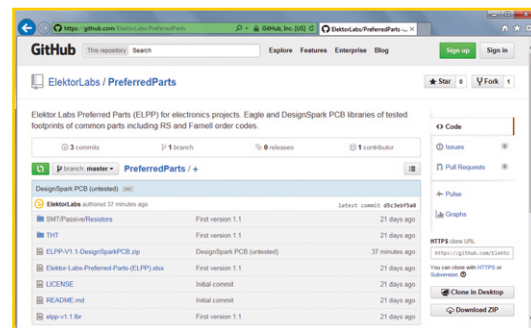
nie jemand davon erzählt. Das ist das Problem mit den Menschen: Immer wenn es Grund gibt, sich zu beschweren, wissen sie, wie sie einen finden, aber wenn alles in Ordnung ist, kein Wort. Na ja, so ist das Leben. *lux36* verdient jetzt gutes Geld und alles, was er dafür tat, war, seine Projekte online zu stellen. Und er kann noch mehr Geld daraus machen, wenn wir uns entscheiden sollten, seine Projekte in Elektor zu veröffentlichen. Aber dafür muss er dem Elektor-Labor mit ein wenig mehr Informationen helfen, weil die Schaltpläne fehlen und das Leiterplatten-Design unvollständig ist. Lieber *lux36*, wenn Sie dies lesen, fügen Sie bitte Ihrem Projekt einige Details an und vielleicht werden Sie dann noch einmal „elektorisert“.

www.elektor-labs.com/node/3169

Elektor.Labs Preferred Parts (ELPP): Nun bei GitHub zu Hause

In der Ausgabe September 2014 haben wir *Elektor.Labs Preferred Parts* (ELPP) vorgestellt, eine Bauteile-Bibliothek mit in Elektor häufig verwendeten

Bauteilen, ihren Anschlussbelegungen und den Bestellnummern bei populären Distributoren. Dies sollte uns und Ihnen die Suche und Beschaffung von Bauteilen erleichtern. Diese Bibliothek enthält nicht nur elektronische, sondern auch elektromechanische Bauteile wie Steckverbinder, Relais und Schalter. Wir haben eine Menge positiver Mails erhalten, was uns natürlich motiviert hat, den (mühsamen) Aufbau der Bibliothek weiter voranzutreiben. Die ELPP-Bibliothek wurde ursprünglich auf unserem FTP-Server gestellt, aber wir dachten, dass



ELPP, gehostet bei GitHub.

es vielleicht besser wäre, sie auf einer gut etablierten und für alle leicht zugänglichen Online-Plattform zu „hosten“. Wir haben uns für GitHub entschieden. Neben der guten Zugänglichkeit ist es das Versionskontrollsystem gewesen, das uns zu diesem Schritt bewog. Durch dieses System verfügen Sie nicht nur immer über die neueste Version der ELPP, sondern erhalten auch einen Überblick über die Veränderungen an der Bibliothek. Übrigens, wir haben gerade der ELPP die *DesignSpark PCB library* hinzugefügt...

<https://github.com/ElektorLabs/PreferredParts>

(140305)



Kundenbewertungen



Rund 97 % unserer Kunden sind vom reichelt-Service überzeugt*

*Quelle: Shopauskunft.de (1. 9. 2014)

NEUESTE
EINPLATINEN-
COMPUTER

ZU EINEM
TOP-PREIS!



Das passt perfekt zum Banana Pi

4-teiliger Kühlsatz

- einfache Montage
- extrem niedriges Profil



TEK-BERRY COOL **4,65**

WLAN-USB-Adapter, EDIMAX 150 Mbit/s

- unterstützt WEP, WPA, WPA2
- WPS-kompatibel



EDIMAX EW-7811UN **6,90**

Micro-USB-Netzteil 6W, 5V, 1,2A

- doppelte Isolation
- Überlast- und kurzschlussfest



HNP06 MICRO USB **6,40**

MicroSDHC-Card Class 10

- inklusive Adapter
- Kapazität: 8 GB



SDC10/8GB **6,80**

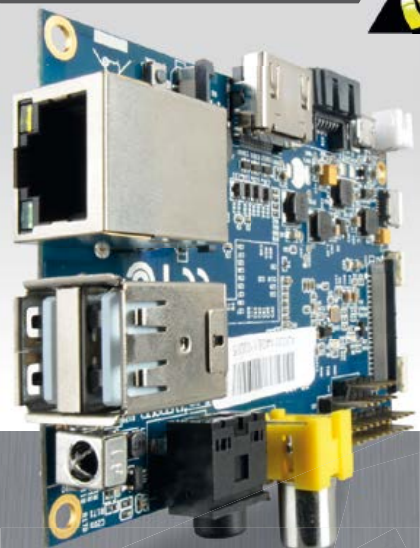
Banana Pi

Kompakt wie der Raspberry Pi,
aber deutlich mehr Leistung!



- ✓ schnellere CPU
- ✓ mehr Arbeitsspeicher
- ✓ schnelles Ethernet
- ✓ Infrarot Empfänger

- CPU: A20 ARM® Cortex™-A7-Dual-Core
- Arbeitsspeicher: 1 GB DDR3 SDRAM
- Anschlüsse: SD-Kartenslot, SATA-Port, HDMI, 10/100/1000-Mbit/s-Ethernet, 2x USB-Port, CSI-Kamera-Anschluss, 26 externe PINs mit I²C, SPI, UART, CAN, 3,5-mm-Audio-Ausgang
- Stromversorgung über Micro-USB



BANANA PI

42,90

Passt wie angegossen! Kunststoffgehäuse für Ihren Banana Pi

Mixen Sie Ihre eigene Farbkomposition ganz nach Ihrem Geschmack. Dank der Clip-in-Nuten ist kein lästiges Schrauben oder Kleben erforderlich.

- alle Ausschnitte vorgeschritten
- rutschhemmende Gummifüße
- Hochglanz-Oberfläche

	Oberteil	CB DBBP AA GE	
	Unterteil	CB DBBP BA GE	4,15
	Oberteil	CB DBBP AA TR	6,20
	Unterteil	CB DBBP BA TR	4,65
	Oberteil	CB DBBP AA SW	6,90
	Unterteil	CB DBBP BA SW	4,15
	Oberteil	CB DBBP AA CF	6,20
	Unterteil	CB DBBP BA CF	4,65
			6,90



Lieferung ohne
Banana-Pi

Jetzt bestellen: www.reichelt.de

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

Katalog
06.1/2014
Kostenlos –
Jetzt anfordern!



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

Preisstand: 30. 09. 2014
Tagesaktuelle Preise:
www.reichelt.de

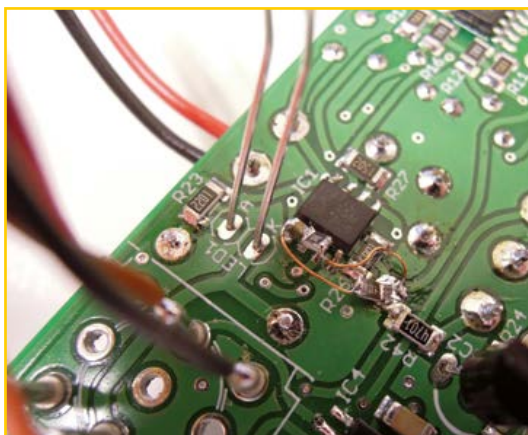
Von Hinkelsteinen und Hügelgräbern

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

Prototyping! Stunden des drögen Anstarrens eines CAD-Bildschirms, ein leichtes Zaudern, wenn Bauteile beschafft und die Platine bestellt werden müssen. Endlich kräftiges Daumendrücken, wenn der On-Schalter des Labornetzgeräts umgelegt wird, um das neu gestaltete und frisch aufgebaute Projekt zum ersten Mal mit Energie zu versorgen. Oft genug gefolgt von „zurück ans Reißbrett“ oder, wenn man Glück

alles. Wie Sie mit einiger Mühe erkennen können, wurde ein Kondensator zwischen Pin 1 und Pin 2 und ein 10-k Ω -Rückkopplungswiderstand zwischen Pin 2 und dem 100- Ω -Serienwiderstand am Ausgang Pin 1 angeschlossen.

Mit bedrahteten Bauteilen wäre die Korrektur der Schaltung ein Klacks gewesen. Aber nein, Ton insistiert auf der „0805“ auf den Widerständen und der „0603“ auf dem Kondensator. Letzte-



hat, zurück zur Lötstation, um Lötstellen auf potenzielle Kurzschlüsse zwischen ein paar SMD-IC-Pins zu überprüfen.

Das Elektor-Labor unterscheidet sich dabei durch nichts von anderen Entwicklungslaboren. Alle Entwickler durchlaufen diese schiere Endlosschleife von Design, Test, Korrektur, Test und alles wieder von vorne. Manchmal ist ein völlig neues Platinenlayout notwendig. Obwohl es manchmal einfacher erscheint, einen Prototyp „mit der Hand“ zu optimieren, kann dies in Hinkelsteinen und Hügelgräbern* enden. So erging es unserem Ingenieur Ton Giesberts beim Prototyping seines aktuellen Projekts (siehe erstes Foto).

Es wurde festgestellt, dass der Ausgang von IC1, eines TLC272-Doppeloperationsverstärkers, zum Schwingen neigt. Um dem abzuhelpfen, sollte ein 100-Ω-Widerstand in Reihe mit dem Ausgang des IC1 (Pin 1) geschaltet werden. Dazu mussten Pin 1 und Pin 2 „entlötet“ und von ihren Löt pads weggebogen werden. Aber das war noch nicht



rer befindet sich zwischen den Pins des SOIC-8-ICs. Einige Hinkelsteine waren die Folge, und das mit Vorsatz! Zum Vergleich, der Durchmesser des Kupferlackdrahts im Foto beträgt 0,2 mm (7,9 mil).

Natürlich ist das kein Einzelfall. Das rechte Foto zeigt ein schönes Beispiel eines Doppel-Hinkelsteins. Statt eines einzigen normalen Kondensators wurde eine Version mit hohem ESR (Equivalent Series Resistance) mit einem Standard-Kondensator und zwei parallelen 13- Ω -Widerständen (also mit 6,5 Ω ESR) erstellt. Eine Drahtbrücke verbindet dann die Hinkelstein-Komponenten und schafft ein echtes Miniatur-Hügelgrab.

(140290)

* Die Begriffe Tombstoning, Stonehanging und Manhattan-Effekt beziehen sich auf die Tendenz kleiner unbedrahteter Komponenten, während des Lötvorgangs wie ein Grabstein aufwärts zu kippen. Grund dafür ist die Oberflächenspannung des geschmolzenen Lotes.

Professional Quality
Trusted Service
Secure Ordering



Elektor PCB Service at a glance:

- ➔ 4 Targeted pooling services and 1 non-pooling service
- ➔ Free online PCB data verification service
- ➔ Online price calculator available
- ➔ No minimum order value
- ➔ No film charges or start-up charges

Delivery
from 2
working
days

Mikrocontroller für Einsteiger (7)

Der I2C-Bus

Von **Burkhard Kainka**

Wenn an einem Arduino Uno die Leitungen knapp werden, gibt es eine Lösung. Der I2C-Bus benötigt nur zwei Portleitungen, doch damit kann man bis zu 127 externe ICs ansprechen. Unzählige Bausteine auf dem Markt sind mit I2C-Interfaces ausgerüstet; die Palette reicht von einer einfachen Porterweiterung über EEPROMs bis zu vielerlei Sensoren. Im letzten Teil unseres Bascom-Kurses zeigen wir, wie der I2C-Bus funktioniert. Im Mittelpunkt stehen wie immer aber interessante Demo-Anwendungen.

Der I²C-Bus (Inter-IC-Bus, auch I2C geschrieben) ist eine 2-Draht-Datenverbindung mit Datenleitung und Taktleitung und wurde ursprünglich von Philips für die Unterhaltungselektronik entwickelt. In einem Fernseher oder Videorekorder gibt es meist einen zentralen Prozessor, der die Kontrolle über viele Baugruppen hat. Mit einem Datenbus aus nur zwei Leitungen lässt sich alles leicht verbinden. Der Prozessor ist der Master, ähnlich wie beim SPI-Bus, und die Peripheriebausteine sind die Slaves.

Der besondere Trick beim I2C-Bus ist, dass jeder Slave-Baustein eine 7 bit lange Adresse hat. Nur so können viele ICs am selben Bus angeschlossen sein, ohne sich gegenseitig zu stören. Neben RAMs, EEPROMs, Porterweiterungsbausteinen, Echtzeit-Uhren, AD- und DA-Wandlern gibt es eine ganze Reihe von speziellen I2C-Chips wie z.B. Anzeigentreiber, PLL-Bausteine und viele mehr.

Datenübertragung und Adressierung

Der I²C-Bus verwendet eine serielle Datenleitung SDA und die Taktleitung SCL. Wie bei Schieberegistern wird pro Takt jeweils ein Bit auf der Datenleitung übertragen, meist zuerst die Bits der Adresse und dann die Daten-Bits. Beide Leitungen sind jeweils mit einem Pull-Up-Widerstand versehen und können von jedem Teilnehmer auf Low-Pegel gezogen werden. **Bild 1** zeigt das Prinzip der Busverbindung. Der Master gibt das Taktsignal aus. Die Daten können jedoch sowohl vom Master als auch vom Slave kommen.

Der I2C-Bus funktioniert mit 5-V-Controllern/

Chips genauso wie mit 3,3-V-Elektronik. Man kann sogar beides mit ein- und demselben Bus verbinden! Die beiden Pullup-Widerstände von z.B. 2,2 k Ω ziehen die Leitungen auf 3,3 V beziehungsweise 5 V = High, solange keiner der Busteilnehmer die Leitungen durch seine Pull-Down-Transistoren auf Low zieht. Und ein 3,3-V-High ist auch für das 5-V-System ein High, weil die Spannung noch deutlich über der halben Betriebsspannung von 2,5 V liegt. Und Low ist überall 0 V. Das bedeutet für uns, dass wir problemlos einen Arduino Uno mit einem 3,3-V-Slave verbinden können. Das ist hilfreich, weil viele neuere ICs nur noch für 3,3 V vorgesehen sind.

Der ATmega328 des Uno verfügt über eine eingebaute I2C-Schnittstelle an den Leitungen PC4 und PC5 („Hardware-I2C“). Bascom bietet aber auch Spezialbefehle an, mit denen sich eine I2C-Software-Schnittstelle an beliebigen Portpins realisieren lässt (und man kann sich natürlich auch selbst Funktionen schreiben, welche die Leitungen mit einzelnen Codezeilen auf High und Low setzen). Hier wird ausschließlich das Software-I2C-Interface von Bascom benutzt, aber mit genau den Portpins, die auch von der Hardware-I2C-Schnittstelle des Controllers verwendet werden.

Diese Portpins werden auch auf dem Elektor Extension-Shield [1] für den Arduino Uno verwendet. Sie führen dort an den EEC/Gnublin-Stecker K2, wo Gnublin-Module mit I2C-Schnittstelle über ein Flachbandkabel angeschlossen werden können, zum Beispiel ein Modul mit acht Relais [2]. In den Busleitungen liegen noch zwei 330- Ω -Wi-

derstände, die man auch einfach weglassen könnte. Aber sie helfen gegen falsche Signale durch Reflexionen auf langen Busleitungen und können Fehler abfedern, die im Zusammenhang mit unterschiedlichen Betriebsspannungen passieren könnten. Wenn jemand z.B. versehentlich die Leitung PC4 dauerhaft auf 5 V hochschaltet, könnte ein unzulässiger Strom in die 3,3-V-Peripherie fließen, der nun aber durch den Widerstand begrenzt wird.

Das I²C-Bus-Protokoll kennt eine Reihe genau definierter Situationen, die es jedem Busteilnehmer gestatten, Anfang und Ende einer Übertragung zu erkennen:

- **Ruhezustand:** SDA und SCL sind high und damit inaktiv. Der Bascom-Befehl I2cinit schaltet die beiden Leitungen in diesen Ruhezustand, aber ohne Pullups, denn die liegen ja außen.
- **Startbedingung:** SDA wird vom Master heruntergezogen, während SCL high bleibt (Bascom: I2cstart).
- **Stoppbedingung:** SDA wechselt von low nach high, während SCL high bleibt (Bascom: I2cstop).
- **Datenübertragung:** Der jeweilige Sender legt acht Datenbits auf die Datenleitung SDA, die durch Taktimpulse auf der Clockleitung SCL vom Master weitergeschoben werden. Die Übertragung beginnt mit dem höchstwertigen Bit (Bascom: I2cwbyte Data).
- **Bestätigung (Acknowledge, Ack):** Der jeweils adressierte Empfänger quittiert den Empfang eines Bytes durch einen Low-Pegel an SDA, bis der Master den neunten Clock-Impuls an SCL erzeugt hat. Die Bestätigung bedeutet zugleich, dass ein weiteres Byte empfangen werden soll. Ein gewünschtes Ende der Übertragung muss durch das Ausbleiben der Bestätigung angekündigt werden (Nack). Die eigentliche Beendigung der Übertragung wird durch die Stoppbedingung erreicht (Bascom: i2crbyte Data, Ack oder i2crbyte Data, Nack).

Adressen werden genau wie Daten übertragen und quittiert. Im einfachsten Fall einer Datenübertragung vom Master zu einem Slave, z.B. einem Ausgabeport, laufen folgende Vorgänge ab: Der Master erzeugt die Startbedingung und überträgt dann in den Bits 7 bis 1 die Adresse des Portbausteins und in Bit 0 die gewünschte Richtung der Datenübertragung, nämlich 0 für

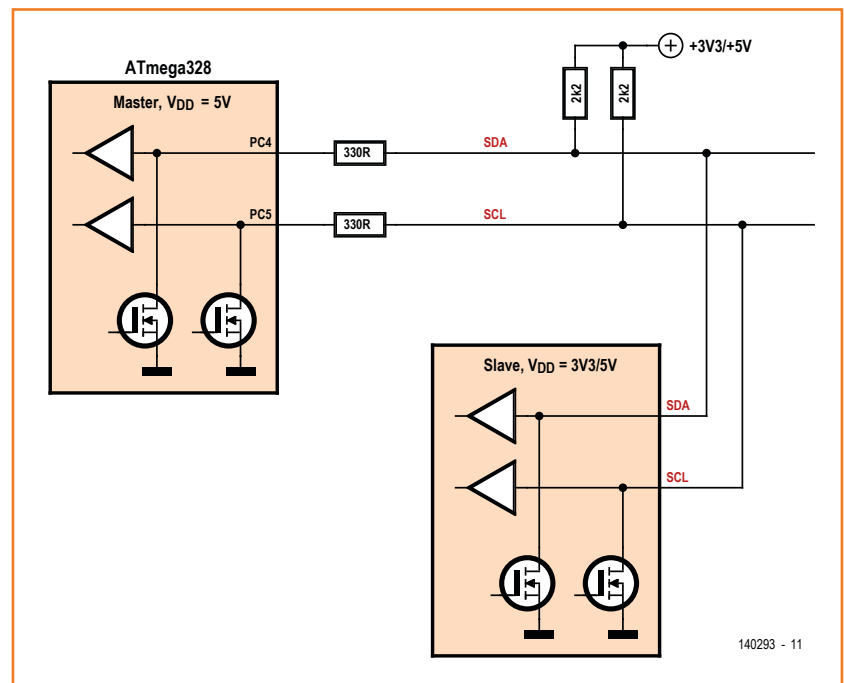
„Schreiben“. Die Adresse wird vom angesprochenen Slave quittiert. Dann sendet der Master das Datenbyte, das ebenfalls quittiert wird. Er kann nun die Verbindung durch die Stoppbedingung unterbrechen oder aber weitere Bytes an denselben Slave abschicken.

I²C-Bus-Adresse mit Datenrichtungsbit

A6	A5	A4	A3	A2	A1	A0	R/W
----	----	----	----	----	----	----	-----

Sollen Daten von einem Slave gelesen werden, muss die Adresse mit gesetztem Datenrichtungsbit übertragen werden. Der Master gibt jeweils acht Clock-Impulse aus und erhält acht Datenbits. Solange er den Empfang beim neunten Clock-Impuls durch Acknowledge bestätigt, kann er weitere Bytes empfangen. Die Übertragung wird schließlich vom Master durch eine ausbleibende Bestätigung und die Stoppbedingung beendet. Jeder I²C-Baustein hat eine festgelegte Adresse, die zu einem Teil typenspezifisch festgelegt ist und zu einem anderen Teil über herausgeführte Adressleitungen A0, A1 usw. vom Benutzer konfiguriert werden kann. Die Adressleitungen werden hierzu in der Schaltung fest auf High oder Low gelegt, um die Bits einzustellen. Bei drei herausgeführten Adressleitungen wie beim Portexpander PCF8574 können bis zu acht verschiedene Adressen eingestellt werden, also kann man bis

Bild 1.
I²C-Bus-Verbindungen zwischen Master- und Slave-ICs.



140293 - 11

Listing 1. Testen der aktiven I2C-Adressen.

```

'-----
'UNO_I2C1.BAS Test for valid Addresses
'-----

$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim Addr As Byte
...
Config Scl = Portc.5
Config Sda = Portc.4
I2cinit

For Addr = 2 To 254 Step 2
  I2cstart
  I2cwrite Addr
  If Err = 0 Then
    Print Addr
    Locate 2 , 1
    Lcd Addr
    Waitms 1000
  End If
  I2cstop
Next Addr
End

```

- RAMs und EEPROMs: $A0_{\text{hex}} = 160$
- Uhrenbaustein PCF8583: $A0_{\text{hex}} = 160$

Vom Benutzer können bei vielen ICs die letzten drei Adressbits eingestellt werden. Eine Echtzeituhr PCF8583 verwendet dieselbe Grundadresse wie ein RAM, weil sie ebenfalls ein RAM enthält. Falls man ein RAM und eine Uhr am selben Bus nutzen will, muss man beiden eine verschiedene Adresse geben.

Es gibt übrigens zwei konkurrierende Lesarten für die Adresse, was manchmal für Verwirrung und aufwendige Fehlversuche sorgt. Manche Datenblätter geben nur die 7-bit-Adresse ohne das Schreib/Lese-Bit an. Der PCF8574 hätte in dieser Lesart die Adresse 32 (20_{hex}). In Bascom dagegen besitzt der Chip eine Schreibadresse 64 ($40_{\text{hex}} = \&H40$) und eine Leseadresse 65 ($\&H41$). Um die Adressen der angeschlossenen ICs zu erfahren, kann ein kleines Bascom-Programm (**Listing 1**) helfen (Download unter [3]). Es klopft überall mal an und fragt, ob jemand antwortet. Nach einer Adressierung eines Bausteins enthält die Bascom-Systemvariable ERR (die man nicht mit Dim anmelden muss, weil Bascom sie schon deklariert hat) eine Eins, wenn kein Acknowledge kommt und eine Null, wenn ein Ack-Signal empfangen wurde. In letzterem Fall war die Adresse also gültig. Getestet werden alle geraden Adressen zwischen 2 und 254 (die entsprechenden ungeraden Adressen wären in Bascom die Leseadressen der gleichen ICs). Für die Schaltung in **Bild 2** werden wie erwartet die Adressen 64, 144, 160 und 162 gemeldet. Noch eine Tücke lauert hinter dem I2C-Bus-Protokoll: Jeder Busteilnehmer kann den Master ausbremsen, wenn er etwas mehr Zeit braucht.

zu acht Bausteine des gleichen Typs an einen Bus anschließen. Solch ein Portexpander stellt acht digitale Ausgänge zur Verfügung, die gewünschten Pegel an diesen Ausgängen werden als Bits innerhalb eines Datenbytes an den Chip übertragen. Der Portexpander besitzt die Grundadresse $40_{\text{hex}} = 64$. Weitere typische I2C-Bausteine der ersten Stunde von Philips sind diese:

- AD-Wandler PCF8591: $90_{\text{hex}} = 144$

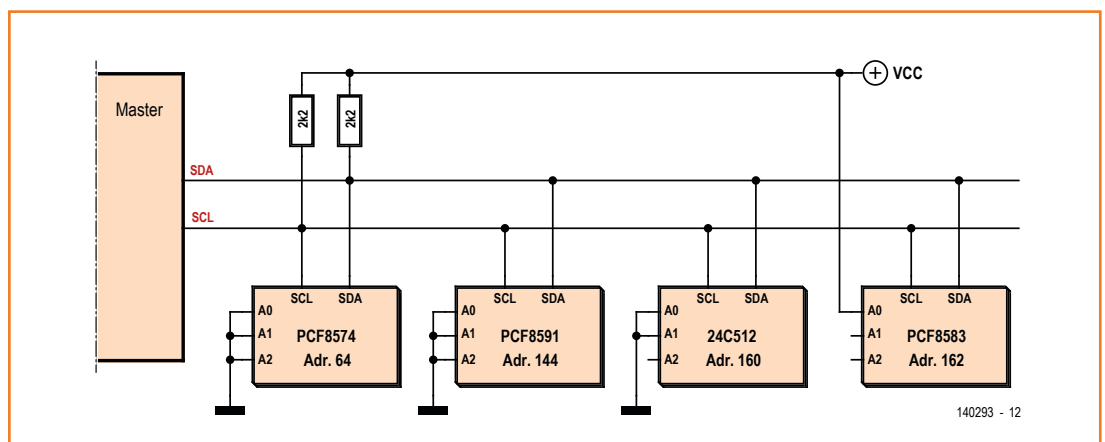


Bild 2.
Ein Master und vier Slaves.

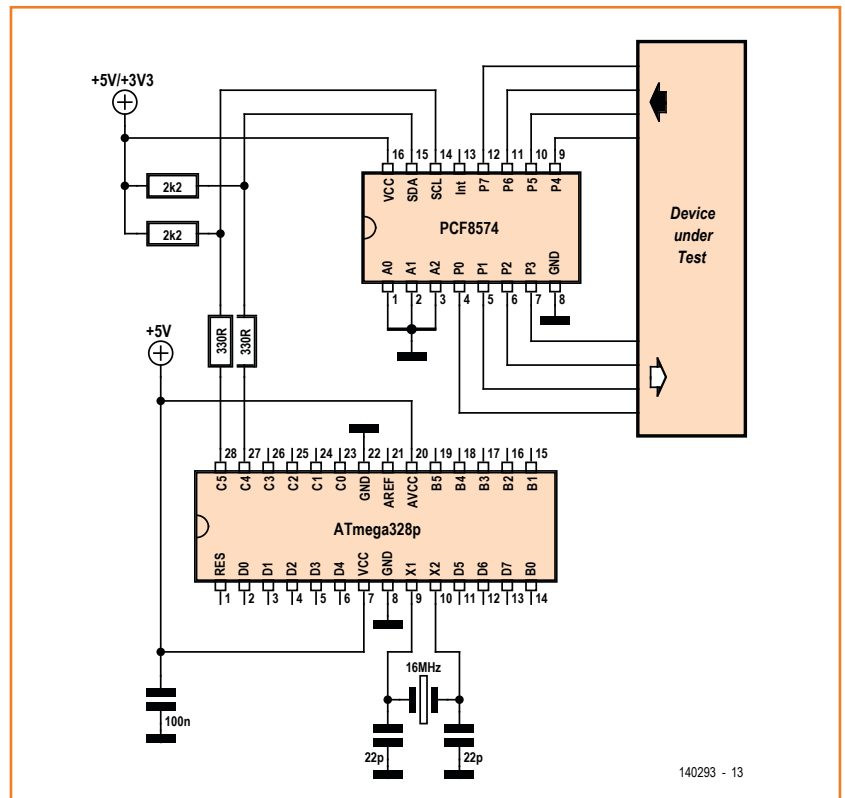
Er zieht dazu die Taktleitung SCL herunter und zwingt den Master damit abzuwarten, bis die Leitung wieder frei ist. Bascom hält sich pflichtgemäß an diese Konvention. Das bedeutet aber zugleich, dass ein Programm mit I2C-Bus hängt, wenn keine Pullups angeschlossen sind. Man ist es ja vielleicht von anderen Anwendungen gewohnt, dass man die Software auch mal ohne angeschlossene Hardware testen kann. Deshalb hat man vielleicht einmal den Flachbandstecker zu einem GnuBlin-Board noch nicht angeschlossen und will erstmal mit dem Oszilloskop nachsehen, ob bei laufendem Programm überhaupt Signale an SCL und SDA ankommen. Reingefallen, denn die I2C-Pullups befinden sich auf dem GnuBlin-Board. Weil ohne dieses Board dann auch kein Pullup vorhanden ist, bleibt alles stehen, und es gibt dementsprechend auch keine Signale.

Der Portexpander PCF8574

Der PCF8574 ist ein Portbaustein mit acht bidirektionalen Ausgängen. Es gibt in diesem Fall keine Datenrichtungs-Umschaltung, sondern die Ports besitzen interne Pullup-Widerstände, die im Ruhezustand einen High-Pegel vorgeben. Nach einem Neustart würde man also erstmal den Portzustand 255 (11111111) lesen. Nur wenn Leitungen von außen aktiv gegen GND gezogen werden, kann man hier Nullen sehen. Es ist möglich, einige der Leitungen als Aus- und einige der Leitungen als Eingänge zu nutzen; alle Leseleitungen müssen zuerst wie ein Ausgang in den High-Zustand versetzt werden.

Bild 3 zeigt die Busverbindung und eine mögliche Anwendung des Portexpanders als digitales Testgerät. Man kann den Portexpander wahlweise mit 3,3 V oder mit 5 V versorgen, entsprechend der Betriebsspannung des Testobjekts. Eine beliebige digitale Schaltung mit vier Eingängen wird über die Ausgänge P0 bis P3 angesteuert. Hier könnte man z.B. alle Pins einmal hochzählen und hätte dann jede erdenkliche Kombination an den Eingängen der Schaltung. Genauso lassen sich auch Kabel und Steckverbindungen testen. Jede Unterbrechung und jeder Kurzschluss würde entdeckt.

Listing 2 zeigt ein Beispiel für diese gemischte Verwendung von Ein- und Ausgängen. P0 bis P3 sind die vier Ausgänge und werden laufend hochgezählt. Die Leitungen P4 bis P7 dienen als Eingänge und müssen dazu während des Schreibvorgangs hochgesetzt werden (Or &B11110000). Der Chip wird zweimal adressiert, einmal in Schreibrichtung (Adresse 64) und einmal in Leserichtung



(Adresse 65). Das Hochzählen der vier Ausgänge ist die einfachste Variante eines Testgeräts. Je nach Testobjekt kann auch ein ganz anderer Prüfablauf nötig werden, der dann eine andere Prüfsoftware benötigt.

Bild 3.
Anwendung des
Portexpanders PCF8574.

16-Bit I/O-Port PCA9555

Manchmal sind acht zusätzliche Leitungen nicht genug. Das GnuBlin-Portexpander-Modul [2] bietet 16 I/O-Ports und verwendet dazu den PCA9555 von NXP. Das Board kann direkt an den EEC-Anschluss auf dem Extension-Shield angesteckt werden. Außerdem gibt es noch die GnuBlin-Relaiskarte, ebenfalls mit diesem Chip. NXP ist das Nachfolgerunternehmen von Philips, und der PCA9555 ist der legitime Nachfolger des PCF8574. Und deshalb hat er auch dieselbe Busadresse 64 (&H40). Diese Art des Adressen-Recyclings macht Sinn, denn der Adressraum ist ja sehr begrenzt. Und man kann mit Recht sagen, wer braucht noch einen PCF8574, wenn er einen PCA9555 hat. Denn der neue Baustein bietet außer der doppelten Anzahl der Portleitungen auch noch weitere Funktionen wie eine Datenrichtungsumschaltung oder eine Invertierung der gelesenen Portzustände.

Bild 4 zeigt den Anschluss des Bausteins über den

Listing 2. Portzugriffe beim PCF8574.

```

'-----
'UNO_I2C2.BAS input/output PCF8574
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim N As Byte
Dim D As Byte
...
Do
  For N = 0 To 15
    I2cstart
    I2cwbyte 64      'write
    D = N Or &B11110000
    I2cwbyte D
    I2cstop
    Print N;

    Print " ";
    Locate 2 , 1
    Lcd N
    Lcd " "

    I2cstart
    I2cwbyte 65      'read
    I2crbyte D , Nack
    I2cstop
    Shift D , Right , 4
    Print D
    Locate 2 , 5
    Lcd D
    Lcd " "
    Waitms 100
  Next N
Loop

```

EEC-Verbinder auf dem Extension-Shield. Über ein Flachbandkabel werden die Betriebsspannung von 3,3 V und die I2C-Busleitungen angeschlossen. Auf dem GnuBlin-Board sind die beiden Pullup-Widerstände über zwei Jumper zugeschaltet. Sie müssen gesetzt sein, wenn nur ein Board angeschlossen ist. Verwendet man aber mehrere Boards, die man

zum Beispiel über die GnuBlin-Verteilerplatine [2] anschließen kann, dann dürfen die Pullups nur auf einem der Boards eingeschaltet sein. Die Verwendung mehrerer Portexpander-Module ist möglich, weil auf der Platine auch die Adressleitungen A0 bis A2 mit Jumpers versehen sind. Im Grundzustand liegen alle drei Leitungen an GND. Man hat

Listing 3. Verwendung des Portexpanders PCA9555.

```

'-----
'UNO_LCD1.BAS input/output PCA9555
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim N As Byte
Dim D As Byte
...
I2cstart
I2cwbyte 64      'GnuBlin port expander
I2cwbyte 6
I2cwbyte 0      'Port 0 output
I2cwbyte 255    'Port 1 input
I2cstop

Do
  For N = 0 To 255
    I2cstart
    I2cwbyte 64      'write
    I2cwbyte 2

    I2cwbyte N
    I2cstop
    Print N;
    Print " ";
    Locate 2 , 1
    Lcd N
    Lcd " "

    I2cstart
    I2cwbyte 64      'write
    I2cwbyte 1
    I2cstart
    I2cwbyte 65      'read
    I2crbyte D , Nack
    I2cstop
    Print D
    Locate 2 , 5
    Lcd D
    Lcd " "
    Waitms 100
  Next N
Loop

```

dann die Bus-Adresse 64. Insgesamt lassen sich aber acht Platinen mit den Adressen 64 bis 72 anschließen. Mit 16 Portleitungen pro Platine käme man dann auf insgesamt 128 Portpins.

Listing 3 zeigt eine Anwendung des PCA9555, die ebenfalls wieder zum Testen anderer Schaltungen genutzt werden kann. Wie im vorigen Beispiel werden die Portpins aufgeteilt, eine Hälfte für die Ausgabe, die andere Hälfte für die Eingabe. Weil die ganze Sache aber hier etwas komplexer ist, muss nach der Adresse immer noch ein Kommandobyte gesendet werden [4]. Das Kommando entspricht einer Registeradresse im Chip. Danach darf man ein oder zwei Bytes senden oder empfangen. Will man z.B. Port 0 (mit acht Portpins) und Port 1 (mit den anderen acht Portpins) konfigurieren, sendet man nach der Adresse das Kommando 6 und schreibt dann zwei Datenbytes, die in Register 6 und Register 7 landen. Nullen in diesen Bytes stehen dafür, dass man die entsprechenden Portpins als Ausgang konfigurieren möchte. Einsen stehen für Eingänge. Bei uns werden alle Leitungen des Port 0 zu Ausgängen und die des Port 1 zu Eingängen. Eingänge heißt übrigens, dass im Chip ein hochohmiger Pullup mit etwa 100 k Ω verwendet wird, im Leerlauf also Einsen gelesen werden. Dies sind die gültigen Kommando-Bytes:

- 0 Input Port 0
- 1 Input Port 1
- 2 Output Port 0
- 3 Output Port 1

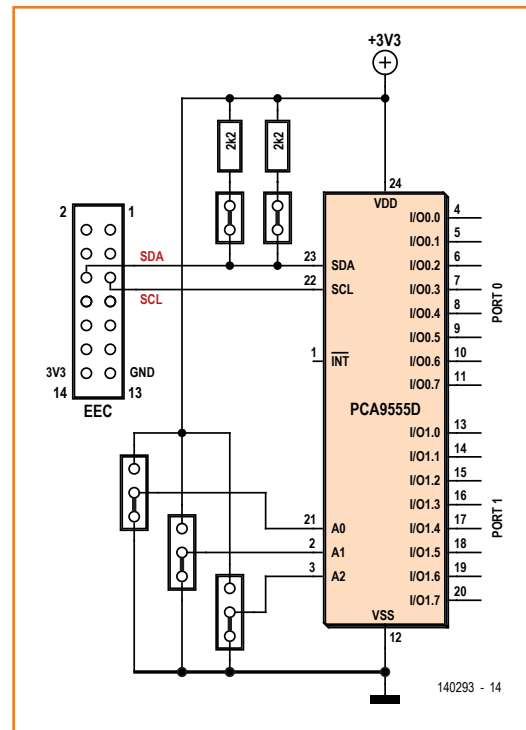


Bild 4.
Der PCA9555 auf dem
Gnublin-Board.

- 4 Polarity Inversion Port 0
- 5 Polarity Inversion Port 1
- 6 Configuration Port 0
- 7 Configuration Port 1

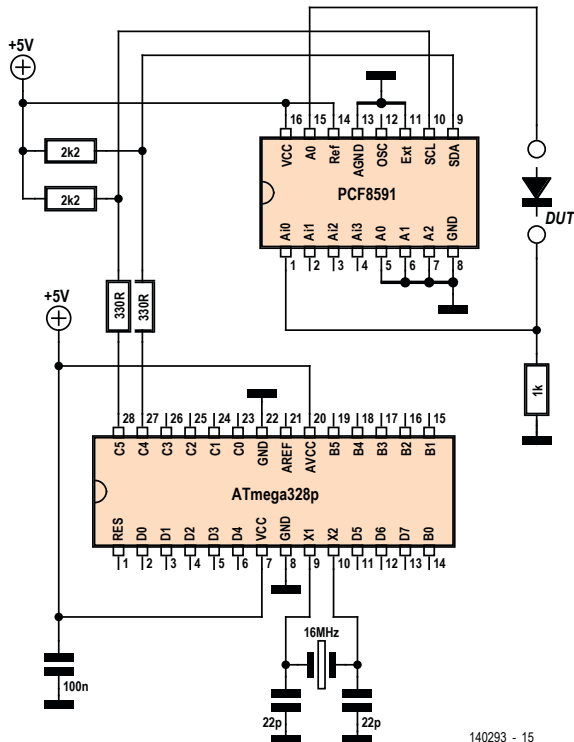
In der eigentlichen Datenschleife des Programms werden laufend die Kommandos 2 (Output Port 0) und 1 (Input Port 1) verwendet. Für jedes Kommando muss der Baustein neu adressiert werden.

Weitere interessante I2C-Bus-Bausteine

Wer regelmäßig Elektor liest, stolpert laufend über interessante ICs mit I2C-Interface. Da findet sich so manche Anregung für künftige Projekte. Einige wichtige Typen sind:

- I2C-EEPROMs bis zu 64 kB (wie zum Beispiel der 24C512). Damit lassen sich Datenlogger und vieles mehr bauen.
- Der CY27EE16 ist ein über I2C programmierbarer Quarzgenerator, der im Software Defined Radio von Elektor verwendet wurde. Die Ansteuerung über einen Mikrocontroller eröffnet neue Möglichkeiten.
- Der SI4735 ist ein kompletter AM/FM-Empfänger und wurde bereits mehrfach in Elektor zusammen mit Bascom verwendet. Mit wenigen I2C-Kommandos lässt sich jede Frequenz einstellen.
- AD- und DA-Wandler mit hoher Auflösung verfügen oft über einen I2C-Anschluss. Ein Beispiel ist der kürzlich in Elektor vorgestellte 16-bit-AD-Wandler ADS1115.

Wer sich tiefer in die Materie einarbeitet, kann sogar sein eigenes I2C-Bus-IC bauen! Dazu gibt es eine Bascom-Library, mit deren Hilfe ein I2C-Slave programmiert werden kann. Das ist aber etwas anspruchsvoller als bei einem Master, weil der Slave mit der hohen Geschwindigkeit umgehen können muss, die ein Master vorgibt.



Und auch zum Auslesen eines Ports muss der Chip neu adressiert werden, diesmal in Leserichtung, also mit gesetztem Lesebit (Adresse 65). Wer genau hinsieht, vermisst vielleicht ein I2cstop. Hier handelt es sich um einen „Repeated Start“ ohne vorherige Stopp-Bedingung, weil beide Zugriffe immer zusammengehören: Schreiben des zu lesenden Registers und das eigentliche Lesen.

Analoge Ein-/Ausgabe beim PCF8591

Der PCF8591 ist ein 8-bit-AD-Wandler mit vier Eingängen und gleichzeitig ein 8-bit-DA-Wandler mit einem Ausgang. Den AD-Wandler allein wird man nur selten brauchen, weil der Uno ja selbst genügend analoge Eingänge hat, die noch dazu über eine bessere Auflösung verfügen. Aber ein echter DA-Wandler kann mal nützlich sein. Anders als ein PWM-Ausgang liefert er eine echte Gleichspannung. Damit kann man einen einfachen Diodentester bauen, der die Durchlassspannung bei einem definierten Strom bestimmt (**Bild 5**). Damit auch LEDs mit hoher Durchlassspannung gemessen werden können, wird die Schaltung mit 5 V betrieben. Der Baustein hat ein Steuerregister, dessen Inhalt gleich nach der Busadresse gesendet werden muss. Mit einem Steuerbyte 64 schaltet man den DA-Wandler ein und legt den Eingangskanal 0 fest. Das folgende Byte gelangt dann in

Bild 5.
Diodentester mit dem
PCF8591.

Listing 4. Diodentester mit dem PCF8591.

```

'-----
'UNO_I2C4.BAS AD/DA PCF8591
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim N As Byte
Dim D As Byte
Dim U As Word
...
N = 0
Do
    I2cstart
    I2cwrite 144          'write
    I2cwrite 64           'DA enable
    I2cwrite N
    Print N;
    Print " ";
    Locate 2 , 1
    Lcd N
    Lcd "  "

    I2cstart
    I2cwrite 145          'read
    I2cwrite D , Nack    'Ain0
    I2cstop
    Print D
    Locate 2 , 5
    Lcd D
    Lcd "  "
    Waitms 100
    N = N + 1
    If D >= 50 Then Exit Do
Loop
Cls
Locate 1 , 1
Lcd " 1 mA"
U = N - D
U = U * 20
Locate 2 , 2
Lcd U
Lcd " mV"
End

```

das DA-Register und wird als Spannung zwischen 0 V (0) und 5 V (255) ausgegeben. Mit einer Auflösung von 8 bit erhält man Spannungsschritte von ca. 20 mV. Das Beispielprogramm nach **Listing 4** erzeugt eine aufsteigende Spannungsrampe. Gleichzeitig wird die Spannung am 1-k Ω -Shunt gemessen. Dazu muss der PCF8591 in Leserichtung (Adresse 145) erneut adressiert werden. Das gelesene Byte steht für die Spannung an Ai0. Ein gelesener Wert von 50 entspricht einer Spannung von 1 V und damit einem Diodenstrom von 1 mA. An dieser Stelle wird die Rampe gestoppt. Die Dioden-Durchlassspannung kann nun aus der Differenz der Ausgangsspannung und der Eingangsspannung berechnet werden und wird dann in Millivolt angezeigt. Am Ende der Messung wurde für eine blaugrüne LED folgende Ausgabe sichtbar:

1 mA
2920 mV

Kleiner Ausblick

Das war die letzte Folge unseres Mikrocontroller-Kurses, doch in lockerer Folge werden noch weitere Bascom-Anwendungen für den Arduino Uno und das Elektor Extension-Shield hinzukommen.

Wir hoffen, dass wir Sie zu eigenen Programmen anregen konnten. Wenn ja, dann lassen Sie doch andere Leser an Ihrer Entwicklung teilhaben unter www.elektor-labs.com! Ausdrücklich willkommen sind dort auch kleinere Projekte oder Anwendungen, die sich noch in der Entwicklung befinden.

(140293)

Weblinks

[1] www.elektor-magazine.de/140009

[2] www.elektor.de/tools/gnublin

[3] www.elektor-magazine.de/140293

[4] www.nxp.com/documents/data_sheet/PCA9555.pdf

Anzeige



progressive
in early
involvement
services

exhibitor at the electronica fair
www.electronica.de • booth A2.409



Vlaktodern 10
3247 CP Dirksland

T +31 187 602 744
F +31 187 603 497

I www.tbp.eu
E info@tbp.nl



Recycling für Klopfensoren

Von **J. T. van Es**
(NL)

Ein Kollege, der vor Jahren nebenbei Automotoren tunte, war auf der Suche nach einem Verfahren, relevante Motorgeräusche aufzuzeichnen und zu analysieren. „Weshalb wertest du nicht die Signale der Klopfensoren aus“, fragte ich, „das Motormanagement jedes neuzeitlichen PKW macht davon Gebrauch.“ Ich schaute in ein ungläubiges Gesicht, und mein Kollege fragte zweifelnd: „Geht denn das?“ Ich antwortete: „Das weiß ich nicht, aber ich werde es herausfinden!“

An jenem Tag habe ich mir von einem Schrottplatz mehrere Klopfensoren unterschiedlicher Bauart besorgt. Zuerst habe ich ein Exemplar zerlegt, ich war gespannt, was sich unter der schwarzen Hülle verbarg. Doch was ich dort vorfand (**Bild 1**), war kaum sensationell: Eine ringförmige Scheibe aus

einem piezo-elektrischen Material, eingespannt zwischen mehreren Metallscheiben. In der kreisförmigen Öffnung in der Ringmitte steckte eine M8-Gewindeschraube, sie stellte die mechanische Verbindung mit dem Motorblock her. Ich habe die Kapazität dieses „Kondensators“ gemessen, sie lag bei 1000 pF. Damals konnte ich kein Datenblatt für den Klopfsensor auftreiben, heute ist das Datenblatt problemlos online zugänglich [1]. Einen Eindruck vom Ausgangssignal erhielt ich, als ich den Klopfsensor auf den Konus eines aus-rangierten Basslautsprechers klebte. Schnell fand ich heraus, dass sich der Sensor ähnlich einem Mikrofon verhielt. Das war zwar schon recht aufschlussreich, doch über die Empfindlichkeit und die Größe der zu messenden Signale sagte diese Erkenntnis noch nichts.

In **Bild 2** ist der Verstärker dargestellt, den ich konstruiert habe, um die Eigenschaften quantitativ zu ergründen. Da sich der Sensor wie ein Kondensator verhielt, habe ich einen Ladungsverstärker gebaut. Die Verstärkung der FET-Opamps beträgt zusammen ungefähr 100, die Bandbreite liegt bei 70 Hz...36 kHz (-3 dB). In der Schaltung sind V1 und C1 die Ersatzgrößen stellvertretend für den Sensor. Widerstand R16 dient lediglich der Simulation, ohne R16 wäre das Potential zwischen C1 und C7 unbestimmt. C7 hält den Leckwiderstand des Sensors vom Opamp-Eingang fern. Die Verstärkung der ersten Stufe hängt von C1 und C2 ab, in der zweiten Stufe sind R14 und R15 maßgebend. Die untere Grenze des Übertragungsbereichs wird von C2 und R1 bestimmt. Der Wert von R1 muss eigentlich 33 M betragen, doch dieser Wert war nicht zur Hand. Deshalb habe ich am Opamp-Ausgang den Spannungsteiler R17/R18 hinzugefügt, er schwächt das Signal auf etwa 1/3 ab. R7 und C1 begrenzen den Übertragungsbereich bei den hohen Frequenzen. Die Messungen am Verstärker bestätigten die Ergebnisse der Simulation (Diagramm in **Bild 3**).

Für den Einsatz mit dem Klopfsensor war die Empfindlichkeit um Größenordnungen zu hoch. Weil sich der Verstärker als überflüssig erwies, hatte ich ihn kurzerhand in meinem Bastelschrank abgelegt. Heute, im Jahr 2014, als mir der Verstärker in die



Bild 1.
In diesem Klopfsensor befindet sich eine Scheibe aus piezo-elektrischem Material, eingebettet zwischen metallischen Scheiben.

Klopfsensor

Klopf- oder Klingelsensoren werden für die Steuerung von Verbrennungsmotoren eingesetzt, um die unkontrollierte Selbstzündung des Kraftstoff-Luft-Gemischs zu verhindern. Verglichen mit dem regulären Verbrennungsprozess sind die Schwingungen bei „klopfendem“ Motor in bestimmten Frequenzbereichen stärker ausgeprägt. Das Sensorsignal wird vom Motormanagement-System (ECU, *Electronic Control Unit*) verarbeitet und ausgewertet. Wenn der Motor klopft, wird der Zündwinkel schrittweise (meistens in 3°-Schritten) zurückgefahren. Nach Aussetzen des Klopfens erhöht das System den Zündwinkel in Schritten von 0,5°, bis wieder der (klopfreie) Normalbetrieb erreicht ist.

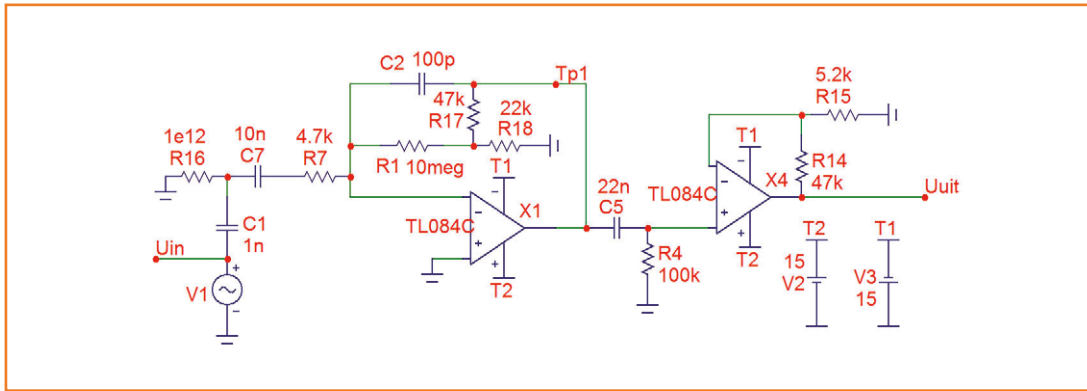


Bild 2.
Die zweistufige Opamp-Schaltung verstärkt das Klopfsensor-Signal insgesamt um den Faktor 100.

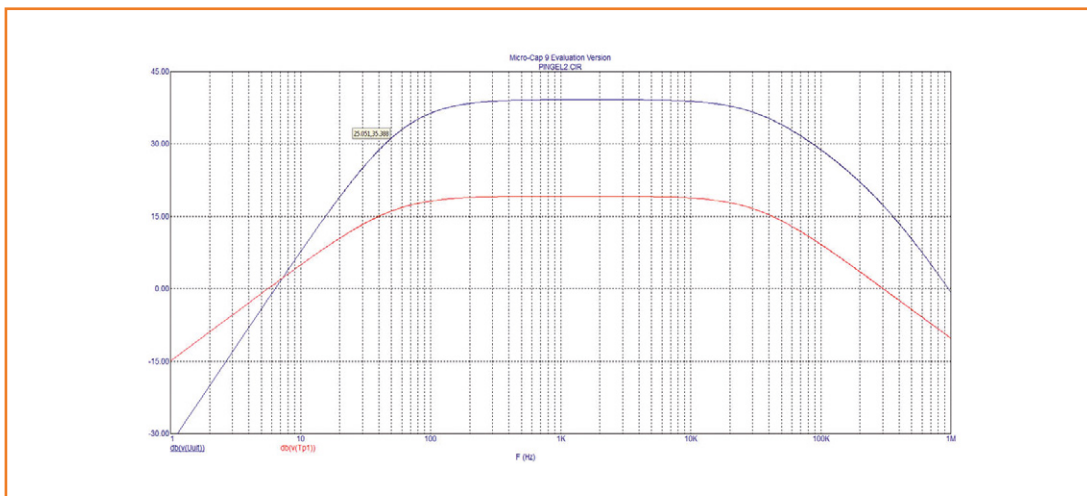


Bild 3.
Die Simulation führt zu diesen Frequenz-Charakteristiken (rot = Ausgang erster Opamp, blau = Ausgang zweiter Opamp).

Hände fiel, habe ich einige weitere Messungen vorgenommen. Die Rauschspannung am Ausgang beträgt 240 μV , umgerechnet liegen folglich 2,4 μV am Eingang. Elektrisch abgeschirmt durch Alufolie ist der Sensor jetzt mit meinem Arbeitstisch verschraubt. Die Empfindlichkeit meines Oszilloskops steht auf 10 mV/div. Leichtes Berühren des Tisches löst auf dem Oszilloskopschirm heftige Reaktionen aus. Ein aus zwei Zentimeter Höhe auf die Tischplatte fallendes Streichholz hat eine Spitze von 40 mV zur Folge, und sogar am Tisch gesprochene Sprache ist erkennbar. Für einen soliden, robusten Sensor, der starke akustische Signale in der Welt der Automobile misst, ist dies eine erstaunliche Empfindlichkeit. In jedem einigermaßen modernen Auto ist mindestens ein Exemplar verbaut. Natürlich gilt dies auch für die Fahrzeuge, die an ihrem Lebensende auf dem Schrottplatz landen. Ein Eldorado voller Bauelemente, die auf ein zweites Leben warten!
Wie könnte es aussehen, das zweite Leben? Ein Erschütterungsmelder, ein Stethoskop? Mein

Klopfsensor und der konstruierte Verstärker sind aus Zeitmangel vorläufig wieder an ihren Platz im Bastelschrank zurückgekehrt.

(140324)gd

Weblink

www.bosch-motorsport.de/media/catalog_resources/Knock_Sensor_KS-R_Datasheet_51_en_2779074187pdf.pdf

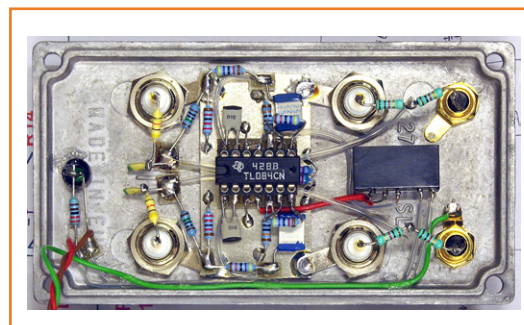


Bild 4.
Der vom Autor aufgebaute Verstärker ist für eine optimale Abschirmung in einem Metallgehäuse untergebracht.

GERADE WENN'S MAL HEISS HERGEHT.

LEITERPLATTEN AUS ALUMINIUM ONLINE BESTELLEN.



LEITON
RECHNEN SIE MIT BESTEM SERVICE

Starke Hitzeeinwirkung ist eine echte Herausforderung – z. B. wenn wichtige Leistungsbauteile hohen Temperaturen ausgesetzt sind. Unsere Aluminium-Platinen überzeugen durch **hohe Hitzeresistenz** sowie einen **Wärmeleitwert von bis zu 3,0 W/mK in der Premiumvariante „Polytherm TC-Lam 3.0“**. Ihre Vorteile: **Kostenreduktion** durch höhere Lebensdauer und Zuverlässigkeit, **Platzersparnis** dank Integration der Kühlkörper sowie **Performancesteigerung** durch höhere Leistungsdichte innerhalb der Anwendung. Sie möchten mehr darüber wissen? Wir bieten persönliche Beratung am Telefon, einen kompetenten Außendienst und Angebote auch per E-Mail in Windeseile. Sie können bei Leiton immer mit bestem Service rechnen.

www.leiton.de

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

Schaeffer **AG**



Gratis
Frontplatten
Designer

SIE DESIGNEN – WIR FERTIGEN

Frontplatten in Profiqualität

Ab einem Stück und zu einem fairen Preis!
Einfach unseren kostenlosen Frontplatten Designer auf www.schaeffer-ag.de herunterladen, Frontplatte entwerfen und direkt bestellen.

www.schaeffer-ag.de

The new EAGLE has landed!

Version 7

ab sofort erhältlich

Mehr Informationen auf www.cadsoft.de



...und was haben *Sie* zwischen dem 11. & 14. November 2014 vor?

CS
EAGLE



Allesmelder

Rauchmelder als Universaldetektor

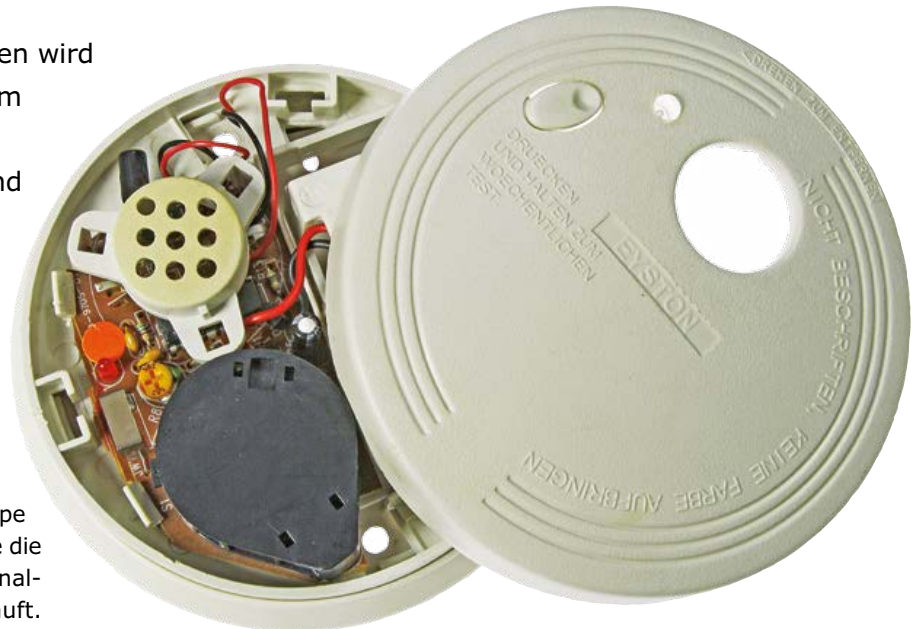
Mit nur einigen externen Komponenten wird ein preiswerter Rauchmelder zu einem universellen Allesmelder. Es können praktisch unbegrenzt Schaltungen und Sensoren gleichzeitig angeschlossen werden. Der Rauchmelder bleibt dabei voll funktionsfähig.

Von **Jürgen Friker (D)**

Ich habe im Keller ein Loch, in dem eine Pumpe einströmendes Grundwasser abpumpt. Sollte die Pumpe einmal ausfallen, soll mich eine Signaleinrichtung warnen, bevor der Keller vollläuft. Also muss im Pumpensumpf ein Wassermelder angebracht werden, der aus einem Geber (Sensor) und einem akustischen Melder (Krachmacher) besteht.

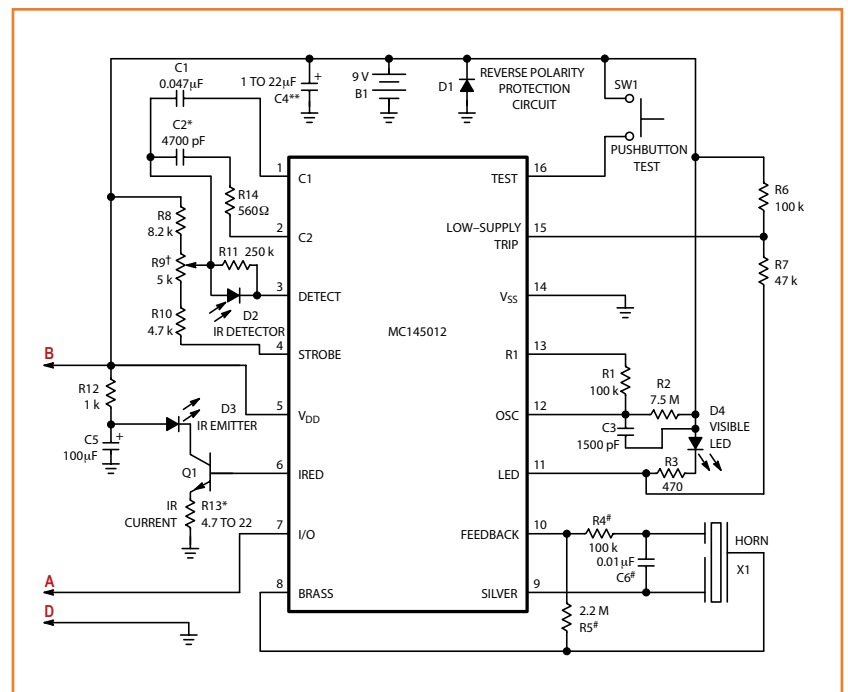
Krachmacher

Ich habe eine Reihe von Signalgebern ausprobiert. Netzbetriebene Hupen zum Beispiel sind zwar laut, versagen aber bei Stromausfall genau wie die Pumpe. Und wenn man nicht gerade direkt neben diesem Signalgeber steht, „überhört“ man den Dauerton meist ohnehin. Auch andere Lärm-erzeuger konnten mich nicht überzeugen, selbst mein Tuba spielender Bruder nicht, weil er sich schlicht geweigert hat, den Job eines Signalhorns dauerhaft zu übernehmen. Als eindeutig beste Wahl entpuppte sich ein stinknormaler Rauchmelder. Der Warnton ist sehr auffällig, das Gerät läuft ohne Netz im Batteriebetrieb und verfügt über eine eigene Batterieüberwachung. Zudem ist ein Rauchmelder viel billiger als besagte Hupe, ja selbst das für den Anschluss des Sensors erforderliche Pärchen aus Mini-DIN-Stecker und Buchse hat im Elektronikladen ungefähr so viel gekostet wie der ganze Rauchmelder. Alle Rauchmelder, die ich aufgemacht habe, arbeiten mit (teilweise verschiedenen) Freescale-Chips (früher Motorola), die mit Pin 7 einen Vernet-



zungsport besitzen (**Bild 1**). Über diesen Port werden mehrere Rauchmelder per Zweidrahtbus zu einer Gruppe verbunden. Löst ein Gerät Alarm aus, ertönt dieser Alarm dann in allen Rauchmeldern. Der

Bild 1.
Standard-Anschlussbelegung des Rauchmelder-ICs MC145012 laut Freescale.



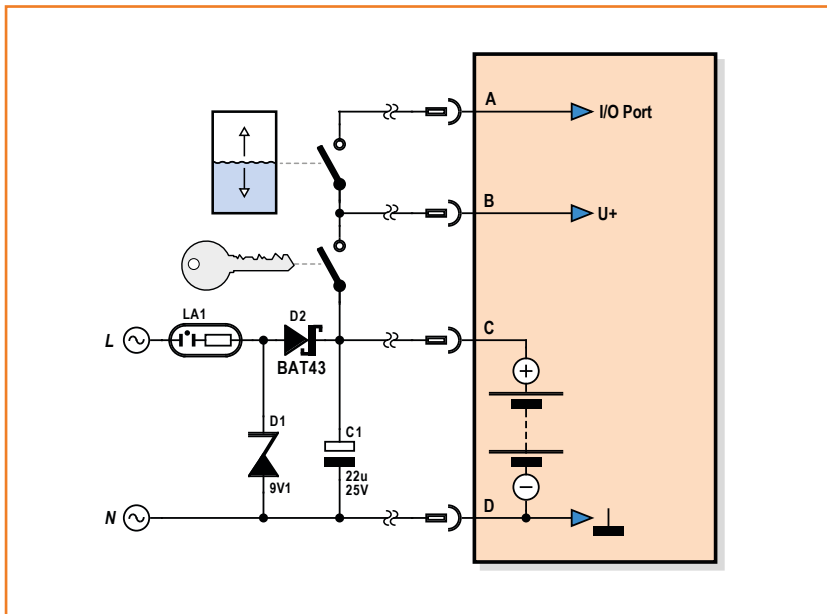


Bild 2.
Die „externe“ Schaltung besteht aus einem Schwimmerschalter, einem Schlüsselschalter und einer kleinen Netzschaltung mit Glühlampe.

Port wird bei Stand-alone-Rauchmeldern nicht gebraucht, ist aber natürlich dennoch nutzbar. Dabei ist keinerlei Eingriff auf der Rauchmelderseite (nicht einmal am Testknopf) erforderlich. Der Rauchmelder bleibt voll funktionsfähig.

Der IO-Pin 7 ist IC-intern mit einem Pulldown-Widerstand auf Masse gelegt. Er wird im Alarmfall High und zieht damit den ganzen Bus auf positives Betriebsspannungspotential. Am IO-Port sind Spannungen von -0,25 V bis VDD+10 V (bei einer 9-V-Blockbatterie also +19 V) erlaubt. Die Schaltschwelle liegt immer bei +3,2 V. Der MC145012 (und ähnliche Chips) besitzen auch aus Gründen der Störsicherheit eine interne Stromsenke. Interessant zu erwähnen ist, dass man am Verhalten der LED erkennen kann, ob ein Alarm an diesem Gerät ausgelöst wurde (LED D4 blinkt im Halbsekundentakt) oder ferngesteuert über den IO-Pin (D4 bleibt dunkel).

Keine gute Idee ist es übrigens, den Test-Pin 16 zur externen Alarmauslösung zu nutzen. Der Test-Pin ist nämlich nicht für längere Anschlussleitungen ausgelegt. Er setzt die Verstärkung des Messkreises auf ein Maximum, damit das immer vorhandene Streulicht in der Rauchkammer ausreicht, um einen Alarm sicher auszulösen. Das würde dann aber auch passieren, wenn über eine lange Sensorleitung irgendwelche Störungen zum IC gelangen. Außerdem löst er nur einen Test der Rauchmelderfunktion aus - und wenn der

Test negativ ausfällt, gibt es auch keinen Alarm.

Schwiegermutteralarm

Dem Rauchmelder ist es natürlich egal, ob jetzt Wasser, das Öffnen der Kühlschranktür oder die Annäherung der Schwiegermutter den Alarm auslöst, anschließen kann man praktisch alles, was in der Lage ist, den Anschluss „A“ in **Bild 2** auf eine (in genannten Grenzen) positive Spannung zu legen. Man kann sogar mehrere Sensoren über ein Wired-OR verknüpfen. Bleibt Anschluss „A“ dagegen offen oder liegt die angelegte Spannung unter 1,5 V, wird kein externer Alarm ausgelöst. Im Fall des Wassermelders habe ich einen Schwimmerschalter angeschlossen, es könnte aber eben auch ein Transistor oder eine komplexere Schaltung sein.

Ein Schwimmerschalter ist genau genommen gar kein Schwimmer-, sondern ein Neigungsschalter. Heute befindet sich im Schalter kein Quecksilber mehr, stattdessen rollt ökologisch korrekt eine kleine Stahlkugel auf einen Mikroschalter oder von selbigem hinunter. Ob der Schalter den Alarm auslösen kann, hängt also nicht direkt vom Wasserstand im Sumpf ab. Damit der Schalter beim richtigen Wasserstand (oberhalb der Pumpe, unterhalb der Überschwemmung) auslöst, braucht es ein zuverlässiges Widerlager. Das kann einfach ein Kabelbinder an einem Rohr sein oder ein Gegengewicht am Kabel. Hier lässt sich nach Herzenslust experimentieren und ausprobieren. Wichtig ist nur, dass das Gegengewicht deutlich schwerer ist als der Auftrieb des Schwimmers, dass weder Kabel noch Gewicht verrutschen können und das freie Ende zwischen Gewicht und Schwimmer nicht zu kurz ist (sonst spielt die Biegesteifigkeit des Kabels mit).

Die Schaltung des Rauchmelders selbst beruht auf einer Applikation von Motorola (Bild 1), die beim Muster auch exakt umgesetzt worden war (mit $R_{13} = 6,8 \text{ Ohm}$).

Ein freches Netzteil

In Bild 2 ist eine externe Ladeschaltung zu sehen, die einen NiMH-Akku im Rauchmelder immer im voll geladenen Zustand hält. Die Glühlampe LA1 ist ein E10-Signallämpchen mit integriertem Vorwiderstand (1,5...1,9 mA an 230 V). Dieser Ladestrom für den Akku kompensiert nur seine Selbstentladung, der Akku kann keinesfalls überladen werden. Das Lämpchen signalisiert auch,

dass Netzspannung vorhanden ist (der Alarm braucht kein Netz, die Pumpe aber schon). Die Z-Diode D1 entnimmt dem Wechselstromkreis eine kleine Gleichspannung. Da sie in der anderen Richtung (auch) leitet, glimmt die Lampe in voller „Stärke“. D2 arbeitet als Entladeschutzdiode, C1 ist der übliche Ladekondensator. Der Kondensator liegt parallel zu C4 auf der Detektorplatine, die dortige D1 schützt vor falsch herum eingelegten Batterien und stört ansonsten nicht.

C1 sollte keinesfalls vergrößert werden, da sonst die Batteriekontrollfunktion des Detektors gestört wird: Der Detektor zieht ungefähr alle 30 s einen Impuls von rund 8 ms Dauer für die Kontroll-LED D4 über den Vorwiderstand R3 und überwacht währenddessen über den Spannungsteiler R6/R7 die Batteriespannung (also ein Test unter Last). Bei zuviel Ladekapazität würde der Impuls weitgehend aus den Kondensatoren gespeist und der Detektor zu spät merken, wenn der Akku

nicht mehr genügend Energie für einen Alarm bereithielte.

So originell dies alles ist, so gefährlich ist es aber auch! Bei Sensoren des Typs „blanker Draht in Wasser“ muss sowieso auf diese Ladefunktion verzichtet werden, aber auch sonst ist, **da die ganze Schaltung Netzspannung führt, eine gute Isolation des Schlüsselschalters (s.u.) und des Sensors unabdingbar.**



Mit dem Schalter kann man den Wasseralarm abschalten, idealerweise mit einem Schlüssel, damit dies kein Unbefugter tut. Ich habe übrigens den Akku beziehungsweise die Batterie versuchsweise durch einen entsprechend fetten Kondensator ersetzt. Das funktionierte in Bereitschaft ganz gut - im Alarmfall ohne Netz ging dem Kondensator aber viel zu schnell die Puste aus!

(140184)

Anzeige

Workshop-DVD

Entwurf eines Röhrenverstärkers

NEU!

Wie entwirft man einen kompletten Röhrenverstärker mit nur einem aktiven, verstärkenden Element pro Kanal? Das war die Wettbewerbsaufgabe des European Triode Festivals, das 2012 in Berlin stattfand. Röhren-Koryphäe Menno van der Veen bespricht in diesem Seminar seine sorgfältig ausgearbeiteten Entwürfe. Er geht dabei auf häufig auftretende Probleme bei Röhren und Trafos ein. Wie viel Ausgangsleistung kann man erwarten? Wie kann man Verzerrungen, die durch den Ausgangsstrom entstehen, minimieren? Wie findet man den besten Kompromiss zwischen maximaler Leistung und der Lebensdauer bei minimalen Verzerrungen? Ist Feedback nötig und warum? Dieses Seminar gibt eine verkürzte, jedoch komplette Übersicht des einjährigen „TubeSociety“-Kurses wieder.



Inhalt:

- 90 Minuten Videomaterial
- Gesamte Präsentation als PDF (44 Seiten)
- Datenblätter von besprochenen Referenzen

ISBN 978-3-89576-284-0

€ 24,90
CHF 30,90

Weitere Infos & Bestellung unter www.elektor.de



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-503** von Circuit Design. Niedriger Stromverbrauch und Batteriebetrieb machen das Modul ideal für

Anwendungen, bei denen es auf Zuverlässigkeit bei großer Reichweite ankommt. **STD-503** 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.**

Besuchen Sie uns auf der **Electronica 2014!**
11.-14. Nov.
Stand A4.345

NEU!
Muster ab November 2014



STD-503

2,4 GHz - weltweit



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 · 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 · info@circuitdesign.de

Rpi Compute Module

CM + CM/IO = Super-RPi

Von **Tony Dixon**
(Großbritannien)

Der Raspberry Pi bietet Studenten und Programmier-Enthusiasten einen Zugang zu einer preisgünstigen und leistungsfähigen Embedded-Computing-Plattform. Auch viele kommerzielle Produkte nutzen den RPi. Die Raspberry Pi Foundation erweitert die Plattform nun. Das erste Ergebnis ist das Raspberry Pi Compute Module (CM) und ein (optionales) Host-Board, mit dem man leicht auf alle I/Os zugreifen kann.

Das CM ist die Einfachheit in Person. Es gibt nur zwei Chips, den Broadcom 2835-Controller mit 512 MB RAM wie beim Raspberry Pi Modell B und einen 4 GB großen eMMC-Flash-Speicher.

Das alles sieht aus wie ein DDR2-SO-DIMM eines Laptops, ein Platinchen mit 200 Leiterbahnan-schlüssen. Die interessantesten Eigenschaften des RPi Compute Moduls sind in **Tabelle 1** aufgelistet.

Der SO-DIMM-Footprint und der Verzicht auf andere Anschlüsse ermöglichen eine viel kleinere Platine als beim RPi. Dennoch gibt es viel mehr Möglichkeiten, nämlich den Zugriff auf fast alle Signale des Broadcom-2835-Controllers einschließlich 46 GPIOs, die zudem jeweils bis zu sechs alternative Funktionen besitzen (siehe **Tabelle 2**).

Bild 1 beweist, wie klein das CM im Vergleich zum RPi ausfällt.

I/O, I/O und nochmals I/O

Die Details der CM-GPIOs sind in **Tabelle 3** aufgeführt. Die Standard-Signale der Raspberry Pi Modelle A und B sind vorhanden, so dass schon vorhandene Erweiterungs-Boards weiter benutzt werden können. Die zusätzlichen GPIO-Signale und weitere Schnittstellen ermöglichen allerdings viel umfangreichere Steuerungs-Szenarien. Die Anschlussbelegung des CM-Boards ist in **Tabelle 4** aufgelistet. Die 200 Anschlüsse führen neben den GPIO-Signalen auch Ports für Standard-HDMI, TV-Out, USB, LCD DSI und CSI-Kamera wie beim Original-RPi. Darüber hinaus gibt es auch Anschlüsse für ein zweites DSI-LCD und eine zweite CSI-Kamera.

Das CM wird auf eine ebenfalls neue Platine namens Raspberry Pi Compute Module I/O-Board gesteckt. **Bild 2** zeigt, wie die Kombination aus I/O-Board und CM aussieht. Das CM benötigt

Bild 1. Der bekannte Raspberry Pi-Computer links und das neue Raspberry Pi Compute Module rechts.

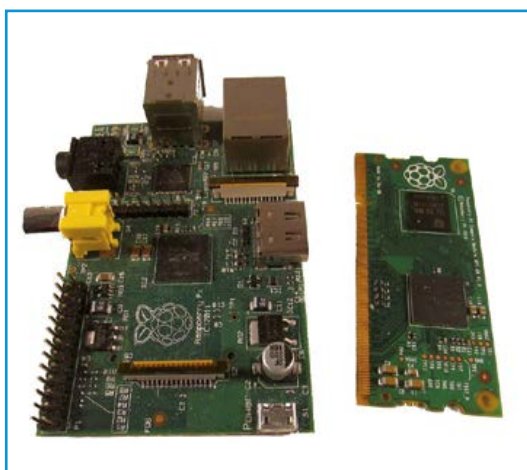


Tabelle 1. Highlights des RPi Compute Module (CM).	
System on Chip (SoC)	Broadcom BCM2835
Architektur	ARM1176JZF-S
Takt	700 MHz
GPU	VideoCore IV GPU
RAM	512 MB
Interface	HDMI, TV Out, USB, 2x DSI LCD, 2x CSI-Kamera
Abmessungen	67,6 mm x 30 mm (SODIMM)

Tabelle 2. Schnittstellen des RPi-CM.	
Number	Alternative Function
1	SD-Card-Interface
2	UART (einer mit CTS- und RTS-Signalen)
2	I ² C
2	SPI (eines mit 2 und das andere mit 3 Chips-Selects)
2	PWM-Ausgänge
1	I ² S/PCM
1	Zweiter Adress/Datenbus
3	Universell einsetzbare Taktausgänge
46	GPIOs

drei separate Spannungsversorgungen von 3,3 V, 1,8 V und 2,5 V, im Idealfall kleine Schaltregler. Wenn die TV-Out-Funktion nicht benötigt wird, kann man sich die 2,5-V-Versorgung sparen, was die Angelegenheit ein wenig vereinfacht. Weitere Details zum Design finden Sie auf der Github-Seite zum Raspberry Pi [1].

Das CM ist zum Booten vom eingebauten eMMC-Speicher eingerichtet, so dass es sich um eine eigenständige Einheit handelt.



Bild 2.
Eine solche Menge an I/O-, Versorgungs- und Steuerleitungen an 200 mikroskopisch kleinen Leiterbahnkontakten eines SO-DIMMs verlangen nach einem geeigneten „Motherboard“. Hier ist das CM in ein CM IO-Board gesteckt und stellt eine Art Super-RPi dar.

Tabelle 3. Funktionen der RPI-CM GPIO-Pins.						
GPIO ID	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
GPIO0	SDA0	SA5	<reserved>			
GPIO1	SCL0	SA4	<reserved>			
GPIO2	SDA1	SA3	<reserved>			
GPIO3	SCL1	SA2	<reserved>			
GPIO4	GPCLK0	SA1	<reserved>			ARM_TDI
GPIO5	GPCLK1	SA0	<reserved>			ARM_TDO
GPIO6	GPCLK2	SOE_N/SE	<reserved>			ARM_RTCK
GPIO7	SPI0_CE1_N	SWE_N/SRW_N	<reserved>			
GPIO8	SPI0_CE0_N	SD0	<reserved>			
GPIO9	SPI0_MISO	SD1	<reserved>			
GPIO10	SPI0_MOSI	SD2	<reserved>			
GPIO11	SPI0_SCLK	SD3	<reserved>			
GPIO12	PWM0	SD4	<reserved>			ARM_TMS
GPIO13	PWM1	SD5	<reserved>			ARM_TCK
GPIO14	TXD0	SD6	<reserved>			TXD1
GPIO15	RXD0	SD7	<reserved>			RXD1
GPIO16	<reserved>	SD8	<reserved>	CTS0	SPI1_CE2_N	CTS1
GPIO17	<reserved>	SD9	<reserved>	RTS0	SPI1_CE1_N	RTS1
GPIO18	PCM_CLK	SD10	<reserved>	BSCSL_SDA/MOSI	SPI1_CE0_N	PWM0
GPIO19	PCM_FS	SD11	<reserved>	BSCSL_SCL/SCLK	SPI1_MISO	PWM1
GPIO20	PCM_DIN	SD12	<reserved>	BSCSL/MISO	SPI1_MOSI	GPCLK0
GPIO21	PCM_DOUT	SD13	<reserved>	BSCSL/CE_N	SPI1_SCLK	GPCLK1
GPIO22	<reserved>	SD14	<reserved>	SD1_CLK	ARM_TRST	
GPIO23	<reserved>	SD15	<reserved>	SD1_CMD	ARM_RTCK	
GPIO24	<reserved>	SD16	<reserved>	SD1_DAT0	ARM_TDO	
GPIO25	<reserved>	SD17	<reserved>	SD1_DAT1	ARM_TCK	
GPIO26	<reserved>	<reserved>	<reserved>	SD1_DAT2	ARM_TDI	
GPIO27	<reserved>	<reserved>	<reserved>	SD1_DAT3	ARM_TMS	
GPIO28	SDA0	SA5	PCM_CLK	<reserved>		
GPIO29	SCL0	SA4	PCM_FS	<reserved>		
GPIO30	<reserved>	SA3	PCM_DIN	CTS0		CTS1
GPIO31	<reserved>	SA2	PCM_DOUT	RTS0		RTS1
GPIO32	GPCLK0	SA1	<reserved>	TXD0		TXD1
GPIO33	<reserved>	SA0	<reserved>	RXD0		RXD1
GPIO34	GPCLK0	SOE_N/SE	<reserved>	<reserved>		
GPIO35	SPI0_CE1_N	SWE_N/SRW_N	<reserved>			
GPIO36	SPI0_CE0_N	SD0	TXD0	<reserved>		
GPIO37	SPI0_MISO	SD1	RXD0	<reserved>		
GPIO38	SPI0_MOSI	SD2	RTS0	<reserved>		
GPIO39	SPI0_SCLK	SD3	CTS0	<reserved>		
GPIO40	PWM0	SD4		<reserved>	SPI2_MISO	TXD1
GPIO41	PWM1	SD5	<reserved>	<reserved>	SPI2_MOSI	RXD1
GPIO42	GPCLK1	SD6	<reserved>	<reserved>	SPI2_SCLK	RTS1
GPIO43	GPCLK2	SD7	<reserved>	<reserved>	SPI2_CE0_N	CTS1
GPIO44	GPCLK1	SDA0	SDA1	<reserved>	SPI2_CE1_N	
GPIO45	PWM1	SCL0	SCL1	<reserved>	SPI2_CE2_N	

Fazit

Das Raspberry Pi Compute Module und das dazu gehörende I/O-Host-Board sind eine schöne Ergänzung der Raspberry-Pi-Produktfamilie. Das CM ist gespickt mit GPIOs und anderen Schnittstellen. Wenn das RPi-Modell A/B für Ihr Projekt nicht leistungsfähig genug und/oder zu groß ist, könnte das CM ideal sein, um die perfekte Schaltung nach Ihren Vorstellungen zu entwickeln. Auch wenn Sie ein größeres Hardware-Projekt

im Sinn haben, können die zusätzlichen I/Os ein großer Vorteil oder gar ein Muss sein.

(140320)

Weblink

[1] RPi Compute Module Hardware Design Guide:
<https://github.com/raspberrypi/documentation/tree/master/hardware/computemodule>

Tabelle 4. Anschlüsse des RPi CM 200-Pin-S0-DIMMs.

GND	1	2	EMMC_DISABLE
GPIO0	3	4	Not Connected
GPIO1	5	6	Not Connected
GND	7	8	Not Connected
GPIO2	9	10	Not Connected
GPIO3	11	12	Not Connected
GND	13	14	Not Connected
GPIO4	15	16	Not Connected
GPIO5	17	18	Not Connected
GND	19	20	Not Connected
GPIO6	21	22	Not Connected
GPIO7	23	24	Not Connected
GND	25	26	GND
GPIO8	27	28	GPIO28
GPIO9	29	30	GPIO29
GND	31	32	GND
GPIO10	33	34	GPIO30
GPIO11	35	36	GPIO31
GND	37	38	GND
GPIO0-27_VREF	39	40	GPIO0-27_VREF
GPIO28-45_VREF	41	42	GPIO28-45_VREF
GND	43	44	GND
GPIO12	45	46	GPIO32
GPIO13	47	48	GPIO33
GND	49	50	GND
GPIO14	51	52	GPIO34
GPIO15	53	54	GPIO35
GND	55	56	GND
GPIO16	57	58	GPIO36
GPIO17	59	60	GPIO37
GND	61	62	GND
GPIO18	63	64	GPIO38
GPIO19	65	66	GPIO39
GND	67	68	GND
GPIO20	69	70	GPIO40
GPIO21	71	72	GPIO41
GND	73	74	GND
GPIO22	75	76	GPIO42
GPIO23	77	78	GPIO43
GND	79	80	GND
GPIO24	81	82	GPIO44
GPIO25	83	84	GPIO45
GND	85	86	GND
GPIO26	87	88	GPIO46_1V8
GPIO27	89	90	GPIO47_1V8
GND	91	92	GND
DSI0_DN1	93	94	DSI1_DP0
DSI0_DP1	95	96	DSI1_DN0
GND	97	98	GND
DSI0_DN0	99	100	DSI1_CP

DSI0_DP0	101	102	DSI1_CN
GND	103	104	GND
DSI0_CN	105	106	DSI1_DP3
DSI0_CP	107	108	DSI1_DN3
GND	109	110	GND
HDMI_CK_N	111	112	DSI1_DP2
HDMI_CK_P	113	114	DSI1_DN2
GND	115	116	GND
HDMI_D0_N	117	118	DSI1_DP1
HDMI_D0_P	119	120	DSI1_DN1
GND	121	122	GND
HDMI_D1_N	123	124	Not Connected
HDMI_D1_P	125	126	Not Connected
GND	127	128	Not Connected
HDMI_D2_N	129	130	Not Connected
HDMI_D2_P	131	132	Not Connected
GND	133	134	GND
CAM1_DP3	135	136	CAM0_DP0
CAM1_DN3	137	138	CAM0_DN0
GND	139	140	GND
CAM1_DP2	141	142	CAM0_CP
CAM1_DN2	143	144	CAM0_CN
GND	145	146	GND
CAM1_CP	147	148	CAM0_DP1
CAM1_CN	149	150	CAM0_DN1
GND	151	152	GND
CAM1_DP1	153	154	Not Connected
CAM1_DN1	155	156	Not Connected
GND	157	158	Not Connected
CAM1_DP0	159	160	Not Connected
CAM1_DN0	161	162	Not Connected
GND	163	164	GND
USB_DP	165	166	TVDAC
USB_DM	167	168	USB_OTGID
GND	169	170	GND
HDMI_CEC	171	172	VC_TRST
HDMI_SDA	173	174	VC_TDI
HDMI_SCL	175	176	VC_TMS
RUN	177	178	VC_TDO
VDD_CORE	179	180	VC_TCK
GND	181	182	GND
1V8	183	184	1V8
1V8	185	186	1V8
GND	187	188	GND
VDAC	189	190	VDAC
3V3	191	192	3V3
3V3	193	194	3V3
GND	195	196	GND
VBAT	197	198	VBAT
VBAT	199	200	VBAT

Professionelle Hard- & Software zum Sonderpreis!

Exklusiv für Studenten!

Als neuer Vertriebspartner von National Instruments bietet Elektor ab sofort die Produkte der NI-Plattform für Ausbildung und Lehre für Studenten und schulische Einrichtungen an. Diese edukative Plattform vereint Hardware, Software und Unterrichtsmaterial, um Schülern und Studenten ein attraktives und inspirierendes Lernumfeld zu ermöglichen.

LabVIEW

Mit der Systemdesignsoftware *LabVIEW* können Studenten praxisorientiert anhand von Projekten und Systemen in einer einzigen Umgebung lernen und sich so Fähigkeiten und Verfahrensweisen aneignen, die im späteren Berufsleben unschätzbar sind.



Circuit Design Suite

Die *Circuit Design Suite* umfasst *Multisim* und *Ultiboard* und ist eine vollständige Plattform für Entwurf, Simulation und Validierung von Schaltplänen sowie den Leiterplattenentwurf. Die Suite verfügt über Funktionen, die speziell auf die Anforderungen von Studenten zugeschnitten sind, die bei der Entwicklung von elektronischen Konzepten hilfreich sind.



myDAQ

Bei *myDAQ* handelt es sich um ein kostengünstiges Datenerfassungsgerät, das überall und jederzeit Messungen und Analysen physikalischer Signale ermöglicht. *myDAQ* ist kompakt und portabel, sodass Studenten auch außerhalb des Labors und unter Einsatz branchenüblicher Werkzeuge und Methoden praktische Erfahrungen sammeln können.



Studentenversionen:

- Der Preis von **LabVIEW** und **Circuit Design Suite** beträgt jeweils nur 23,95 Euro.
- Das **myDAQ Education-Kit** bestehend aus *myDAQ* + 3 Software-DVDs (*LabVIEW*, *Circuit Design Suite* und *DIAdem*) kostet nur 174,95 Euro.

Jetzt bestellen unter www.elektor.de/ni-plattform!

Arduino mit Atmel Studio

Arduino-Due-Anwendungen debuggen

Von **Wolfram Pioch** (D)



Wer bisher Mikrocontroller der AT(x)mega-Reihe mit Atmel Studio programmiert hat und die Vorteile eines Debuggers kennt, wird diesen Komfort beim Umstieg auf den ARM-Cortex-M3-Controller des Arduino Due und die Arduino-IDE schmerzlich vermissen. Diesem Manko wird nun abgeholfen.

So schön es ist, mit den vielen verfügbaren und lauffähigen Arduino-Beispiel-Sketchen zu schnellen Ergebnissen zu kommen, so fallen bei ernsthafter

Beschäftigung mit der Kombination von ARM & Arduino vor allem folgende Nachteile ins Gewicht:

- Das Übersetzen des Programms in der Arduino-IDE dauert zu lange.
- Das Uploaden des Programms dauert zu lange.
- Ein echter Hardware-Debugger fehlt.

Bild 1.
Auswahl von aktueller
Arduino-Version, Board und
Programmierschnittstelle in
der Toolbar.

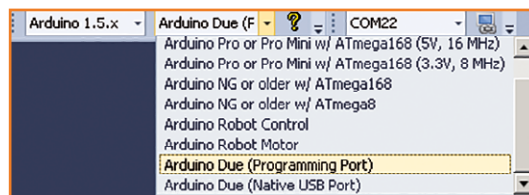
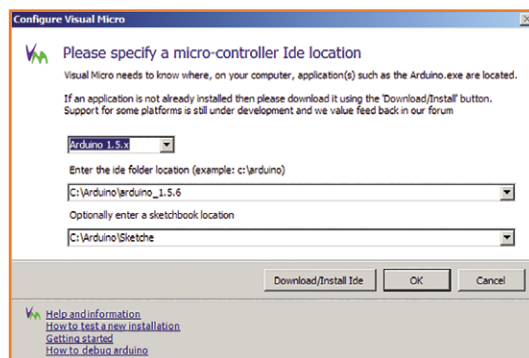


Bild 2.
Eingabe der Pfade
der Arduino-IDE im
Configuration-Manager.



Der Autor hat sich deshalb mit einer Alternative beschäftigt, welche diese Nachteile beseitigt. Ein Grund für das Umsatteln auf einen 32-bit-Controller ist der Geschwindigkeitsgewinn beim Ansprechen von Hardware - insbesondere der IO-Pins. Ein guter Test für die Leistungsfähigkeit eines Mikrocontrollers ist daher die Vermessung ausgegebener Spannungsimpulse, die innerhalb eines Arduino-Sketches als kürzestmögliche Impulse programmiert wurden. Nachfolgend wird beschrieben, wie man solch einen Test-Sketch auch unter Atmel Studio 6.2 [1] schreiben kann.

Arduino-Kompatibilität

Das hier beschriebene Vorgehen setzt ein ins-

talliertes Atmel Studio 6.2 (oder neuer) unter Windows voraus. Um Atmels IDE arduino-kompatibel zu machen muss man noch ein kostenloses Add-In namens „Visual Micro“ [2] installieren. Dabei handelt es sich um eine Arduino-IDE für Microsofts *Visual Studio* und Atmels *Atmel Studio*. Visual Micro steht sofort nach der Installation in Atmel Studio zur Verfügung.

Der unter [2] angebotene kostenpflichtige USB-gestützte „Arduino Debugger“ ist hingegen nicht notwendig, denn das Debuggen mit dem Debugger Atmel-ICE ist wesentlich komfortabler. Dieses Tool ist über diverse Webshops für etwa 100 bis 120 € erhältlich (siehe den Artikel in der letzten Ausgabe [3]). Die Ausgabe lohnt sich, wenn man mehr als selten mit Atmels Mikrocontrollern zu tun hat.

Mit diesem Debugger können echte Breakpoints ohne Neukompilierung im Programm gesetzt, Variablen im Watch-Fenster beobachtet sowie Speicherinhalte betrachtet und geändert werden. Vor allem aber lassen sich die umfangreichen IO-Register inspizieren und per Mausklick verändern.

Nach der Installation von Visual Micro erscheint in Atmel Studio eine neue Toolbar, mit der die aktuelle Arduino-Version (1.5.x), das Board (Arduino Due) und die Programmierschnittstelle (Programming Port) ausgewählt werden (**Bild 1**). Anschließend muss man noch die passende virtuelle serielle Schnittstelle einstellen, die man durch einen Blick in den Gerätemanager herausfinden kann, wenn Arduino Due mit einem USB-Kabel an den PC angeschlossen ist. Wie in der Arduino-IDE befindet sich rechts daneben die Schaltfläche für den seriellen Monitor.

Wenn diese Informationen nicht automatisch erscheinen, weil sich z.B. die Arduino-IDE nicht im Standardverzeichnis befindet, muss der Eintrag „Configuration Manager“ gewählt werden, wodurch sich das Fenster von **Bild 2** öffnet. Hier muss man dann die korrekten Pfade manuell eingeben, denn eine Dialogauswahl ist nicht möglich. Ein Klick auf das Fragezeichen im Menü von Bild 1 bringt den sogenannten „Micro-Explorer“ zum Vorschein (**Bild 3**), ein Gegenstück zum „Solution Explorer“ von Atmel Studio. Unter dem Reiter „Reference“ erscheinen dann die Arduino-Referenzen als Links. Ein Klick auf die einzelnen Einträge öffnet die passende Webseite im voreingestellten Browser. Unter dem Reiter „Examples“ finden sich eine thematisch grup-

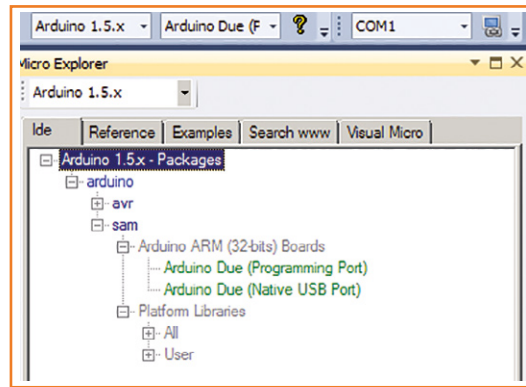


Bild 3.
Der Micro-Explorer von Visual Micro.

pierte Liste von Beispielen und „Library-Examples“ der Arduino-IDE.

Sketche in Atmel Studio

Ein großer Vorteil der Atmel-Studio-IDE ist die Code-Vervollständigung, die man nun auch für Arduino-Sketche zur Verfügung hat. Hierzu muss man „Visual Assist X“ via *VAssistX* -> *Enable/Disable* aktivieren. Gibt man dann im Code-Fenster einen Buchstaben ein, werden alle vorhandenen Möglichkeiten zur Vervollständigung aufgelistet. Wird bei einem Sketch in einer neuen Zeile beispielsweise ein „S“ eingegeben, dann werden „Serial, Server, SPI...“ usw. vorgeschlagen. Diese Begriffe können dann direkt und tipfehlerfrei ausgewählt werden. Noch besser: Nach der Eingabe eines Punktes z.B. hinter „Serial“ werden die vorhandenen Eigenschaften und Methoden dieser Klasse zur Auswahl aufgelistet (**Bild 4**).

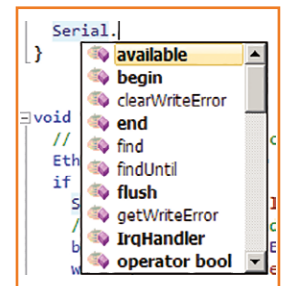
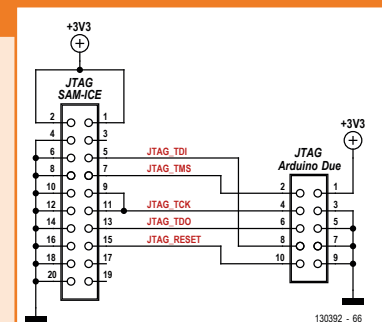


Bild 4.
Der Editor von Atmel Studio erleichtert mit seinen Vorschlägen die fehlerfreie Eingabe von Funktionen und Optionen.

Man könnte nun ganz normal einen Sketch in der Arduino-IDE anlegen oder öffnen und dann in Atmel Studio importieren. Atmel Studio erlaubt es jetzt aber auch, direkt einen Arduino-Sketch neu anzulegen. Hat man den Sketch „Speed-Test“ (siehe **Listing 1**) auf einer dieser beiden Arten angelegt und gesichert, erscheint im Solu-

Adapter für SAM-ICE

Zum Anschluss des Debuggers SAM-ICE an den Arduino Due ist diese Adapterschaltung erforderlich. Sie verbindet die 10-polige JTAG-Stiftleiste im Rastermaß 1/20“ (1,27 mm) auf dem Due-Board mit dem 20-poligen JTAG-Anschluss des SAM-ICE.



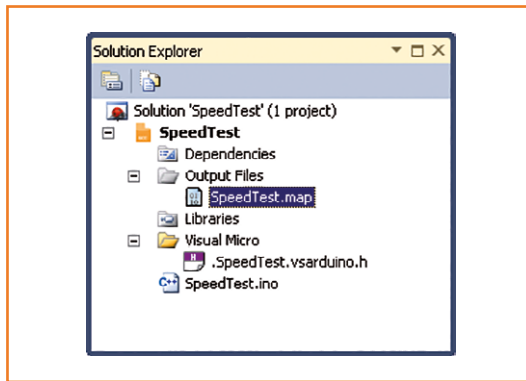


Bild 5.
Der Solution-Explorer von
Atmel Studio.

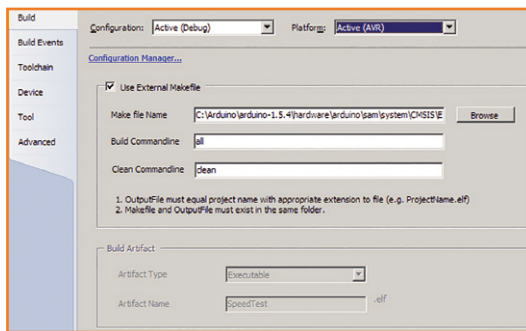


Bild 6.
Die falsche Anzeige der
Plattform; alles funktioniert
aber dennoch.

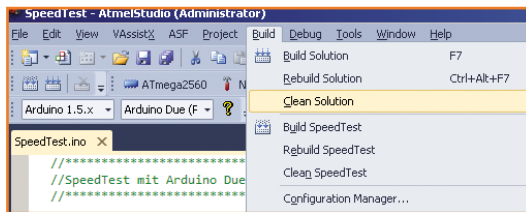


Bild 7.
Compiler-Lauf mit der
Option „Clean Solution“.

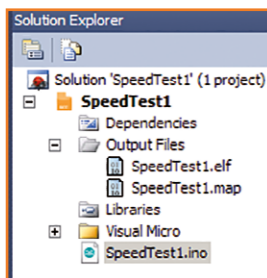


Bild 8.
Output-Files im Solution-
Explorer.

tion-Explorer eine Struktur wie in **Bild 5**. Man kann den Sketch dann öffnen und editieren wie von Atmel Studio gewohnt. Ab hier braucht man die Arduino-IDE eigentlich nicht mehr zu starten. Dennoch darf man sie nicht löschen, denn sie muss auf dem PC verbleiben.

Etwas verwirrend ist, dass zumindest in der aktuellen Version 6.2 bei den Eigenschaften des Projekts (rechte Maustaste auf SpeedTest: *Properties* -> *Build*) als Plattform „Active AVR“ angezeigt wird. Dies lässt sich leider nicht ändern. Auch unter „Toolchain“ und „Device“ wird immer die AVR-Plattform angezeigt. Trotzdem funktioniert alles. Die Auswahl von „Arduino Due“ als Board bewirkt das Kompilieren des Projekts wie unter der SAM-Plattform (für Atmels ARM-Controller). In **Bild 6** ist angeblich die AVR-Plattform aktiv. Der „Make file Name“ ist aber korrekt für die SAM-Tool-Chain der Arduino-IDE.

Wie von Atmel Studio gewohnt wird das Projekt

mit F7 kompiliert und das Programm mit F5 (oder Klick auf den ausgefüllten grünen Pfeil) mit Hilfe des Bootloaders in das Board geladen. Wichtig ist ein Build mit dem Menüpunkt „Clean Solution“ (**Bild 7**), denn nur dann werden alle Dateien wie bei der Arduino-IDE kompiliert.

Wie schon erwähnt ist einer der großen Vorteile des Atmel Studio (gegenüber einer reinen Arduino-IDE) die größere Geschwindigkeit bei der Kompilierung, was die Zyklen „Kompilieren-Ändern-Kompilieren-Ändern...“ deutlich beschleunigt. Während nämlich die Arduino-IDE immer alle Programmkomponenten kompiliert, berücksichtigt Atmel Studio nur die geänderten Dateien. Dadurch wird in der Regel nur die .ino-Datei kompiliert, die den Sketch-Code enthält. Bei einem ersten Compilerlauf des Beispiels „SpeedTest“ benötigte der PC des Autors 3,7 s. Ein zweiter Lauf nach einer Änderung war schon nach 0,23 s abgeschlossen.

Debuggen mit Atmel-ICE

Ein „richtiger“ Hardware-Debugger wie Atmel-ICE macht die Software-Entwicklung sehr viel komfortabler. Zum Anschluss von Atmel-ICE an Arduino Due wird nur das mitgelieferte Basic-Kabel benötigt. Arduino Due wird entweder über einen seiner USB-Anschlüsse oder ein externes Netzteil versorgt. Atmel-ICE selbst benötigt nur den USB-Anschluss.

Beim Atmel-ICE gibt es zwei Steckmöglichkeiten für das Debug-Kabel. Für Arduino Due ist der SAM-Anschluss richtig. Das andere Ende des Kabels kommt an den JTAG-Anschluss des Boards. Eigentlich ist es in der aktuellen 6.2er-Version von Atmel Studio nicht möglich, einen Arduino-Sketch unter der Atmel-ARM-Plattform zu bearbeiten. Aber es gibt einen Ausweg: Zum Debuggen mit dem Atmel-ICE wird einfach eine zweite Instanz von Atmel Studio geöffnet. Hier wählt man dann *Open -> Open Object File For Debugging*. Die von Atmel Studio beim Kompilieren erzeugten Dateien befinden sich unter Windows 7 im Ordner:

```
C:\Users\xxxx\AppData\Local\
  VMicro\Arduino\Builds\SpeedTest\
    arduino_due_x_dbg\
```

Darin befindet sich die Datei „SpeedTest.elf“. Sie wird jetzt zum Debuggen geöffnet.

Unter der Voraussetzung, dass der Sketch mit der Arduino-IDE übersetzt und gesichert wurde, befindet sich auch eine Kopie dieser Datei im

Debug-Verzeichnis des Sketches. Sie ist im Solution Explorer des erzeugenden Atmel Studios unter „Output Files“ zu sehen (**Bild 8**).

Die Ausgabe-Datei „SpeedTest.elf“ wird per Datei-Dialog gesucht und ausgewählt. Das Ergebnis sollte dann so aussehen wie in **Bild 9**. Der Projektname und der Ablageort des Debug-Projekts wurden mit „_Debug“ frei gewählt. Beim nächsten Start des Debug-Projekts enthält dieses bereits die gespeicherten Einstellungen.

Als nächstes erscheint ein Fenster zur Wahl von Plattform und Controller-Typ (**Bild 10**). Bei Arduino Due ist die Device-Family „SAM3, 32 bit“ und der Mikrocontroller „ATSAM3X8E“ richtig. Im Solution-Explorer erscheint nun die Datei „SpeedTest.cpp“. Sie ist nichts anderes als die Programmdatei des Sketches, die ursprünglich die Extension „.ino“ hatte. Ein Rechtsklick auf den Projektnamen „SpeedTest_Debug“ öffnet das Property-Fenster.

Wenn Atmel-ICE über USB mit dem PC verbunden ist, lässt sich dieser Debugger unter *Tool-> Selected debugger/programmer* auswählen. Wichtig: Als Interface ist anstelle von „JTAG“ unbedingt „SWD“ anzugeben.

Atmel-ICE ist bereit und kann jetzt auch als Programmer benutzt werden: Mit *Ctrl-Shift-P* oder einem Klick auf das Chip-Symbol mit dem Blitz öffnet sich das Programmer-Fenster. Nach Betätigen von „Apply“ sind die gewünschten Funktionen verfügbar. Unter den „Interface-Settings“ kann man den maximalen SWD-Takt von 2 MHz einstellen. Unter dem Reiter „Memory“ kann man die Datei „SpeedTest.elf“ direkt in den Flash-Speicher des Controllers laden. Das Programm wird nach Beenden des Programmiervorgangs gestartet. Für das Hochladen des Programms zum Debuggen oder zum Start wird aber effizienterweise nicht diese Device-Programmierung verwendet, sondern *Debug -> Start Debugging and Break* oder *Start Without Debugging* (bzw. der blaue Pfeil mit den blauen senkrechten Balken oder der ausgefüllte grüne Pfeil).

Das Hochladen des gesamten Programms auf den Controller dauert mit Atmel-ICE nur etwa 1 s - ist also richtig schnell.

Im Solution-Explorer wird nach dem Hochladen die wahre Struktur des Arduino-Programms sichtbar, und der Cursor steht in der Datei „main.cpp“ in der ersten Programmzeile (**Bild 11**). Es wird deutlich, dass die Datei „SpeedTest.cpp“ nur ein kleiner Teil des Gesamtprogramms ist, weshalb eine komplette Kompilierung so lange dauert.

Mit dem Debugger stehen nun alle Vorteile von Atmel Studio wie Einzelschritte mit F11, Breakpoints mit F9, Bedingungen für Breakpoints im Breakpoint-Fenster etc. zur Verfügung.

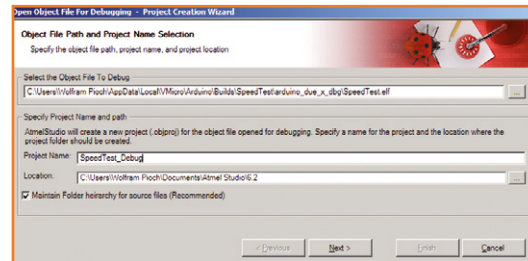


Bild 9.
Dialog zur Auswahl der erzeugten Ausgabe-Datei.

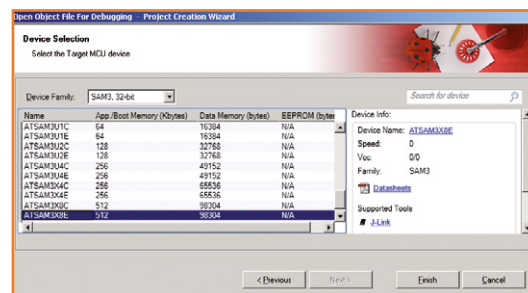


Bild 10.
Dialog zur Auswahl von Chip-Familie und Chip.

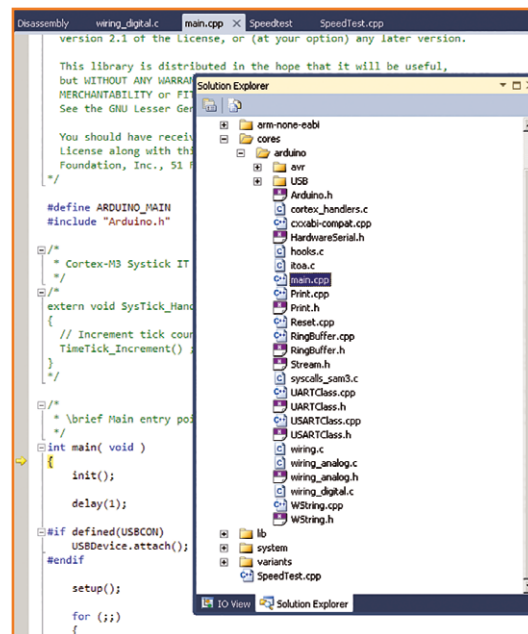


Bild 11.
Im Solution-Explorer sieht man die wahre Komplexität eines Arduino-Projekts.

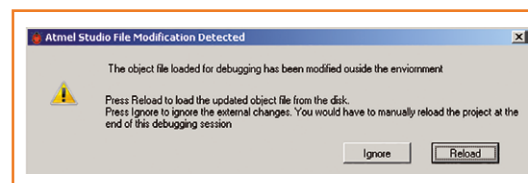


Bild 12.
Die Fehlermeldung bei geänderter Datei kann man wegklicken.

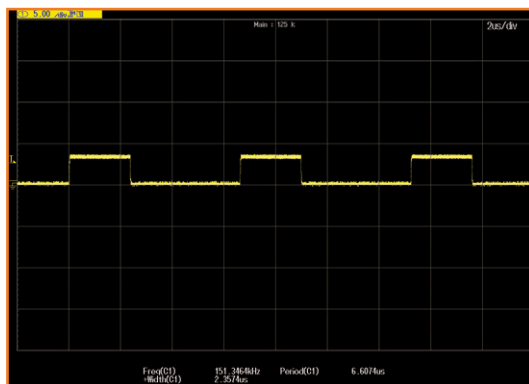


Bild 13.
Oszillogramm des
resultierenden
Rechtecksignals mit
150 kHz.

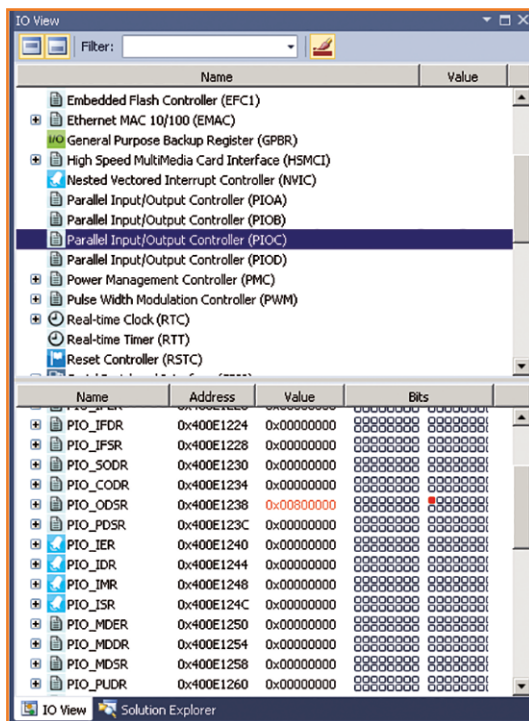


Bild 14.
PIO-Register des Controllers.

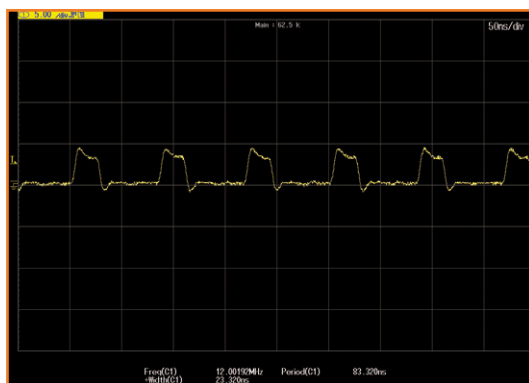


Bild 15.
Oszillogramm des
Rechtecksignals bei der
Programmversion 2.0 mit
12 MHz.

Wird die Datei „SpeedTest.ino“ in der ersten Instanz von Atmel Studios geändert und neu kompiliert, erscheint in der zweiten Instanz (Debug) die folgende Meldung von **Bild 12**. Durch Betätigung von „Reload“ wird der Debug-Status verlassen und das geänderte Programm in den Controller geladen. Der Cursor steht nun wieder auf der ersten Programmzeile und das Programm kann erneut gestartet werden.

Wer schon unter Atmel Studio mit 8-bit-AVR-Controllern gearbeitet hat, der wird auch mit 32-bit-ARM-Controllern klar kommen, da sich die Bedienung praktisch nicht unterscheidet. Übrigens hat das Löschen und Neubeschreiben des Flash-Speichers mit Atmel-ICE keinen Einfluss auf den Bootloader, denn dieser befindet sich im ROM des Controllers und kann daher gar nicht gelöscht oder überschrieben werden. Das simple Hochladen eines Programms über die USB-Schnittstelle funktioniert also nach wie vor.

Der Speed-Test

Im Beispiel-Sketch (**Listing SpeedTest**) werden in der Hauptschleife nur diese beiden Befehle ausgeführt:

```
digitalWrite(TP1,1); //On
digitalWrite(TP1,0); //Off
```

Mit dem Oszilloskop ist an diesem Pin nach Programmstart ein Rechtecksignal mit der Impulsbreite von 2,35 µs und einer Periodendauer von 6,6 µs zu sehen (**Bild 13**), was einer Frequenz von rund 150 kHz entspricht. Das ist nicht wirklich viel und der erste Gedanke ist sicher, ob man dafür einen ARM-Controller mit einem Takt von immerhin 84 MHz benutzen muss, denn das kann ja ein simpler 8-bit-AVR-Controller besser. Des Rätsels Lösung: Hinter der Arduino-Anweisung „digitalWrite“ stecken etliche C-Routinen, wie man mit dem Debugger im Einzelschrittbetrieb (F11) überprüfen kann. Um tatsächlich eine hohe Verarbeitungsgeschwindigkeit zu erreichen, muss man diesen Overhead komplett vermeiden. Glücklicherweise ist die Arduino-Programmiersprache keine „richtige“ Sprache, sondern ganz normales C (plus GNU-Compiler in einer speziellen Arduino-Konfiguration).

Ein als Output konfigurierter Pin kann somit durch direktes Setzen eines Bits im PIO-Register angesprochen werden. Mit der Anweisung:

```
const int TP1 = 7; //Testpin
```

wurde als Testpin der Arduino-Pin 7 gewählt. Dies entspricht laut Arduino-Pin-Mapping dem Controller-Pin PC23 bzw. Bit 23 im PIO-Register (Parallel Input/Output). Bei angehaltenem Programm kann man im IO-View des Debuggers (**Bild 14**) mit der Maus auf Bit 23 des Registers „PIO_ODSR“ (= Set) klicken und der Pin wird high. Zum Rücksetzen dient das Register PIO_CODR (= Clear). In der Hauptschleife kann dieser Pin mit der simplen Anweisung:

```
PIOC->PIO_ODR = 1<<23;
```

direkt und ohne Umschweife gesetzt werden.

Für einen gleichzeitigen Test, ob auch die bedingte Kompilierung funktioniert, wurde eine neue Version des Sketches erstellt (siehe **Listing Speed-Test 2.0**). Wenn der Ausdruck „Direct“ definiert ist, wird der Programmteil nach „#ifdef Direct“ übersetzt – andernfalls die alte Version mit dem langsamen „digitalwrite“. Selbstverständlich würde diese Änderung auch in der normalen Arduino-IDE funktionieren.

Das Ergebnis dieser Optimierung in **Bild 15** ist beachtlich: Das Rechtecksignal hat nur noch eine Impulsbreite von 23,2 ns und eine Periodendauer von 83,3 ns. Die resultierenden 12 MHz sind etwa um den Faktor 80 flotter! Der Einsatz eines ARM-Controllers wirkt sich also deutlich aus.

Fazit

Die Verwendung von Atmel Studio beschleunigt die Arduino-Entwicklung dank deutlich kürzeren Turn-Around-Zeiten auf ein ganz neues Niveau, denn hier wird nur der geänderte

Listing 1. Speed-Test.

```
//*****
//Speed-Test with Arduino Due
//*****

const int TP1 = 7; //Testpin

void setup()
{
  /* add setup code here */
  // set the digital pin as output:
  pinMode(TP1, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(TP1,1); //On
  digitalWrite(TP1,0); //Aus
}
```

Electronica
und überzeugen Sie sich selbst
11. - 14. November 2014
Messe München
Hall A1 Stand 115

pico
Technology

PicoScope® PC-OSZILLOSKOPE

Für jede Anwendung gibt es ein passendes

- Bandbreite von 5 MHz bis 1 GHz
- Sampling von 10 MS/s bis 5 GS/s
- Pufferspeicher von 8 ks bis 2 GS

PICOSCOPE 2200A SERIE

Das PicoScope im Hemdtaschen-Format



PICOSCOPE 3400 SERIE

Hochleistungs-Oszilloskope



PICOSCOPE MSOs

Mixed-Signal-Analyse



PICOSCOPE 4000 SERIE

Präzisions-Oszilloskope



PICOSCOPE 4824

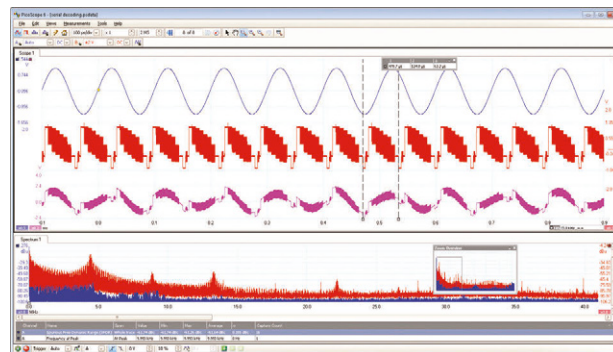
8 Kanäle, hohe Auflösung

PICOSCOPE 5000 SERIE

Oszilloskope mit flexibler Auflösung

PICOSCOPE 6000 SERIE

Ultimative Performance, jetzt mit USB 3.0



Alle Modelle inkl. vollständiger Software und 5 Jahren Gewährleistung. Software inkl. Messungen, Spektrum-Analysator, SDK, erweiterten Triggern, farbigem Nachleuchten, seriellen Decoding (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), Masken, Mathematik-Kanälen, alles standardmäßig, mit freien Updates.

www.picotech.com/PS364

Listing 2. Speed-Test 2.0.

```
//*****
//Speed-Test 2.0 with Arduino Due
//*****
#include „arduino.h“
//
//
const int LED1 = 13;
      int LED2 = 12;
      int LED3 = 11;
      int TP1  = 7; //Testpin

#define Direct

void setup()
{
  /* add setup code here */
  // set the digital pin as output:
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(TP1, OUTPUT);
  // set output low
  digitalWrite(LED1,0);
  digitalWrite(LED2,0);
  digitalWrite(LED3,0);
  digitalWrite(TP1,0);
}

void loop() {
  // put your main code here, to run repeatedly:
#ifdef Direct
  //x= state of PIO_SODR with bit 23 = 1
  int x = PIOC->PIO_SODR | 1<<23;

  while (1){ //Only for testpurposes, don't exit this loop
    PIOC->PIO_SODR = x;
    PIOC->PIO_CODR = x;
  }

#else
  digitalWrite(TP1,1);
  digitalWrite(TP1,0);
#endif
}
```

Code neu kompiliert. Atmel Studio und der damit mögliche Einsatz von Tools wie Atmel-ICE als Debugger und Programmer lohnen sich daher immer, wenn man mit Atmel-Controllern zu tun hat, da so nicht nur professionelle Debugging-Möglichkeiten hinzukommen, sondern auch das Beschreiben des Controller-Speichers deutlich flotter vonstatten geht.

Übrigens kann man für ARM-Projekte statt des Atmel-ICE auch den von Atmel angebotenen SAM-ICE verwenden. Mit diesem Hardware-Debugger/Programmer lässt sich das Programmieren eines Controllers nochmals um den Faktor 4 steigern, da dieser mit 8 statt 2 MHz arbeiten kann. Hierfür ist dann der im **Kasten** (Artikelanfang) beschriebene Adapter erforderlich. Wenn man professionell entwickelt, lohnt sich der Aufwand auf alle Fälle sehr schnell, auch wenn man damit lediglich die Atmel-Controller der AT91-Serie betreiben kann.

(130392)

Weblinks:

- [1] Atmel Studio 6.2:
www.atmel.com/microsite/atmel_studio6
- [2] Arduino-IDE for Visual Studio:
www.visualmicro.com
- [3] Atmel-ICE:
www.atmel.com/tools/atmel-ice.aspx
Review: www.elektor-magazine.de/140275
- [4] SAM-ICE:
www.atmel.com/tools/atmelsam-ice.aspx

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



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

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

UMWELTSCHONEND – GÜNSTIG – GREEN

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



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

Telefunken-Funktelefon von 1965

Von Gerd Kowalewski
(D)

HighTech aus der Wirtschaftswunderzeit



Nachdruck mit freundlicher Genehmigung des Museums für Kommunikation, Frankfurt.

Heute haben wir jederzeit ein kleines und leichtes Handy zur Hand und können damit überall auf der Welt anrufen. Und nicht nur das: Unsere Kommunikations- und Informationsbedürfnisse befriedigen wir heutzutage mit energiesparenden Smartphones, die mehrere Mikroprozessoren und oft auch ein hochauflösendes Display mitbringen. Passend dazu hat sich der Mobilfunk zu einem leistungsfähigen drahtlosen Netzwerk entwickelt. Diese modernen elektronischen Begleiter bieten weit mehr als lediglich Telefonie wie anno dazumal. Mobile Kommunikation bedeutet heute nicht nur die Übertragung digitalisierter Sprache, sondern auch direkte schriftliche Konversation in Form von E-Mails oder Text-Nachrichten. Wir verschicken Musik, Fotos und Dateien beliebigen Inhalts. Mehr noch: Wir vernetzen uns multimedial, empfangen Geodaten zwecks Navigation und betrachten Videos aus Dateien mit hochkomprimierten audiovisuellen Inhalten. Wir können sogar per Live-Streaming virtuell bei fast jedem Ereignis weltweit dabei sein und reizen die quasi unendlichen Möglichkeiten des World Wide Web immer weiter aus.

Bild 1.
Der Garagenfund: Zwei unscheinbare Blechkoffer mit gleichen Abmessungen. Auf der Frontplatte prangt der Schriftzug „Telefunken“.



Bild 2.
Die beiden Geräteköffer dieses Mobiltelefons mitsamt Kabelsatz.



Ein Standard-Handy für die Mobilfunknetze D, E bis LTE verfügt heute über eine Rechenleistung, die einen Desktop-PC von vor einigen Jahren wie Spielzeug aussehen lässt. Dank des technischen Fortschritts im Allgemeinen und in der Mikroelektronik im Besonderen verfügen diese leichtgewichtigen Gadgets über so viel Ausdauer, dass ihr Akku erst nach tagelanger Benutzung wieder aufgeladen werden muss. Dabei übersieht man leicht eine zentrale Voraussetzung: die Infrastruktur der Mobilfunknetze als intelligente Technik im Hintergrund.

Garagenfund

Im Nachlass eines Verstorbenen fand sich ein verstaubter Karton, in dem für Jahrzehnte zwei unscheinbare Blechkisten eingelagert waren, auf denen der Name des einst bekannten Herstellers

Telefunken prangte (**Bild 1**). Allein diese Entdeckung war ausreichend, die Neugier eines Elektronikers zu wecken. Dass der Autor hier zufällig über ein altes UKW-Mobilfunkgerät stolperte, war ihm außerdem Anlass, sich Gedanken über die Geschichte des Mobilfunks in Deutschland zu machen. Im Vergleich zu heutigen Handys sind diese Geräte eine Seltenheit, wenn man die kleinen damaligen Stückzahlen bedenkt.

Weitere Nachforschungen ergaben, dass dieser Fund ein ganz besonderes „Schätzchen“ war: ein Sende-Empfänger des Typs S/E 160E15 öbL B mit Stromversorgung und Selektivrufeinheit für den Einbau in Fahrzeuge mitsamt komplettem Kabelsatz (**Bild 2**). Dieses Gerät stammt aus den ersten Tagen des öffentlichen deutschen Mobilfunks, dem so genannten A-Netz aus den 1960er Jahren, und es war sehr gut erhalten. Die **Bilder 3a, 3b, 4a** und **4b** zeigen, wie die beiden damals „Gerätekoffer“ genannten Module miteinander verbunden werden.

Das Suffix „öbL B“ auf dem hinteren Typenschild und die UKW-Buchse nach SO-239 geben Hinweise auf den Zweck dieses Apparats (**Bild 3b**). Die Typenbezeichnung „S/E 160E15“ verrät: Es handelt sich um einen Sender/Empfänger, der im Bereich von 160 MHz arbeitet und über wählbare 15 Kommunikationskanäle des damaligen A1-Netzes verfügt (insgesamt gab es 17). Zwei UKW-Träger mit einem Abstand von 4,5 MHz bilden die Basis für Sprechfunk bzw. eine Vollduplex-Verbindung mit Basisstationen in Entfernungen von bis zu 50 km.

Dieses äußerst seltene Stück Elektronik wurde einst in Ulm vom Funktechnikspezialisten Telefunken hergestellt und war der kompaktere Nachfolger des Modells S/E 160E11. Es wurde in relativ kleinen Stückzahlen mit viel Handarbeit ab 1965 (siehe FTZ-Nummer) produziert.

Anfangs nutzte nur eine Handvoll besonders wohlhabender Kunden das A-Netz [1] – das offiziell als „öffentlicher beweglicher Landfunkdienst“ (öbL) bezeichnet und von der DBP (Deutsche Bundespost) betrieben wurde. Die Kapazitätsgrenze der letzten Ausbaustufe wurde schon 1971 mit der damals unvorstellbaren Zahl von 10.784 Teilnehmern erreicht.

Die Netzkapazität war sicher einer der Gründe für die beschränkte Verbreitung dieser Art von Mobiltelefonen. Ein besonders triftiger Grund war



Bild 3.
(3a) Kabel zwischen dem Modul S/E 169E15 öbL B und der Stromversorgung.
(3b) Seitenansicht: Steckverbinder des Telekommunikationsmoduls.

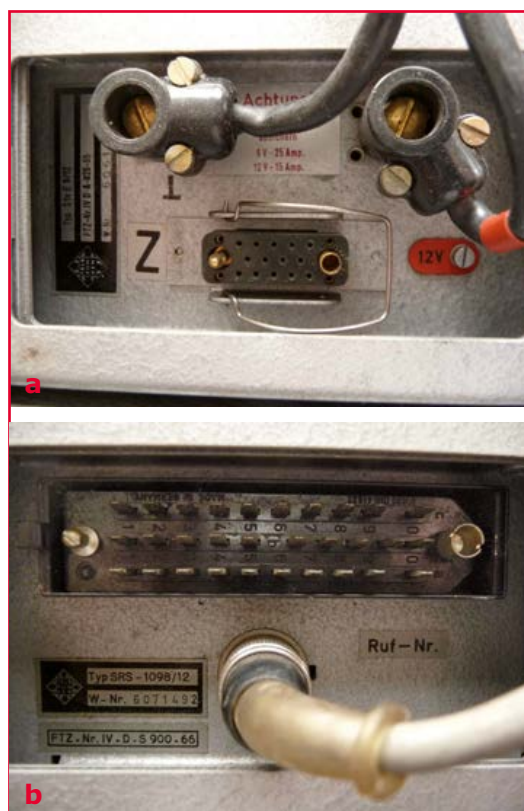


Bild 4.
Seitenansicht: Steckverbinder von Stromversorgung STV E 6/12 (4a) und Selektivruf SRS-1098/12 (4b).

Bild 5.
Innenansicht des Gegentakt-
Spannungswandlers
STV E 6/12 und des
Selektivrufs SRS-1098/12.

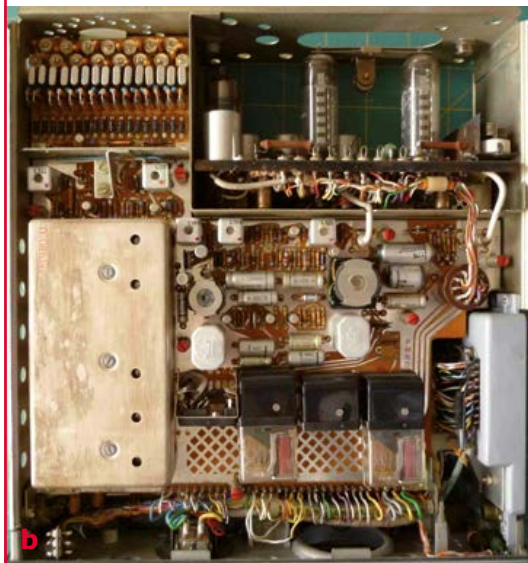
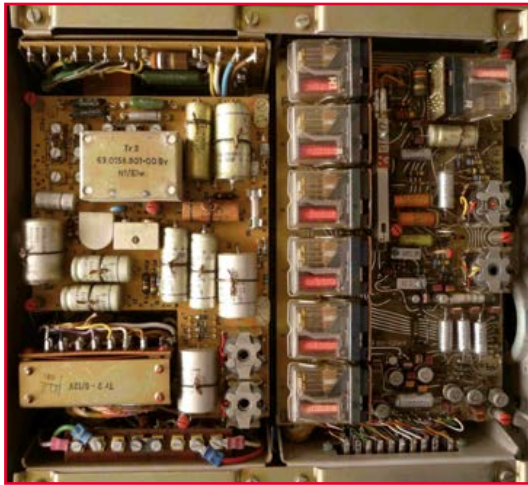


Bild 6.
Innenansicht des
Transceivers.

aber, dass so eine Autotelefonanlage aus zwei im Kofferraum montierten „Gerätekoffern“ und einem Bedienteil für das Armaturenbrett damals ein Vermögen kostete.

Dieses Stück 60er-Jahre-HighTech wog etwa 15 kg. Es benötigte mit maximal 24 W erstaunlich wenig Energie (es war mit 2 A abgesichert). Das Gerät war sogar schon weitgehend transistorisiert: Beispielsweise arbeitete der Gegentakt-Hochspannungswandler zur Erzeugung der Versorgung der wenigen Röhren im zweiten Koffer (**Bild 5**) schon mit Halbleitern. Beim Mischer, dem Treiber und der 10-W-Endstufe des Senders setzte man noch auf Röhrentechnik. In **Bild 6** kann man gut zwei Doppeltetroden des Typs 6360 (QQE03/12) und eine dritte Röhre erkennen. Die Anlage musste mit „Laschen“ bzw. Schaltbrücken an das jeweilige Bordnetz angepasst werden (**Bild 7**), denn längst nicht jedes Fahrzeug hatte damals ein 12-V-Bordnetz mit dem Minuspol am Chassis.

Die Gesamtkosten für die Anlage betrugen 6.550 DM [6] – ohne Einbaukosten versteht sich! In heutiger Kaufkraft entspricht dies der stolzen Summe von 25.000 €. Damals konnte man für diesen Preis auch locker einen recht luxuriösen „Geschäftswagen“ kaufen.

Eine Ausgabe in dieser Größenordnung lag – ganz abgesehen von den Betriebs- und Lizenzkosten – weit jenseits der Möglichkeiten eines durchschnittlichen deutschen Arbeitnehmers. Die monatlichen Belastungen waren mit 65 DM ($\approx$ 250 € für 2014) auch nicht gerade ohne. Später wurde der Ansturm auf das Funknetz behördlicherseits durch Monatsbeiträge von recht unerschämten 270 DM ($\approx$ 1.000 € für 2014) begrenzt. Zu den Kunden dieses Mobilfunkdienstes zählten übrigens nicht nur Politiker und begüterte Unternehmer, sondern auch die Straßenmeisteren, Forstverwaltungen, Bodenlenkungs-Teams an Flughäfen und die Binnen-Schifffahrt. Auch die Deutsche Bahn nutzte das A-Netz für ihren Zugfunk.

Beim Vorgänger des A1-Netzes reichte ein Kanal für etwa 20 bis 25 Teilnehmer aus. Dabei konnten alle Teilnehmer einfach alles mithören. Damit ein Mobilfunkteilnehmer einen Anruf tätigen konnte, musste dieser anhand einer Code-Karte zuerst seinen lokalen Kanal ausfindig machen. Er musste sich also einigermaßen in Deutschland ausken-

nen. Ein 2.280-Hz-Bereitschaftssignal zeigte an, dass man sich im Funkbereich eines freien Kanals der Basisstation aufhielt, die mit dem Festnetz verbunden war. Zur Besetztanzeige von Kanälen diente ein 1.750-Hz-Signal. Diese Töne wurden selbstverständlich im normalen Sprachkanal von den Basisstationen ausgesendet. Ohne „Freizeichen“ war es nicht möglich, einen Kanal zu belegen.

Selektivruf

Die siebenstellige Rufnummer [2, 3] eines Mobilfunkteilnehmers unterlag einem relativ komplexen Kodierungsschema. Die erste und zweite Stelle repräsentierten den so genannten *Teilnehmerschlüssel*. Damit wurde Bereiche der *Funknetz*karten belegt. Beim A1-Netz waren das die Bereiche 21 bis 25 und 31, beim A2-Netzwerk 32 und 61 bis 65. Für die bewährte Tonfrequenzsignalisierung im Rufkanal wurde mit präzisen Tonfrequenzen (1 % Genauigkeit) das Anrufschema kodiert. Die dritte Stelle bezog sich auf den *Gruppenschlüssel* [3]. Er diente beim Selektivrufsystem „4-über-30“ zur Auswahl von vier aus 30 Tonfrequenzen, die eine vierstellige *Ordnungszahl* (die Ziffern 4 bis 7 der Mobilfunknummer) kodierten.

Mit diesem Verfahren konnte man einen sogenannten *Dauerruf* im Rufkanal zur teilautomatischen Anrufrufkodierung von maximal 27.405 Teilnehmern etablieren

- dies aber nur unter der Voraussetzung, dass der Standort des Teilnehmers zumindest grob bekannt war. Es gab also weder Roaming, Triangulation noch Weitergabe des Gesprächs an andere Funkzellen.

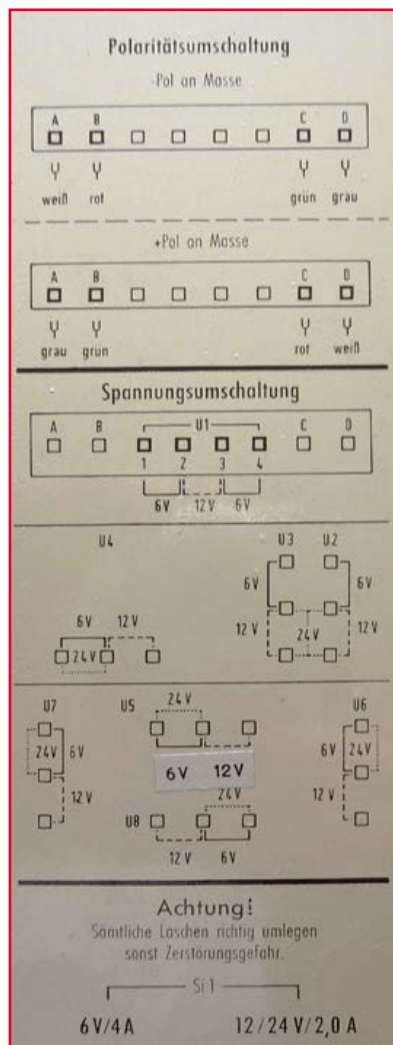
Das eingesetzte Prinzip machte beim Empfänger sehr schmalbandige Resonanzkreise erforderlich, die dazu in der Lage waren, gleichzeitig vier Tonfrequenzen in einem Kanal nach einem Tonfrequenzschema mit: $F(n) = 337,5 \text{ Hz} + n \cdot 15 \text{ Hz}$; mit $n = 1$ bis 30 zu detektieren.

Zunächst reichte $n = 20$ für die Zahl der Teilnehmer aus. Später wurden die Möglichkeiten des Selektivrufs beim A3-Netz auf $n = 30$ erweitert.

Geöffnete Gerätekoffer

Gerät 1 ist der eigentliche UKW-Funkfern-sprech-Sende-Empfänger *160E15 öbL* in der Variante B, Typ G 50-3. Es umfasst damit die per Kabel fernbediente eigentliche Nach-richtentechnik mit folgenden technischen Daten:

Bild 7. Wie bitte? Betrieb an 6 V, 12 V oder gar 24 V? Und auch noch umschaltbar, ob der Pluspol oder der Minuspol des Akkus am Chassis liegen? Das war damals tatsächlich notwendig!



UNSCHLAGBAR

beim Preis-Leistungsverhältnis.



Rigol DS1000E Oszilloskope

2 Kanäle, 50/100 MHz, 1 GSa/s Abtastrate, 1 Millionen Messpunkte Speicher, USB, LAN, einfache Messfunktionen, 3 Jahre Garantie

ab € 284,41
inkl. MwSt. und Versand



Rigol DS1000Z Oszilloskope

4 Kanäle, 50-100 MHz, 1 GSa/s Abtastrate, 12 Millionen Messpunkte Speicher, USB, LAN, professionelle Mess- & Analysefunktionen, optional mit eingebautem Funktionsgenerator, 3 Jahre Garantie

ab € 355,81
inkl. MwSt. und Versand



electronica
Halle A1 Stand 654

Geräte einfach auf dem Batronix Messestand ausprobieren und Messerabatte sichern!

Batronix Elektronik

✓ Rechnungsbau

100% sicher und schnell. Erst nach Erhalt der Ware zahlen.

✓ Große Auswahl ab Lager

✓ 30 Tage testen

✓ Geld zurück Garantie

Jetzt Angebote nutzen:

www.batronix.com/go/41

Batronix Elektronik
Lise-Meiner-Str. 1-7
24223 Schwentinental

Tel.: +49 (0)4307/8243-20
service@batronix.com
www.batronix.com

NEU

EST^D 2004

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

- 15 von insgesamt 17 Duplex-Kanälen aus dem A1-Netz = 34 quarzstabilisierte UKW-Trägerfrequenzen (über 30 Quarze). Bestückt sind die Kanäle:
- Kanäle 30 bis 44; 157,55/162,05 MHz bis 158,25/162,75 MHz.
- Frequenzabstand: 4,5 MHz für Vollduplexbetrieb im 50-kHz-Raster.
- Betriebsart: FM (F3); Hub: 4 kHz.
- Anrufquittung: 1.750 Hz $\pm$ 1 Hz.
- HF-Sendeleistung: 10 W, vor dem Duplexer.

- Antennenbuchse: PL-Typ (SO-239).

Gerät 2 enthält den Gerätesatz STV/SRS mit der Baugruppe STV E 6/12. Dabei handelt es sich um einen voll transistorisierten Spannungswandler, der aus einem 6/12/24-V-Bordnetz mit wahlweise negativem oder positivem Pol am Chassis alle notwendigen Betriebsspannungen erzeugt und stabilisiert. Beim Betrieb an 12 V und 24 V ist das Gerät mit 2 A abgesichert. Die Leistungsaufnahme an einem typischen 12-V-Bordnetz beträgt daher maximal 24 W. Es wurde empfohlen, die Anlage zum Bordnetz hin mit 15 A abzusichern. Enthalten ist außerdem auch noch die Selektivrufereinheit SRS 1098/12 für die Anrufsignalisierung über den Service-Kanal.

Das Batteriekabel, die Original-Verbindungsleitungen und selbst die alte Antennenleitung sind samt Antennenfuß (aber ohne eigentliche Antenne) auch nach 50 Jahren noch komplett erhalten!

Geschichte des Mobilfunks in Deutschland

Schon 1918 gab es bei der damaligen Deutschen Reichsbahn erste Experimente im Bereich Zugfunktelefonie. Anfangs wurde noch mit so genannten Langwellen und dann mit „Grenzwellen“ genannten Frequenzen im Bereich um 30 MHz operiert. Ab 1926 wurde den Passagieren in den schnellen „D-Zügen“ auf der Strecke Berlin-Hamburg ein regulärer Telefoniedienst per Funk angeboten. Ab 1932 schließlich konnte die Bahn-Servicegesellschaft Mitropa sogar handvermittelte Telefonie zwischen fahrenden Zügen und Schiffen auf den Weltmeeren per Seefunk anbieten. Damit war in Deutschland eine erste Form mobiler Telefonie bereits vor der Machtergreifung der NSDAP im Jahre 1933 recht weit ausgebaut. Mit dem beginnenden zweiten Weltkrieg wurden diese Dienste aber wieder eingestellt.

In den zunächst durch die alliierten Siegermächte besetzten Gebieten Nachkriegsdeutschlands änderte sich die Situation nach Demilitarisierung und Entnazifizierung in den drei westlichen Besatzungszonen wieder, nachdem der amerikanische Marshall-Plan 1947 angelaufen, die Währungsreform 1948 durchgeführt war und der Wiederaufbau der zerstörten Infrastruktur und der Städte begonnen hatte. Im Jahr 1949 endlich wurde in der „Trizone“ Westdeutschlands eine demokratische Verfassung verabschiedet, die Bundesländer eingeführt und die Bundesrepublik Deutschland (BRD) gegründet. Die damals staatliche Deutsche Bundespost (DBP) erhielt die Rechte für die Nutzung von Funkfrequenzen gemäß den internationalen Vereinbarungen und innerhalb der Einschränkungen und Auflagen durch die alliierten Siegermächte.

In den 1950er Jahren schließlich kam es zu einem enormen ökonomischen Aufschwung, der im Begriff „Wirtschaftswunder“ eindrücklich beschrieben ist. Ab 1958 fasste die DBP, der damals

alle Telekommunikation in der noch jungen Bundesrepublik unterstand, die einzelnen Funkdienste zum sogenannten A-Netz mit den Frequenzbereichen von 156 bis 174 MHz im so genannten „öffentlichen beweglichen Landfunkdienst“ (öbL) zusammen. In den weiteren Ausbaustufen A1, A2 und A3 wurde dieses Funknetz optimiert und erweitert. 1968 konnte dann schon eine Flächenabdeckung von immerhin 80 % in der Bundesrepublik erreicht werden.

Mit dem B1-Netz wurde ab 1972 dann ein leistungsfähigeres Funknetz mit 38 Kanälen als Nachfolger aufgebaut. Bis zum Ende der 1970er Jahre wuchs die Teilnehmerzahl stark an, obwohl die Kosten für die Inanspruchnahme dieses Dienstes immer noch sehr hoch lagen. Aufgrund des Mangels an freien Frequenzen begann man später damit, die Trägerfrequenzen des A-Netzes für das B2-Netz zu „recyclen“. Ab 1980 standen somit insgesamt 75 Sprachkanäle zur Verfügung. Das neue Frequenzraster für die Kanäle wurde sehr viel enger gefasst und lag statt bei ursprünglich 50 kHz nun bei nur 20 kHz. Kanal 19 wurde bundeseinheitlich als Rufkanal festgelegt. Die Kanalfrequenzen wurden in hinreichend weit genug voneinander entfernten Funkzellen parallel genutzt.

Mit all diesen Maßnahmen gelang es der DBP, insgesamt 850 gleichzeitig nutzbare Frequenzen für die Telefonie zur Verfügung zu stellen. Im B-Netz wurde außerdem erstmals das Telefonieren ohne Handvermittlung durch „das Fräulein vom Amt“ möglich – jedenfalls dann, wenn man anhand einer Funkzonenkarte herausfinden konnte, in welcher der 158 Funkzonen im Bundesgebiet sich der mobile Teilnehmer gerade aufhielt und aus dem Festnetz dann die passende Vorwahl wählte. Roaming über Ländergrenzen hinweg war damals noch undenkbar. Das schnelle Anwachsen der Nutzerzahlen brachte aber auch das B-Netz schon 1986 mit etwa 27.000 Teilnehmern an seine Kapazitätsgrenze, was zu einer

Die Abmessungen der beiden Module betragen: 305 x 270 x 85 mm (B x T x H). Das Gesamtgewicht inklusive Kabel liegt bei 11,6 kg.

Was fehlt...

Leider fehlt das Bedienteil, bei dem es sich damals typischerweise um ein Becker AT400 [4] handelte. Vermutlich wurde es samt dem Fahrzeug, in dem die Telefonanlage damals eingebaut war, verschrottet. Bei solch einem Bedienteil (**Bild 8**) handelt es sich im Prinzip lediglich um eine Ansammlung von einigen Signallämpchen, einem Summer und ein paar Schaltern. Hinzu kam noch ein Telefonhörer des Typs „Funk 70“ [5]. Die eigentliche Elektronik steckte ja in den beiden Blechkisten.

Das Bediengerät zeigte die von den *Funkfernvermittlungsstellen* kommenden Anrufe mittels Signallämpchen und dem Summer an. Falls eine Verbindung zur *FuVst* möglich war, leuchtete ein grünes Lämpchen bei vorhandenem Freizeichen.



Bild 8.
Das Steuergerät AT400 des Herstellers „Becker“ für den formschönen Einbau ins Armaturenbrett. Abbildung mit freundlicher Genehmigung von oebf.de.

Mit einem Drehschalter wählte man den zuständigen Funkkanal der jeweiligen FuVst passend zum Aufenthaltsort des Autos und der jeweiligen Funkzone. Später erschienen auch noch verbesserte Bediengeräte (**Bild 9**). Es gab dann sogar „automatische“ Ausführungen (**Bild 10**). Neben der Empfangsfeldstärke ermöglichten einige Luxusausführungen auch die Anzeige der Spannung der Autobatterie, da etliche Eigner vergaßen, vor dem Verlassen des Autos das Funktelefon auszu-

Aufnahmesperre der DBP führte.

Interessant in diesem Zusammenhang: Das erste Handy, das diesen Namen verdient, ist Motorolas Dynatac 8000 (der berühmte „Knochen“) mit einer Betriebszeit von lediglich 30 Minuten. Dieses Modell wurde am 13.6.1983 vorgestellt und ist also gerade einmal 31 Jahre alt!

Ab dem 1.9.1985 gab es mit dem C-Netz für die Autotelefonie schon eine Alternative im 450-MHz-Band. Erstmals wurde hier die vollautomatische digitale Vermittlung mit einheitlicher Vorwahl 0161 im gesamten Bundesgebiet eingeführt. Auch das „Handover“ (unterbrechungsfreie Weiterleitung eines Gespräches von einer Funkzelle zur nächsten) war durch die automatisierte Ermittlung des Aufenthaltsortes der mobilen Teilnehmer erstmals in einem Funknetz realisiert. Das B-Netz wurde aber noch weiterbetrieben und erst 1994 eingestellt.

Das C-Netz hatte zu seinen Hochzeiten etwa 800.000 Teilnehmer. Die monatlichen Bereitstellungsgebühren von anfangs 120 DM sanken mit der Zeit bis auf niedrige 19 DM. Außerdem wurden die Autotelefone immer kleiner und leichter. Dadurch wurden sogar erstmals tragbare Mobiltelefone in Deutschland möglich. Auch das C-Netz wurde noch bis weit nach der Einführung seines Nachfolgers aufrechterhalten und erst im Jahre 2000 deaktiviert.

Heute fasst man die A-, B- und C-Netze sprachlich zusammen und spricht lediglich noch von den Mobilfunknetzen der ersten Generation (1G). Während 1G-Geräte noch echte „Sprechfunkgeräte“ mit analoger Sprachsignalübertragung darstellten, waren die nachfolgenden Funknetze voll digital und ermöglichten so Millionen Teilnehmer bei reduziertem Frequenzbedarf.

Weitere Informationen:

<http://izmf.de>

<http://www.oebf.de/A-Netz/ANetz.html>

<http://www.oebf.de/B-Netz/BNetz.html>

<http://www.oebf.de/C-Netz/CNetz.html>

<http://www.mobilfunk-geschichte.de>

<http://wissen.de/die-geschichte-der-mobiltelefone>

<http://de.wikipedia.org/wiki/A-Netz>

<http://de.wikipedia.org/wiki/B-Netz>

<http://de.wikipedia.org/wiki/C-Netz>

<http://de.wikipedia.org/wiki/D-Netz>

<http://de.wikipedia.org/wiki/E-Netz>

<http://www.3gpp.org>

http://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Marktbeobachtung/Deutschland/Mobilfunkteilnehmer/Mobilfunkteilnehmer_node.html

<http://en.wikipedia.org/wiki/>

[Universal_Mobile_TeleKommunikations_System](#)

[http://en.wikipedia.org/wiki/LTE_\(teleKommunikation\)](http://en.wikipedia.org/wiki/LTE_(teleKommunikation))

http://en.wikipedia.org/wiki/LTE_Advanced

<http://www.focus.de/digital/handy/mobilfunkgeschichte/>

Über den Autor

Schon als Jugendlicher sammelte Gerd Kowalewski alte Röhrenfernseher und ähnliche Apparate und versuchte sie zu reparieren. Etwas später begann er ein Studium als Elektro-Ingenieur, nach dessen Abschluss er sich in der jungen Firma CPV zunächst mit der Entwicklung von Produkten zur Datenkommunikation beschäftigte und dann auch mit der (fernöstlichen) Produktentwicklung befasst war. Er war auch mehrere Jahre im Bereich Elektrofahrzeuge aktiv. Die nächste Station seines Berufslebens betraf den Think Tank mpe (Mannesmann Pilotentwicklung) in München, was in einigen von ihm gehaltenen Patenten mündete. Nachdem er sich mit der Firma GKC (GK Electronic Consulting) selbstständig machte, erzwangen gesundheitliche Gründe zehn Jahre später seine Pensionierung. Gerd ist schon 40 Jahre lang ein treuer Elektor-Leser und interessiert sich vor allen Dingen für trickreiche Schaltungen und effiziente Hardware-Lösungen sowie für wissenschaftliche Fortschritte – auch die der Vergangenheit.



schalten. Als Sprechereinrichtung fungierte übrigens ein dem damaligen verbreiteten Telefentyp W48 ähnlicher Telefonhörer [5]. Er wurde meist zusammen mit dem Bediengerät in der Fahr-

zeugkonsole neben dem Autoradio montiert, doch wenn man sich einen Chauffeur leisten konnte, natürlich im Fond.

Wie schon erwähnt war Telefunken nicht der alleinige Hersteller solcher Bediengeräte. Auch die bis vor kurzem noch eigenständige und auf Autoradios etc. spezialisierte Firma Becker lieferte welche, vermutlich aufgrund von Verträgen mit den Herstellern von Premium-Autos wie Mercedes-Benz und (damals noch) der Opel AG. Die Autoradios wurden damals übrigens auch häufig als „Autosuper“ bezeichnet, was sich auf das Empfängerprinzip bezieht.

Bild 9.
Spezialversion des Steuergeräts für das Armaturenbrett eines Mercedes W108/W109. Abbildung mit freundlicher Genehmigung von oeb1.de.



Bild 10.
Automatik- und Luxusversionen von Steuergeräten lösten die einfache Standardvariante schnell ab.



Fazit

Das Telefunken-Funktelefon des Typs S/E 160E15 öbL macht ganz besonders deutlich, wie rasant sich der Mobilfunk in den letzten 50 Jahren entwickelt hat. Doch ohne die Entwicklungen der dahinter stehenden Mobilfunknetze, angefangen beim ersten A-Netz im „öbL“ bis hin zu den heutigen LTE-Breitband-Funknetzen (Long Term Evolution), wären moderne Handys gar nicht denkbar.

Das Fundstück wirkte zwar soweit funktionsfähig, doch der Autor konnte der Versuchung widerstehen, es zu testen und einzuschalten. Ein fähiger Restaurator oder Sammler würde sicherlich **vor** dem Einschalten einige Kondensatoren entweder neu formieren oder ersetzen, um Folgeschäden zu vermeiden. Das komplette Funktelefon hat inzwischen einen guten Platz in einer privaten Techniksammlung gefunden.

(140268)

Informationsquellen:

- [1] www.oeb1.de (eine bemerkenswerte private Sammlung von Stephan Hessberger)
- [2] www.oeb1.de/A-Netz/Technik/Technik.html
- [3] www.oeb1.de/A-Netz/Technik/Tabelle.html
- [4] www.oeb1.de/A-Netz/Geraete/becker/AT400/AT400.html
- [5] www.oeb1.de/sonstiges/Hoerer/hoerer.html
- [6] www.oeb1.de/A-Netz/Doku/AT400_Angebot.JPG

Hexadoku Sudoku für Elektroniker

Die Wissenschaft weiß heute, dass regelmäßige Betätigung auch unser Gehirn fit hält. Das ist natürlich ein Extra-Grund für Sie, unser monatliches Hexadoku anzugehen.

Ein weiterer Grund sind sicher unsere fünf Elektor-Buchgutscheine. Um einen davon gewinnen können, müssen Sie uns nur die Zahlen in den grauen Kästchen zusenden!

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 einen von fünf Buchgutscheinen im Wert von 50 Euro gewinnen!

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 2014!

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

Die richtige Lösung ist: **E80F4**.

Einen Elektor-Buchgutschein über je 50 € haben gewonnen:

Ulf Claesson, Larry Burns, A. van Maris, Jairo Rotava und Zvi Herman.

Herzlichen Glückwunsch!

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

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

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



Mit Visual Basic in die analoge Welt

1 VB-Express und die Hardware

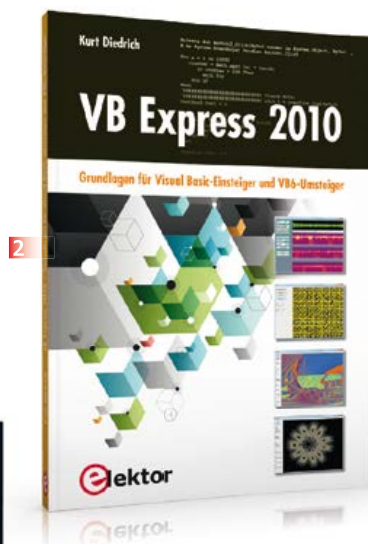
Visual Basic zählt nach wie vor zu den sehr weit verbreiteten Programmiersprachen. Seine Beliebtheit resultiert gerade für den Einsteiger aus der schnellen Erlernbarkeit und der einfachen Lesbarkeit des Programmcodes. Dieses Buch ist für Einsteiger in die Programmiersprache Visual Basic.NET gedacht, auch und gerade unter Berücksichtigung der Bedürfnisse von Elektronikern.

287 Seiten (kart.) • ISBN 978-3-89576-270-3
€ 36,80 • CHF 45,95

Grundlagen für Visual Basic-Einsteiger und VB6-Umsteiger

2 VB Express 2010

Dieses Buch unterstützt den Anwender bei den ersten Schritten mit Visual Basic, in dem es sich auf die Werkzeuge der Toolbox und deren Eigenschaften konzentriert, die zum Schreiben praktisch verwertbarer Programme notwendig sind. Zu jedem Thema findet



der Leser ausführlich kommentierte Beispielprogramme, die er selbst ausprobieren kann und die sich auf das Mindeste beschränken, was zum Starten der Software notwendig ist.

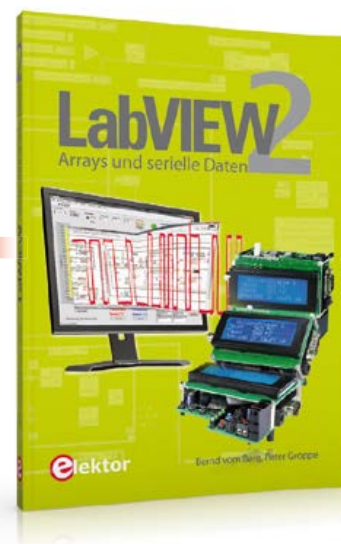
284 Seiten (kart.) • ISBN 978-3-89576-269-7
€ 34,80 • CHF 43,95

45 Experimente mit Hard- und Software für Elektroniker

3 Raspberry Pi

Dieses Buch beschreibt 45 spannende und interessante Projekte mit Raspberry Pi, wie zum Beispiel ein Wechselblinklicht, eine Motorregelung, Erzeugen und Verarbeiten analoger Signale, ein digitales Thermometer, ein Lichtmesser. Aber auch kompliziertere Projekte wie eine Motor-Geschwindigkeitsregelung, ein Webserver mit CGI (Common Gateway Interface) und Client-Server-Programme werden vorgestellt. Sie können dieses Buch als Projektbuch verwenden und die Projekte nachbauen, um sie dann in der Praxis einzusetzen.

271 Seiten (kart.) • ISBN 978-3-89576-273-4
€ 39,80 • CHF 49,95



Arrays und serielle Daten

4 LabVIEW 2

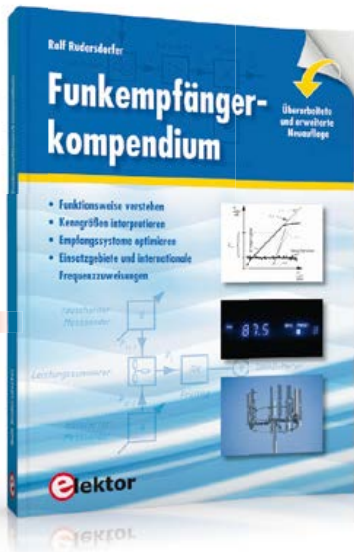
Der zweite Band der LabVIEW-Lehrbuchreihe beschäftigt sich u.a. mit Arrays, Cluster und den seriellen VISA-Funktionen. Als Erstes werden vier neue zusammengesetzte Datentypen (Enum, Ring, Array, Cluster) vorgestellt und deren Verwendung wird anhand zahlreicher praktischer Beispiele und Übungen erläutert. Danach wird es praktisch: Ein 8051er-Mikrocontroller dient dabei als Datenquelle und -senke für verschiedene LabVIEW-VIs.

248 Seiten (kart.) • ISBN 978-3-89576-274-1
€ 34,80 • CHF 43,95

Der professionelle Ratgeber für Funkempfangstechnik

5 Funkempfänger-kompodium

Wollten Sie schon immer wissen, wie sich die klassische Funkempfängertechnik fortentwickelt hat? Wie funktionieren professionelle Funkempfänger heute und was können sie leisten? Welche Empfangssysteme und Techniken stehen heute zur Verfügung? Möchten Sie

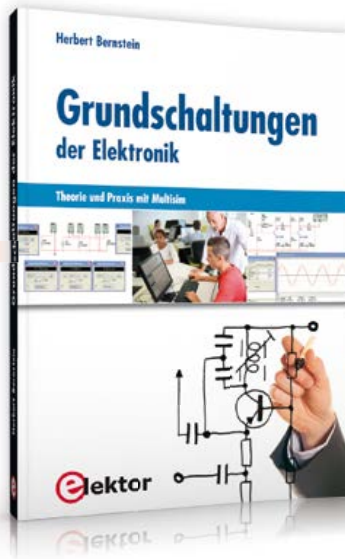


auch ausgefallene Anwendungen von Empfängern kennenlernen und wissen, wie ein Software Defined Radio (Digitalempfänger) nun wirklich funktioniert und was der letzte Stand der entsprechenden Technik kann? In diesem Buch findet man die Antworten!
397 Seiten (geb.) • ISBN 978-3-89576-276-5
€ 49,00 • CHF 61,95

Theorie und Praxis mit Multisim **6 Grundsaltungen der Elektronik**

Dieses Buch ist ein Nachschlagewerk über Elektronik mit praxisorientierten Fakten und ausführlichen Erklärungen. Der Autor hat selbst für komplexe Vorgänge oder Formeln praktische kurze Erklärungen und Näherungsrechnungen entwickelt, ohne die Darstellungen zu simplifizieren. Als Ausgangspunkt wurde das Simulationsprogramm Multisim gewählt, das zahlreiche Bauelemente und umfangreiche Messinstrumente zur Verfügung stellt.

360 Seiten (kart.) • ISBN 978-3-89576-286-4
€ 44,00 • CHF 54,95



Schnell und einfach mit Arduino und Elektor-Shield

7 Mikrocontroller verstehen und anwenden

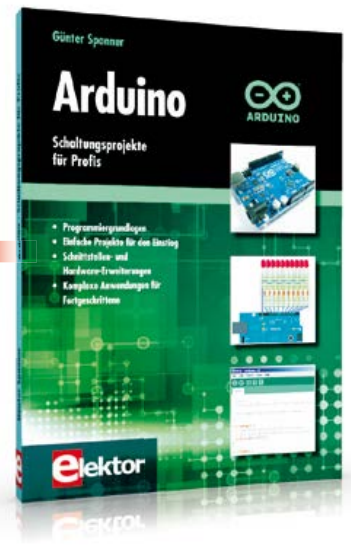
Mit diesem Buch erweitert der Anfänger auf dem Gebiet der Mikrocontroller, der Arduino-User bzw. -Enthusiast seine Mikrocontroller-Kenntnisse auf Grund eigener Erfahrungen und Erfolgserlebnisse und wird dazu noch ganz nebenbei in die Welt des Arduino und seiner Entwicklungsumgebung eingeführt.

392 Seiten (kart.) • ISBN 978-3-89576-296-3
€ 42,00 • CHF 52,95

Schaltungsprojekte für Profis

8 Arduino

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



Einführungsbeispielen ist der weitere Weg nicht mehr klar erkennbar, weil oft detaillierte Projektbeschreibungen fehlen. Hier setzt dieses Buch an. Systematisch werden Projekte vorgestellt, die in verschiedene Themengebiete einführen. Dabei wird neben den erforderlichen theoretischen Grundlagen stets größter Wert auf eine praxisorientierte Ausrichtung gelegt.

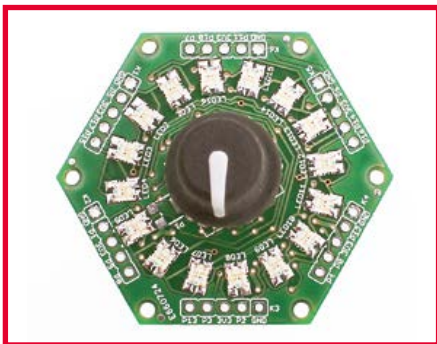
270 Seiten (kart.) • ISBN 978-3-89576-257-4
€ 39,80 • CHF 49,95

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

•Nächsten Monat in Elektor



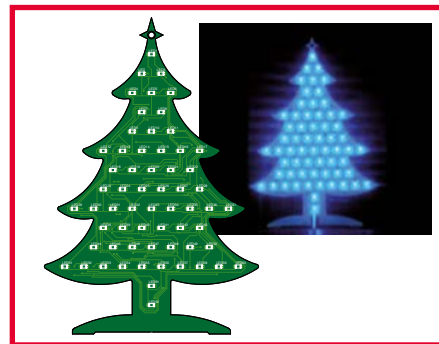
Cool Controller Concept CoCo-ri-Co

Viele Geräte benötigen zur Bedienung eigentlich nur eine oder zwei Tasten. Wir haben deshalb eine kleine sechseckige Platine entworfen, die in Ein- oder Mehrzahl hinter der Frontplatte Ihres Selbstbauprojekts montiert werden kann. Das Gerät wird dann mit einem Drehencoder oder Drucktaster bedient, ein Ring von 16 zweifarbigen LEDs gibt die Schalterposition an.



Displayplatine für VariLab 402

Die Displayplatine garantiert eine einfache und übersichtliche Bedienung des in diesem Heft vorgestellten VariLab-402-Netzgeräts. Auf dem 4-zeiligen LCD werden alle gemessenen und errechneten Werte und Einstellungen angezeigt. Das Gerät wird mit zwei Drehgebern und einem Drucktaster eingestellt. Die Schaltung ist so ausgelegt, dass sie auch für andere (Mess- und Regel-)Elektronik verwendet werden kann.



Programmierbarer Weihnachtsbaum

Da ist sie wieder, die Weihnachtsschaltung in der Dezemberausgabe! Dieses Mal haben wir in Zusammenarbeit mit Eurocircuits einen Platinen-Weihnachtsbaum entwickelt, der mit einer Matrix von 62 LEDs bestückt ist. Die Besonderheit ist, dass man allerlei Lichteffekte und Muster programmieren kann. Einfach an den Computer hängen und die LED-Muster mit der Maus in einem Browser eingeben!

Änderungen vorbehalten. Elektor Dezember 2014 erscheint am 19. November 2014. Verkaufsstellen findet man unter www.pressekaufen.de.

Rund um die Uhr und
sieben Tage die Woche

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

The screenshot shows the 'elektorlabs' website with the tagline 'Sharing Electronics Projects'. It features a navigation bar with 'Home', 'Proposals', 'In Progress', and 'Finished'. A prominent banner for 'JOIN ELEKTOR TOMORROW' is visible. Below this, there are sections for 'Proposals', 'In Progress', and 'Finished' projects, each with a 'Popular' sub-section. A 'Featured Video' section shows a video player. At the bottom, there is a 'Create a Project' section with a button to 'Create a new project or enter a proposal'.

Integrierte 3D-Gesten- und Positionserfassung

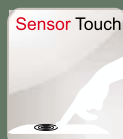
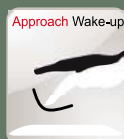
mit Microchips GestIC®-Technologie



Microchips patentierte GestIC®-3D-Technologie ist eine Einchip-Lösung für die Gestenerkennung in Echtzeit. Der MGC3130 ermöglicht in jedem Produkt zu sehr geringen Kosten den nächsten Durchbruch bei der Entwicklung von Benutzerschnittstellen.

MGC3130

- Der MGC3130 läuft mit der Colibri Gesture Suite und sorgt für eine sichere Erkennung komplexer 3D-Gesten ohne Host-Datenverarbeitung
- Der MGC3130 liefert die Ergebnisse (erkannte Gesten / XYZ-Position) über eine I²C™-Bus-Schnittstelle oder GPIOs an den Host



microchip
DIRECT
www.microchipdirect.com

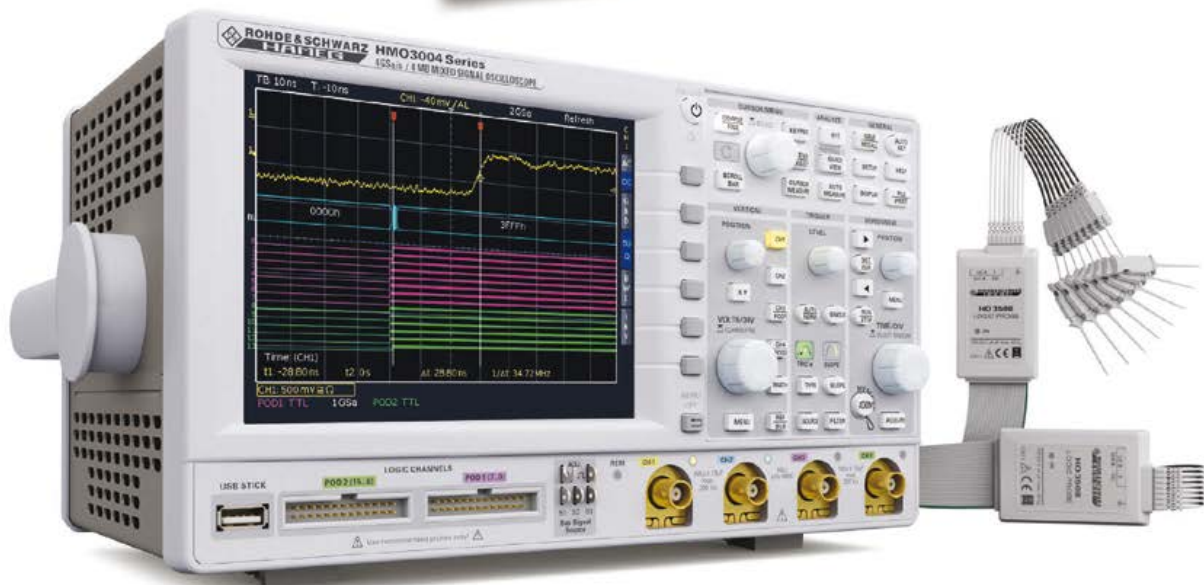
 **MICROCHIP**
microchip.com/gestic

Alles drin: HMO Complete

Nur bis zum 31.12.2014: Sparen Sie bis zu 40 % beim Kauf eines HMO Complete-Bundles, bestehend aus einem HMO3000 wahlweise mit zwei oder vier Kanälen, 500 MHz Bandbreite, allen Software-Optionen und zwei Digital-Tastköpfen.

Neugierig?
www.hameg.com

Bis zu
40%
sparen!



HAMEG®
Instruments
A Rohde & Schwarz Company

inklusive H0010
Busanalyse für I²C, SPI, UART

inklusive H0012
Busanalyse für CAN/LIN

inklusive H0014
Segmentierter Speicher

inklusive H03516
2x 8-Kanal Digital-Tastköpfe

f+ YouTube #HAMEGcom


ROHDE & SCHWARZ